How Can a Wellness Technology Company Play It Smart?

Frida Ekner

2024-02-22

Abstract

Bellabeat, a high-tech company specializing in health-focused smart products, has experienced rapid growth since its founding in 2013. With a mission to empower women with knowledge about their health and habits, Bellabeat collects data on activity, sleep, stress, and reproductive health through its smart devices. In this project, conducted as part of the Google Data Analytics Certificate program, an analysis of smart device usage data was performed to uncover trends and insights. The study aimed to provide high-level recommendations for informing Bellabeat's marketing strategy. The analysis revealed significant trends, including users spending a considerable portion of their time inactive, a positive correlation between steps taken and calories burned, and distinct patterns in daily activity and sleep. Based on these findings, recommendations were made to enhance Bellabeat's marketing strategy, such as introducing activity reminders, implementing fitness challenge groups, and incorporating morning exercise initiatives. These insights can help Bellabeat better serve its customers and optimize revenue generation opportunities.

1 Introduction and purpose

This project is made as an optional part of the Google Data Analytics Certificate. Bellabeat is a high-tech company that manufactures health-focused smart products, founded by Urška Sršen and Sando Mur in 2013 and has grown rapidly and quickly positioned itself as a tech-driven wellness company for women since. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website.

Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy. Sršen requests an analysis of smart device usage data in order to gain insight into how consumers use one Bellabeat smart devices. The research questions that will guide this project are the following.

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?

2 Data sources

The FitBit Fitness Tracker Data is a publicly accessible dataset provided by Möbius under the CC0 database protection license. This dataset consists of 18 .csv files and encompasses the aggregated personal fitness tracker information from thirty FitBit users who have provided consent for the submission of their personal data. The data includes parameters such as heart rate, sleep patterns, activity intensities, physical exercises, and other pertinent information essential for analyzing their lifestyle habits. The following files from the dataset were selected:

- dailyActivity_merged.csv
- hourlyCalories_merged.csv
- hourlySteps_merged.csv
- sleepDay_merged.csv

3 Cleaning and manipulation

3.1 Data cleaning

Prior to the process of cleaning the data, some necessary packages first need to be downloaded, the data has to be imported and a preview is imperative to be familiarized with he structure. After that the data will be checked for null values and duplicated records will be removed.

```
library(tidyverse)
library(skimr)
library(here)
library(janitor)
```

```
daily_activity <- read.csv("dailyActivity_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
hourly_steps <- read.csv("hourlySteps_merged.csv")
sleep <- read.csv("sleepDay_merged.csv")
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
data_preview <- list(
  daily_activity = sample_n(daily_activity,1),
  hourly_calories = sample_n(hourly_calories,1),
  hourly_steps = sample_n(hourly_steps,1),
  sleep = sample_n(sleep,1)
)
data_preview
```

```
## $daily_activity
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 4445114986     5/4/2016       2923          1.96            1.96
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0                  0                        0
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                1.96                       0                 0
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                   0                  180              897     2070
##
## $hourly_calories
##           Id         ActivityHour Calories
## 1 6117666160 4/12/2016 7:00:00 AM       62
##
## $hourly_steps
##           Id        ActivityHour StepTotal
## 1 4702921684 5/4/2016 6:00:00 PM       730
##
## $sleep
##           Id             SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 4445114986 5/9/2016 12:00:00 AM                 1                457
##   TotalTimeInBed
## 1            533
```

3.1.1 Checking for null values

```
null_values <- list(
  daily_activity = any(is.na(daily_activity)),
  hourly_calories = any(is.na(hourly_calories)),
  hourly_steps = any(is.na(hourly_steps)),
  sleep = any(is.na(sleep))
)
null_values
```

```
## $daily_activity
## [1] FALSE
##
## $hourly_calories
## [1] FALSE
##
## $hourly_steps
## [1] FALSE
##
## $sleep
## [1] FALSE
```

3.1.2 Removing duplicated records

```
library(dplyr)
daily_activity <- distinct(daily_activity)
```

```r
hourly_calories <- distinct(hourly_calories)
hourly_steps <- distinct(hourly_steps)
sleep <- distinct(sleep)
```

3.1.3 Checking the number of unique IDs

```r
unique_ids <- list(
  daily_activity = daily_activity %>% summarise(n_distinct(Id)),
  hourly_calories = hourly_calories %>% summarise(n_distinct(Id)),
  hourly_steps = hourly_steps %>% summarise(n_distinct(Id)),
  sleep = sleep %>% summarise(n_distinct(Id))
)
unique_ids
```

```
## $daily_activity
##   n_distinct(Id)
## 1             33
##
## $hourly_calories
##   n_distinct(Id)
## 1             33
##
## $hourly_steps
##   n_distinct(Id)
## 1             33
##
## $sleep
##   n_distinct(Id)
## 1             24
```

3.1.4 Finding the data types

```r
data_types <- list(
  daily_activity = str(daily_activity),
  hourly_calories = str(hourly_calories),
  hourly_steps = str(hourly_steps),
  sleep = str(sleep)
)
```

```
## '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 ...
## '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
##  $ Calories    : int  81 61 59 47 48 48 48 47 68 141 ...
## '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 ...
## 'data.frame':    410 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 ...
```

3.2 Data transformation

In the previous steps when the data was explored some possibilities for improvements were found. It was for example noticed that the dates in each table are in string types, which is quite inconvenient. A transformation of the types to datetime is necessary to prepare the data for the next step, analysis.

3.2.1 Renaming columns

```
daily_activity <- daily_activity %>%
  rename(Date = ActivityDate,
         Steps = TotalSteps,
         Distance = TotalDistance)

hourly_calories <- hourly_calories %>%
  rename(Time = ActivityHour)

hourly_steps <- hourly_steps %>%
  rename(Time = ActivityHour,
         Steps = StepTotal)

sleep <- sleep %>%
  rename(Date = SleepDay,
         MinutesAsleep = TotalMinutesAsleep,
         TimeInBed = TotalTimeInBed)
```

3.2.2 Converting types

```
daily_activity$Date <- as.POSIXct(daily_activity$Date, format = "%m/%d/%Y")
hourly_calories$Time <- as.POSIXct(hourly_calories$Time, format = "%m/%d/%Y %I:%M:%S %p")
hourly_steps$Time <- as.POSIXct(hourly_steps$Time, format = "%m/%d/%Y %I:%M:%S %p")
sleep$Date <- as.POSIXct(sleep$Date, format = "%m/%d/%Y %I:%M:%S %p")
```

3.2.3 Creating new columns

```
library(lubridate)
```

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

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

```
daily_activity$DayOfTheWeek <- weekdays(daily_activity$Date)
hourly_calories$DayOfTheWeek <- weekdays(hourly_calories$Time)
```

3.2.3 Merging data

```
library(dplyr)
merged_daily <- merge(daily_activity, sleep, by = "Date")
hourly_steps$Time <- as.POSIXct(hourly_steps$Time, tz = "UTC")
hourly_calories$Time <- as.POSIXct(hourly_calories$Time, tz = "UTC")
hourly_steps$Time <- round(hourly_steps$Time, units = "secs")
hourly_calories$Time <- round(hourly_calories$Time, units = "secs")
merged_hourly <- merge(hourly_steps, hourly_calories, by = "Time")
```

4 Analysis

To start of the analysis some general statistics from the data is first viewed. One interesting finding is that the average number of steps is 7675, with an average of 320 steps every hour.

```
summary_stats_1 <- summary(merged_daily)
summary_stats_2 <- summary(merged_hourly)
print(summary_stats_1)
```

```
##       Date                              Id.x               Steps
##  Min.   :2016-04-12 00:00:00.00   Min.   :1.504e+09   Min.   :    0
##  1st Qu.:2016-04-19 00:00:00.00   1st Qu.:2.320e+09   1st Qu.: 3821
##  Median :2016-04-26 00:00:00.00   Median :4.445e+09   Median : 7412
##  Mean   :2016-04-25 20:40:27.38   Mean   :4.861e+09   Mean   : 7675
##  3rd Qu.:2016-05-03 00:00:00.00   3rd Qu.:6.962e+09   3rd Qu.:10735
##  Max.   :2016-05-12 00:00:00.00   Max.   :8.878e+09   Max.   :36019
##     Distance      TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.630   1st Qu.: 2.630   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.260   Median : 5.260   Median :0.0000           Median : 0.210
##  Mean   : 5.516   Mean   : 5.502   Mean   :0.1049           Mean   : 1.508
##  3rd Qu.: 7.720   3rd Qu.: 7.720   3rd Qu.:0.0000           3rd Qu.: 2.040
##  Max.   :28.030   Max.   :28.030   Max.   :4.9421           Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.00       Min.   :0.00000
##  1st Qu.:0.0000           1st Qu.: 1.96       1st Qu.:0.00000
##  Median :0.2400           Median : 3.38       Median :0.00000
##  Mean   :0.5699           Mean   : 3.36       Mean   :0.00158
##  3rd Qu.:0.8100           3rd Qu.: 4.79       3rd Qu.:0.00000
##  Max.   :6.4800           Max.   :10.71       Max.   :0.11000
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:129.0        1st Qu.: 730.0
##  Median :  4.00    Median :  7.00      Median :199.0        Median :1057.0
##  Mean   : 21.25    Mean   : 13.66      Mean   :194.2        Mean   : 992.7
##  3rd Qu.: 32.00    3rd Qu.: 20.00      3rd Qu.:266.0        3rd Qu.:1229.0
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories    DayOfTheWeek            Id.y           TotalSleepRecords
##  Min.   :   0   Length:12535       Min.   :1.504e+09   Min.   :1.000
##  1st Qu.:1838   Class :character   1st Qu.:3.977e+09   1st Qu.:1.000
##  Median :2150   Mode  :character   Median :4.703e+09   Median :1.000
##  Mean   :2315                      Mean   :4.993e+09   Mean   :1.123
##  3rd Qu.:2796                      3rd Qu.:6.962e+09   3rd Qu.:1.000
##  Max.   :4900                      Max.   :8.792e+09   Max.   :3.000
##  MinutesAsleep   TimeInBed
##  Min.   : 58   Min.   : 61.0
##  1st Qu.:361   1st Qu.:403.0
##  Median :432   Median :463.0
##  Mean   :419   Mean   :458.5
##  3rd Qu.:490   3rd Qu.:526.0
##  Max.   :796   Max.   :961.0
```

```
print(summary_stats_2)
```

```
##       Time                              Id.x               Steps
##  Min.   :2016-04-11 22:00:00.00   Min.   :1.504e+09   Min.   :    0.0
##  1st Qu.:2016-04-18 12:00:00.00   1st Qu.:2.320e+09   1st Qu.:    0.0
##  Median :2016-04-25 08:00:00.00   Median :4.445e+09   Median :   40.0
##  Mean   :2016-04-25 17:20:15.47   Mean   :4.856e+09   Mean   :  319.6
##  3rd Qu.:2016-05-02 15:00:00.00   3rd Qu.:6.962e+09   3rd Qu.:  355.0
##  Max.   :2016-05-12 13:00:00.00   Max.   :8.878e+09   Max.   :10554.0
##       Id.y              Calories      DayOfTheWeek
##  Min.   :1.504e+09   Min.   : 42.00   Length:672301
##  1st Qu.:2.320e+09   1st Qu.: 63.00   Class :character
##  Median :4.445e+09   Median : 83.00   Mode  :character
##  Mean   :4.856e+09   Mean   : 97.33
##  3rd Qu.:6.962e+09   3rd Qu.:108.00
##  Max.   :8.878e+09   Max.   :948.00
```

Averages by day of the week

From the table average_by_day one interesting, but maybe not that surprising finding, is that the longest average time spent awake in bed, 50 minutes, occurs on Sundays. A probable explanation is that many people try to go to bed earlier on Sundays than the previous two days, and have rescheduled their inner sleeping clock, and they therefor lay awake longer in bed.

```
library(dplyr)
averages_by_hour <- merged_hourly %>%
  group_by(DayOfTheWeek) %>%
  summarise(AverageHourlyCalories = mean(Calories),
            AverageHourlySteps = mean(Steps))
averages_by_day <- merged_daily %>%
  group_by(DayOfTheWeek) %>%
  summarise(AverageTimeInBed = mean(TimeInBed),
            AverageMinutesAsleep = mean(MinutesAsleep),
            AverageTimeAwakeInBed = (mean(TimeInBed)-mean(MinutesAsleep)))

print(averages_by_hour)
```

```
## # A tibble: 7 x 3
##   DayOfTheWeek AverageHourlyCalories AverageHourlySteps
##   <chr>                        <dbl>              <dbl>
## 1 Friday                        97.7               311.
## 2 Monday                        96.9               321.
## 3 Saturday                      99.8               344.
## 4 Sunday                        94.3               288.
## 5 Thursday                      97.2               323.
## 6 Tuesday                       98.5               333.
## 7 Wednesday                     96.7               314.
```

```
print(averages_by_day)
```

```
## # A tibble: 7 x 4
```

```
##   DayOfTheWeek AverageTimeInBed AverageMinutesAsleep AverageTimeAwakeInBed
##   <chr>                   <dbl>                <dbl>                  <dbl>
## 1 Friday                   445.                 406.                   39.6
## 2 Monday                   457.                 419.                   38.0
## 3 Saturday                 460.                 419.                   40.8
## 4 Sunday                   505.                 454.                   50.9
## 5 Thursday                 434.                 400.                   33.8
## 6 Tuesday                  443.                 405.                   38.7
## 7 Wednesday                470.                 435.                   35.9
```

Time slept on weekdays

```
library(dplyr)
convert_to_hours_minutes <- function(minutes) {
  hours <- floor(minutes / 60)
  remaining_minutes <- minutes %% 60
  return(paste(hours, "hours", remaining_minutes, "minutes", sep = " "))
}

filtered_data <- merged_daily %>%
  filter(DayOfTheWeek %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"),
         MinutesAsleep > 0)

filtered_data$HoursMinutesAsleep <- convert_to_hours_minutes(filtered_data$MinutesAsleep)
average_minutes_asleep <- filtered_data %>%
  group_by(DayOfTheWeek) %>%
  summarise(AverageHoursMinutesAsleep = convert_to_hours_minutes(mean(MinutesAsleep)))
print(average_minutes_asleep)
```

```
## # A tibble: 5 x 2
##   DayOfTheWeek AverageHoursMinutesAsleep
##   <chr>        <chr>
## 1 Friday       6 hours 45.5041551246537 minutes
## 2 Monday       6 hours 58.9477503628447 minutes
## 3 Thursday     6 hours 40.4636035105834 minutes
## 4 Tuesday      6 hours 44.8156171284635 minutes
## 5 Wednesday    7 hours 14.5664335664335 minutes
```

Most active hours

```
library(dplyr)

merged_hourly$Time <- hour(merged_hourly$Time)
hourly_calories_mean <- merged_hourly %>%
  group_by(Time) %>%
  summarise(mean_calories = mean(Calories))
print(hourly_calories_mean)
```

```
## # A tibble: 24 x 2
```

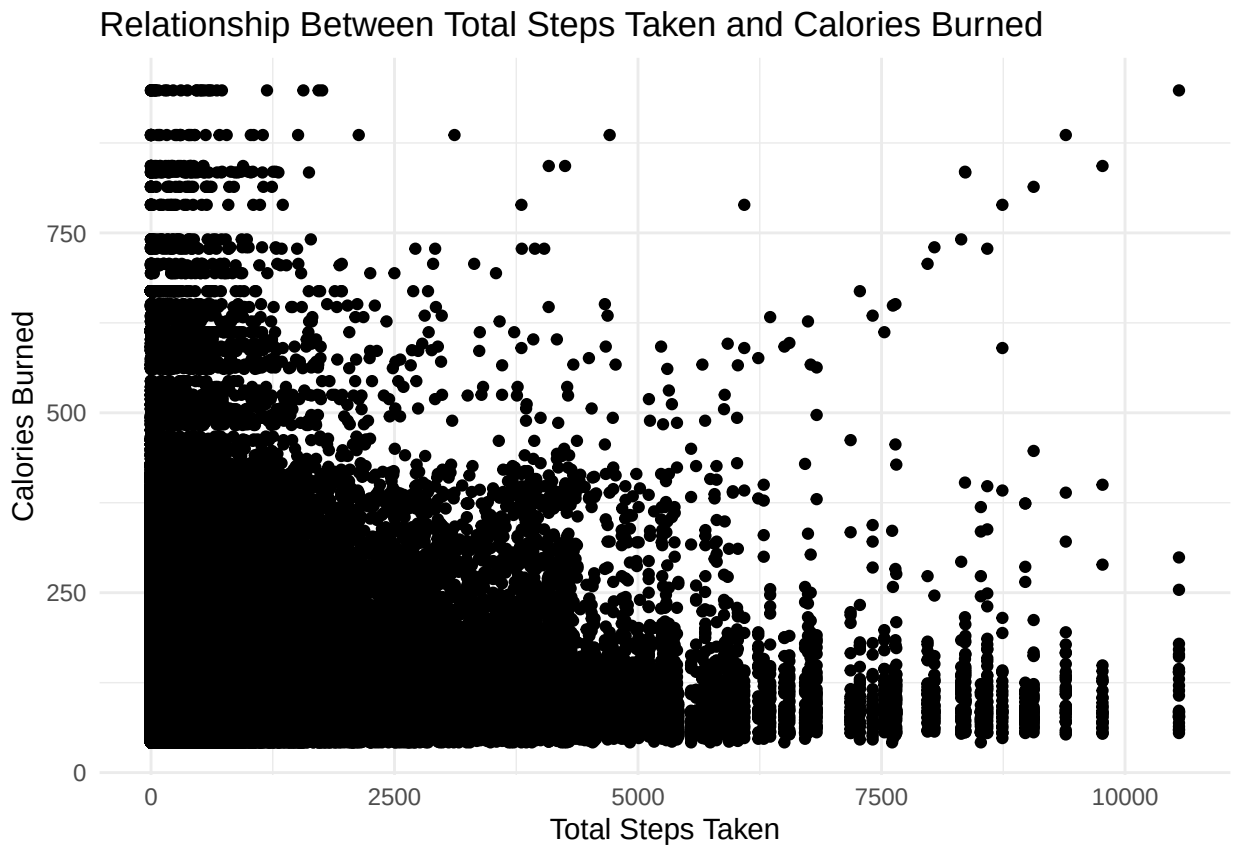
```
##     Time mean_calories
##    <int>         <dbl>
##  1     0          69.2
##  2     1          67.5
##  3     2          68.2
##  4     3          81.5
##  5     4          87.0
##  6     5          94.6
##  7     6         104.
##  8     7         106.
##  9     8         110.
## 10     9         110.
## # i 14 more rows
```

5 Results

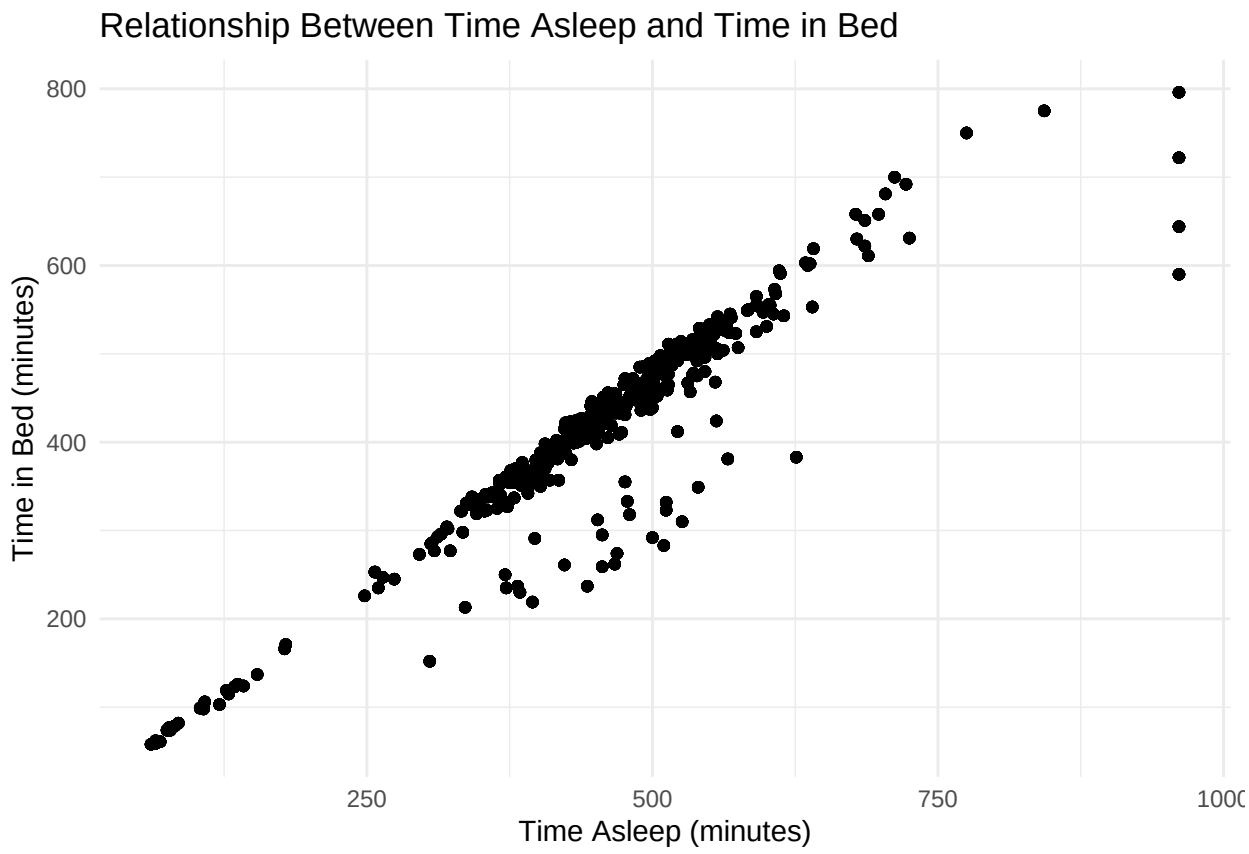
In this section some of the key findings are visualied, to get a better understanding of the results.

```
library(ggplot2)

scatter_plot <- ggplot(merged_hourly, aes(x = Steps, y = Calories)) +
  geom_point() +
  labs(title = "Relationship Between Total Steps Taken and Calories Burned",
       x = "Total Steps Taken",
       y = "Calories Burned") +
  theme_minimal()
print(scatter_plot)
```



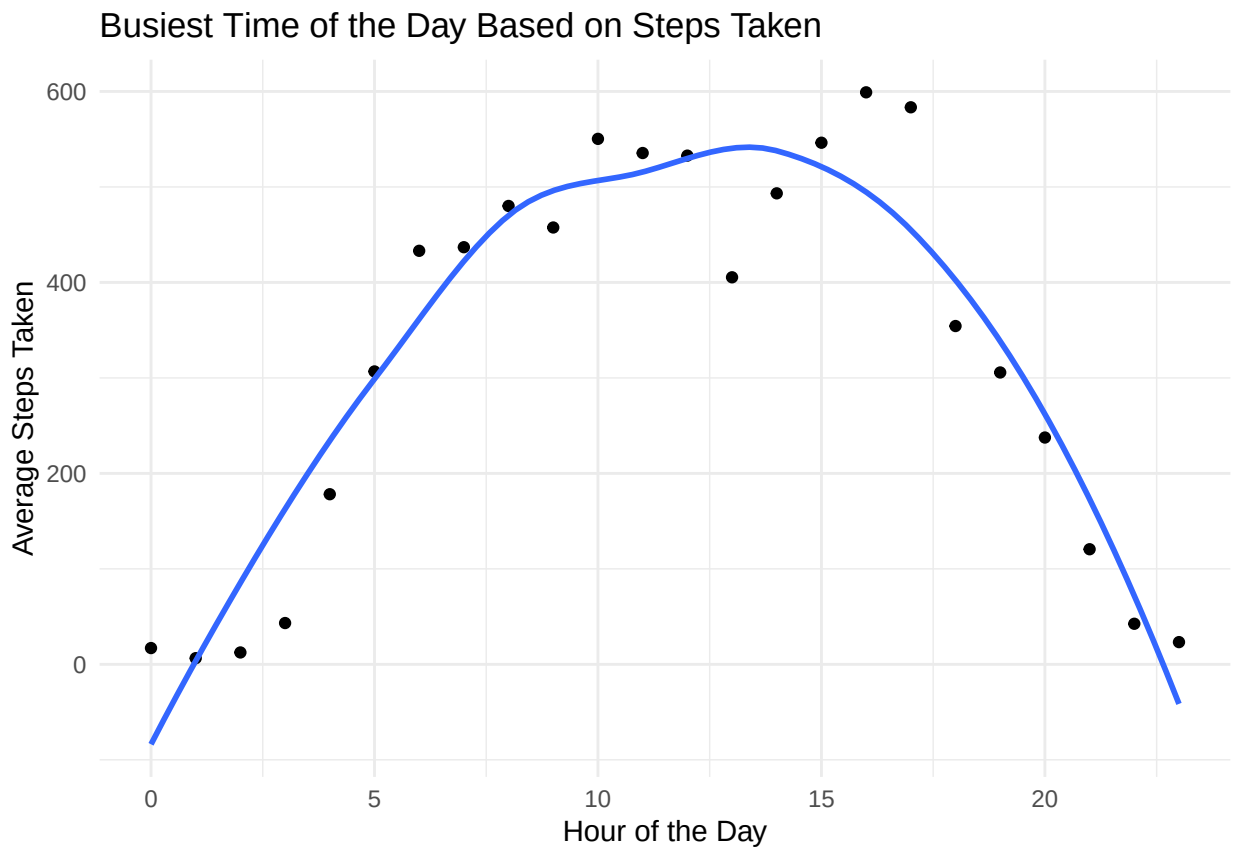

```
library(ggplot2)
scatter_plot <- ggplot(merged_daily, aes(x = TimeInBed, y = MinutesAsleep)) +
  geom_point(alpha = 0.5) +
  labs(title = "Relationship Between Time Asleep and Time in Bed",
       x = "Time Asleep (minutes)",
       y = "Time in Bed (minutes)") +  # Set axis labels
  theme_minimal()
print(scatter_plot)
```



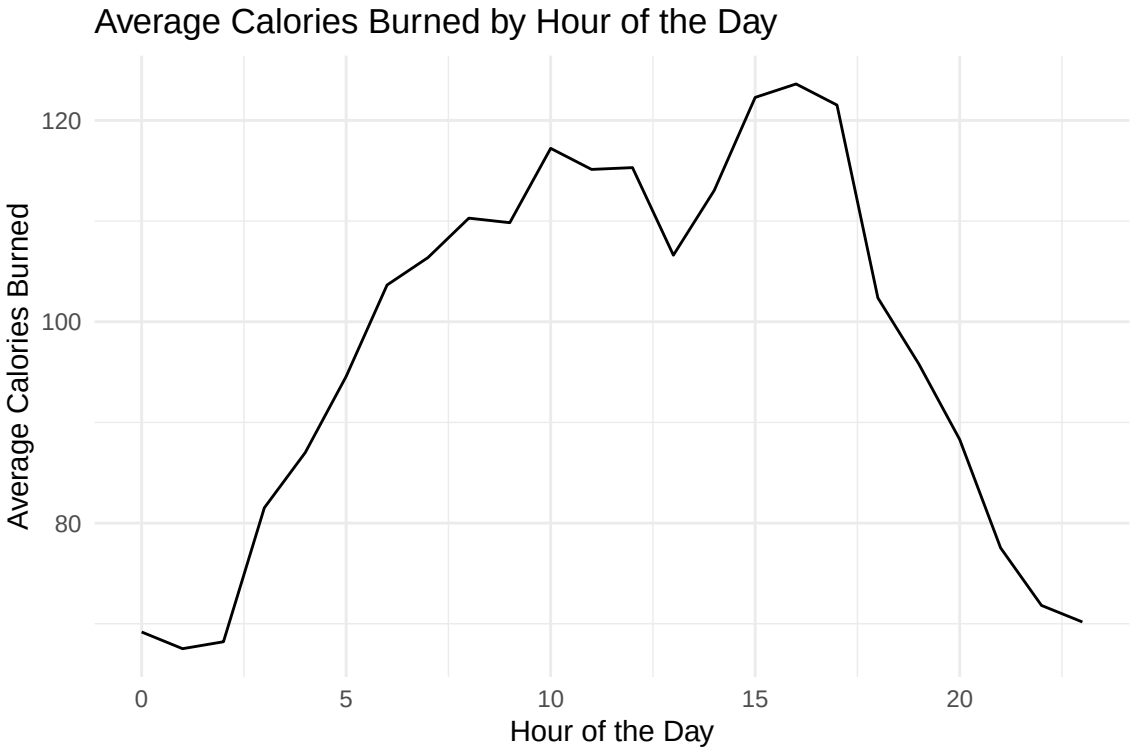

```
library(ggplot2)

hourly_steps <- aggregate(Steps ~ Time, data = merged_hourly, FUN = mean)
busy_time_plot <- ggplot(hourly_steps, aes(x = Time, y = Steps)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Busiest Time of the Day Based on Steps Taken",
       x = "Hour of the Day",
       y = "Average Steps Taken") +  # Set axis labels
  theme_minimal()
print(busy_time_plot)
```

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




```
ggplot(hourly_calories_mean, aes(x = Time, y = mean_calories)) +
  geom_line() +
  labs(title = "Average Calories Burned by Hour of the Day",
       x = "Hour of the Day",
       y = "Average Calories Burned") +
  theme_minimal()
```



6 Conclusion

The analysis of smart device usage revealed several noteworthy trends among users. Notably, users spend a significant portion of their time, approximately 81.2%, in an inactive state. Moreover, a positive correlation exists between the total number of steps taken and the total number of calories burned, indicating the importance of physical activity for caloric expenditure. Daily activity patterns exhibit distinct peaks during midday and early evening, with activity levels tapering off by late evening. Sleep patterns indicate an average nightly sleep duration of approximately 7 hours, with Sundays emerging as the day users tend to sleep the longest. In light of these findings, several recommendations can enhance Bellabeat's marketing strategy. These include the introduction of activity reminders within the Bellabeat app, the implementation of fitness challenge groups to encourage user engagement, and the incorporation of morning exercise initiatives to capitalize on peak activity periods. Additionally, establishing a customer feedback mechanism and introducing premium user features such as a "User Nearby" function can further optimize user experience and revenue generation opportunities for Bellabeat.