Bellabeat Product Analysis

Mingyu Lin

3/4/2022

Step 1: Ask

Background:

Bellabeat is a women's wearable tech company that provides smart wearable devices. The cofounder Urška Sršen believes analyzing smart device data can unlock potentials for growth. She wants the marketing analytics team to analyze smart device data to gain insights about consumer usage of company's products. The products range from Bellabeat app to various health trackers to Bellabeat membership service. She asks to apply the insight gained to one of these products.

Key Stakeholders:

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer
- Sando Mur: Bellabeat's cofounder and key member of the Bellabeat executive team
- Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy

Business Task:

To analyze smart device data in order to gain insights about smart device usage pertained to Bellabeat app and predict trends that guide marketing strategy for the company.

Business Questions:

It is important to define a framework to answer the business task. Since the goal is to analyze smart device usage data and apply insights gained to a Bellabeat product, in this case, the Bellabeat app, three questions can be asked:

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?

Step 2: Prepare

Information:

- The dataset is available on Kaggle from the user Mobius. It contains FitBit fitness data collected from 30 individuals who consented to data collection. The data was collected from Amazon Mechanical Turk, a crowdsourcing website.
- The dataset contains data collected from 4/12/16 to 5/12/16.
- There are a total of 18 csv files containing data about physical activity, steps, heart rate and sleep.
- Some of the tables are organized in long data format while others are in wide data format.

ROCCC:

To assess the credibility of the dataset, ROCCC framework was used.

- Reliable: Medium. The dataset contains accurate information because the data collected came from FitBIt fitness tracker, which collects real time information. However, the dataset came from only 30 users. The gender and other demographic information are not stated, which doesn't reflect the whole population.
- Original: High. Ultimately, the data are original since they are collected by fitness trackers from real individuals
- Comprehensive: Medium. While the dataset doesn't offer full health status reports about individuals, it is comprehensive enough for including information about physical activity, steps, heart rate, and sleep.
- Current: Low. The dataset is from 2016, which is not current.
- Cited: Low. The dataset doesn't come from vetted public official website. Instead, it was uploaded by a Kaggle user.

Limitations:

- The dataset has data from only 30 individuals. It may not reflect the whole population.
- The dataset is not currently up to date.
- The dataset may not be officially recognized. The data are not carefully examined by a credible source like the government.

Selection:

The following tables are used in this analysis:

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

Step 3: Process

Let's load the necessary libraries in R:

```
library(tidyverse) # collection of essential packages for data analysis
library(scales) # control the appearance of axis and legend labels
library(ggrepel) # repel overlapping text labels
```

Let's load the tables to R:

```
activity <- read.csv("dailyActivity_merged.csv")
sleep <- read.csv("sleepDay_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
```

Data Cleaning:

First, let's clean the activity table.

```
# summary of data
glimpse(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               <int> 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        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
# number of unique user ID
n_distinct(activity$Id)
```

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

```
# number of missing values
sum(is.na(activity))
```

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

```
# number of duplicated values
sum(duplicated(activity))
```

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

```
# adding new columns of total minutes and total hours
activity <- activity %>%
  mutate(TotalMinutes = VeryActiveMinutes + FairlyActiveMinutes + LightlyActiveMinutes +
           SedentaryMinutes, TotalHours = TotalMinutes/60)

head(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
##   TotalMinutes TotalHours
## 1         1094   18.23333
## 2         1033   17.21667
## 3         1440   24.00000
## 4          998   16.63333
## 5         1040   17.33333
## 6          761   12.68333
```

Next, let's clean the sleep table.

```
glimpse(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  <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```
n_distinct(sleep$Id)
```

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

```
sum(is.na(sleep))
```

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

```
sum(duplicated(sleep))
```

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

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

#adding a new column of portion asleep
sleep <- sleep %>% mutate(PortionAsleep =TotalMinutesAsleep/TotalTimeInBed)

head(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 PortionAsleep
## 1            346     0.9450867
## 2            407     0.9434889
## 3            442     0.9321267
## 4            367     0.9264305
## 5            712     0.9831461
## 6            320     0.9500000
```

Finally, let's clean the hourly_calories table.

```
glimpse(hourly_calories)
```

```
## Rows: 22,099
## Columns: 3
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/20~
## $ Calories     <int> 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73, 66, ~
```

```
n_distinct(hourly_calories$Id)
```

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

```
sum(is.na(hourly_calories))
```

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

```
sum(duplicated(hourly_calories))
```

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

```
# separating datetime into date, time, and AM or PM.
hourly_calories <- separate(hourly_calories, col=ActivityHour,
                            into=c('date', 'time', 'MorA'), sep=' ')

# separating time into hour, minutes, and second
hourly_calories <- separate(hourly_calories, col=time,
                            into=c('hour', 'minute', 'second'), sep=':')

# combining hour and AM or PM
hourly_calories <- unite(hourly_calories, "hour_new", c("hour", "MorA"), sep = " ")

# recoding 12-hour format to 24 hour format
hourly_calories <- hourly_calories %>%
  mutate(hour_new = recode(hour_new, "1 AM" = '1', "2 AM" = '2', "3 AM" = '3',
                           "4 AM" = '4', "5 AM" = '5', "6 AM" = '6',
                           "7 AM" = '7', "8 AM" = '8', "9 AM" = '9',
                           "10 AM" = '10',"11 AM" = '11', "12 PM" = '12',
                           "1 PM" = '13', "2 PM" = '14',"3 PM" = '15',
                           "4 PM" = '16', "5 PM" = '17', "6 PM" = '18',
                           "7 PM" = '19', "8 PM" = '20',"9 PM" = '21',
                           "10 PM" = '22', "11 PM" = '23', "12 AM" = '24'))

# converting character to integer data type
hourly_calories <- hourly_calories %>% transform(hour_new = as.integer(hour_new))

# creating new table using group_by and summarizing the average
hourly_calories_sum <- hourly_calories %>% group_by(hour_new) %>%
  summarize(avg_calories = mean(Calories))

head(hourly_calories_sum)
```

```
## # A tibble: 6 x 2
##   hour_new avg_calories
##      <int>        <dbl>
## 1        1         70.2
## 2        2         69.2
## 3        3         67.5
## 4        4         68.3
## 5        5         81.7
## 6        6         87.0
```

Step 4: Analyze

Let’s analyze the data by producing some descriptive statistical summaries.

activity table:

```
activity %>%
  select(TotalSteps,
         TotalDistance, VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance,
         SedentaryActiveDistance,
         TotalMinutes, TotalHours, VeryActiveMinutes,
         FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    VeryActiveDistance ModeratelyActiveDistance
##  Min.   :    0   Min.   : 0.000   Min.   : 0.000     Min.   :0.0000
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 0.000     1st Qu.:0.0000
##  Median : 7406   Median : 5.245   Median : 0.210     Median :0.2400
##  Mean   : 7638   Mean   : 5.490   Mean   : 1.503     Mean   :0.5675
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 2.053     3rd Qu.:0.8000
##  Max.   :36019   Max.   :28.030   Max.   :21.920     Max.   :6.4800
##  LightActiveDistance SedentaryActiveDistance  TotalMinutes
##  Min.   : 0.000      Min.   :0.000000        Min.   :   2.0
##  1st Qu.: 1.945      1st Qu.:0.000000        1st Qu.: 989.8
##  Median : 3.365      Median :0.000000        Median :1440.0
##  Mean   : 3.341      Mean   :0.001606        Mean   :1218.8
##  3rd Qu.: 4.782      3rd Qu.:0.000000        3rd Qu.:1440.0
##  Max.   :10.710      Max.   :0.110000        Max.   :1440.0
##    TotalHours       VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   : 0.03333   Min.   :  0.00    Min.   :  0.00      Min.   :  0.0
##  1st Qu.:16.49583   1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0
##  Median :24.00000   Median :  4.00    Median :  6.00      Median :199.0
##  Mean   :20.31255   Mean   : 21.16    Mean   : 13.56      Mean   :192.8
##  3rd Qu.:24.00000   3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0
##  Max.   :24.00000   Max.   :210.00    Max.   :143.00      Max.   :518.0
##  SedentaryMinutes    Calories
##  Min.   :   0.0   Min.   :   0
##  1st Qu.: 729.8   1st Qu.:1828
##  Median :1057.5   Median :2134
##  Mean   : 991.2   Mean   :2304
##  3rd Qu.:1229.5   3rd Qu.:2793
##  Max.   :1440.0   Max.   :4900
```

Statistical Interpretations:

1. Light active distance makes up the majority of total distance traveled (60.86%), which may indicate users didn’t do moderate or intense exercises 60.86% of the time.
2. Sedentary minutes makes up the majority of the total minutes (81.33%), which may indicate users didn’t move very much for a large portion of day.

sleep table:

```
sleep %>%
  select(TotalMinutesAsleep,
         TotalTimeInBed,
         PortionAsleep) %>%
  summary()
```

```
##  TotalMinutesAsleep TotalTimeInBed  PortionAsleep
##  Min.   : 58.0      Min.   : 61.0   Min.   :0.4984
##  1st Qu.:361.0      1st Qu.:403.8   1st Qu.:0.9118
##  Median :432.5      Median :463.0   Median :0.9426
##  Mean   :419.2      Mean   :458.5   Mean   :0.9165
##  3rd Qu.:490.0      3rd Qu.:526.0   3rd Qu.:0.9606
##  Max.   :796.0      Max.   :961.0   Max.   :1.0000
```

Statistical Interpretations:

1. On average, users were asleep 91.65% of their time in bed. This could be a good sign to market the product as the app could manage and maintain sleep time very well.

hourly__calories__sum table:

```
hourly_calories_sum %>%
  select(avg_calories) %>%
  summary()
```

```
##   avg_calories
##  Min.   : 67.54
##  1st Qu.: 80.68
##  Median :102.85
##  Mean   : 97.50
##  3rd Qu.:113.82
##  Max.   :123.49
```

Statistical Interpretations:

1. Average calories burned by hour range from 67.54 to 123.49, which indicates some hours users were on the move and some hours users were sedentary. This may inform the best hours to market the product.
2. Average calories burned by hour is 97.50. This is 4.23% of the average total calories burned per day. Burning calories is a consistent and not a speedy process in that it takes time.

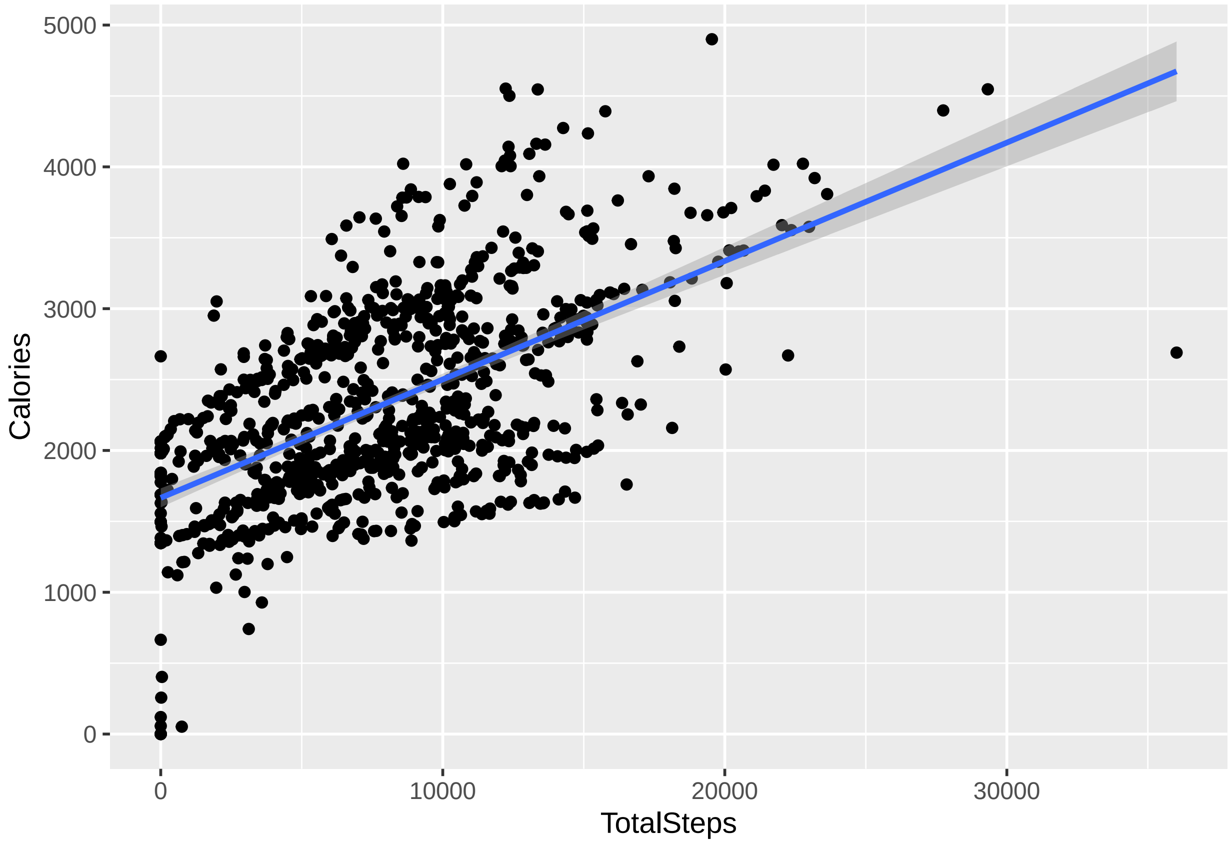
Step 5: Share

Let's visualize the data by plotting some explorations.

Chart 1:

```
ggplot(data=activity, aes(x=TotalSteps, y = Calories)) +
  geom_point() +
  stat_smooth(method=lm)
```

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



Findings and Implications:

1. There is a positive relationship between total steps taken and calories.
2. Some outliers existed when the total steps and calories are near zero. Another existed around 360000 total steps and 2700 calories. Some reasons might be human errors and miscalcuations.
3. The app can be marketed as beginner friendly in that one doesn't need to take a large number of steps to burn calories and get motivated.

Chart 2:

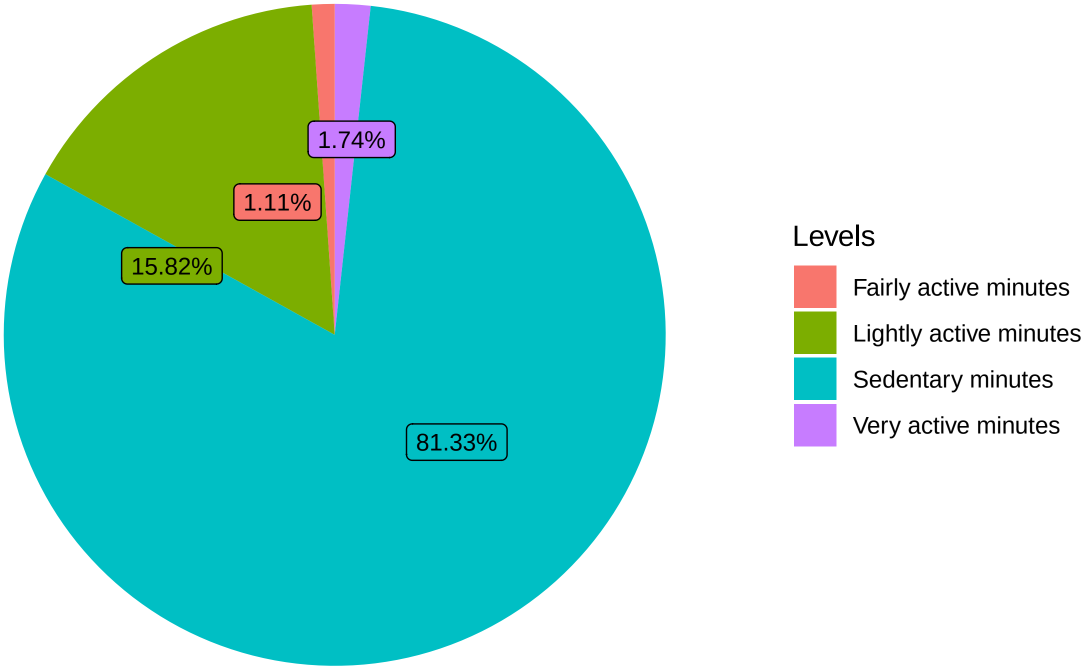
```
# creating variables for different durations of activity level
very_active_mins <- mean(activity$VeryActiveMinutes)/mean(activity$TotalMinutes)
fairly_active_mins <- mean(activity$FairlyActiveMinutes)/mean(activity$TotalMinutes)
lightly_active_mins <- mean(activity$LightlyActiveMinutes)/mean(activity$TotalMinutes)
```

```r
sedentary_mins <- mean(activity$SedentaryMinutes)/mean(activity$TotalMinutes)

# create a new table to organize data for creating a pie chart
percentage <- data.frame(
  Levels = c("Very active minutes", "Fairly active minutes", "Lightly active minutes",
             "Sedentary minutes"),
  value = c(very_active_mins, fairly_active_mins, lightly_active_mins, sedentary_mins)
)

# Set the positions for ggrepel
percentage <- percentage %>%
  mutate(csum = rev(cumsum(rev(value))),
         pos = value/4 + lead(csum, 1),
         pos = if_else(is.na(pos), value/2, pos))

ggplot(percentage, aes(x = "", y=value, fill=Levels)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  theme_minimal()+
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_blank(),
    panel.border = element_blank(),
    panel.grid = element_blank(),
    axis.ticks = element_blank(),
    plot.title = element_text(size=14, face="bold")
    ) +
  geom_label_repel(data = percentage,
                   aes(y = pos, label = percent(value)),
                   size = 3, nudge_x = 0, show.legend = FALSE)
```



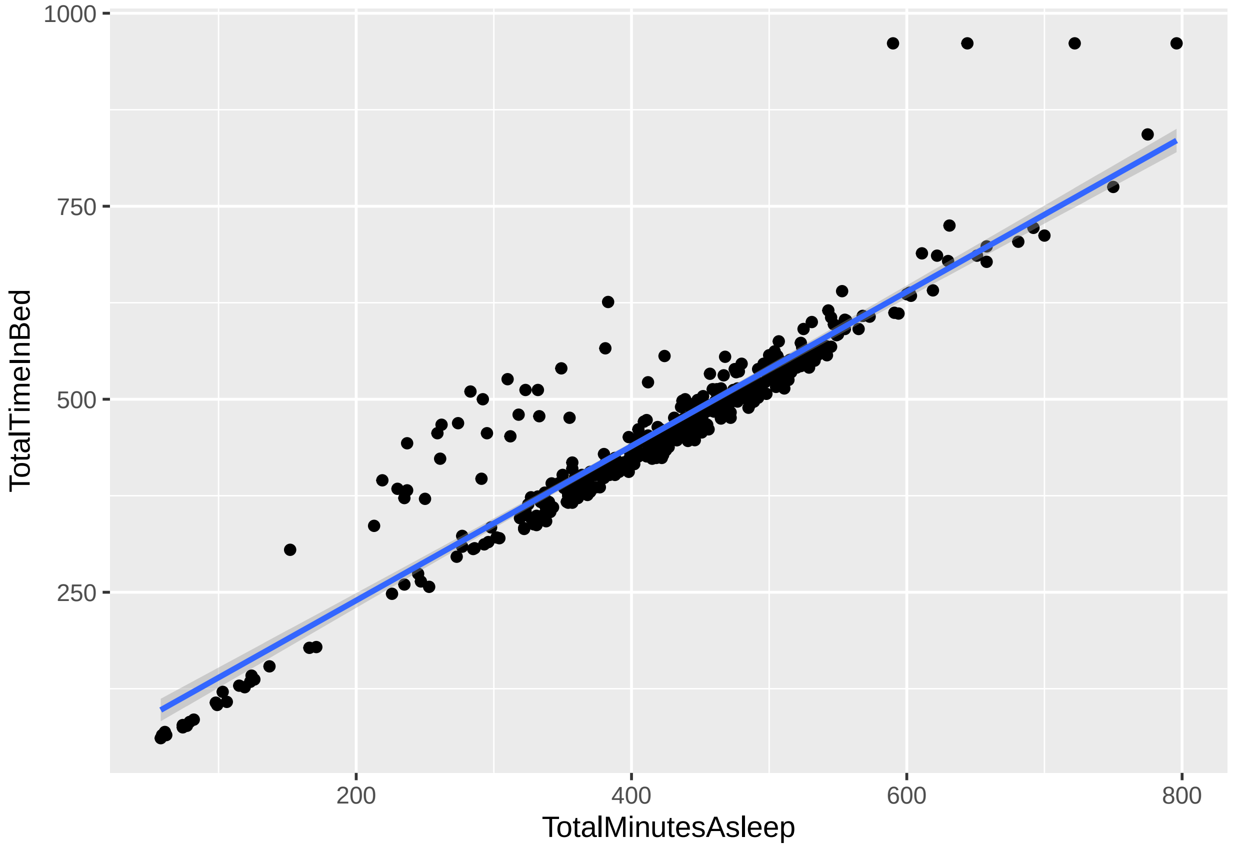
Findings and Implications:

1. On average, sedentary activity takes 81.33% of the users' time and fairly active and very active activities only takes up 2.85% of users' time.
2. The graph might indicate that most users are inactive throughout their days. This might be due to sedentary work during the day and sleeping during the night.
3. Within the app, implementing a point based system in which exercising would gain points while not exercising would deduct points, might motivate users and provide a selling point to potential customers.

Chart 3:

```
ggplot(data=sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
  geom_point() +
  stat_smooth(method=lm)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



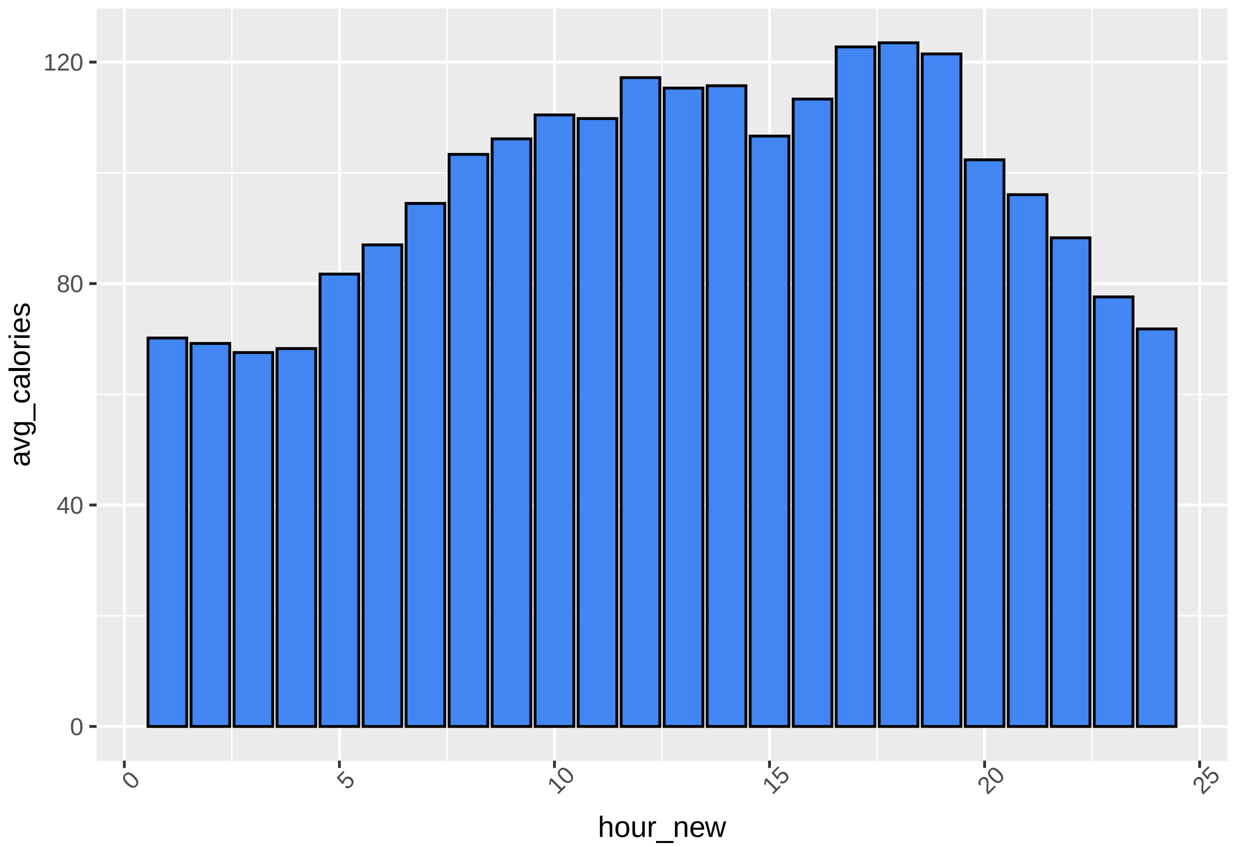
Findings and Implications:

1. There is a positive relationship between the total time in bed and total minutes asleep.
2. There is a group of users who spend a lot of time in bed and not sleeping and some users who spend a lot of time in bed and sleep a lot.

3. Videos instructing relaxation such as meditation and breathing exercise can be incorporated in the app and market as a selling point to prevent insomnia.

Chart 4:

```
ggplot(data = hourly_calories_sum) +
  geom_bar(aes(x = hour_new, y = avg_calories), stat="identity", color = "black",
           fill = "#4186f2") +
  theme(axis.text.x = element_text(angle = 45))
```



Findings and Implications:

1. There is a inverse U-shaped curve for calories burned throughout each day.
2. Most of the calories are burned in the afternoon and early evening, which may indicate that most exercises are done during the afternoon and early evening.
3. With this knowledge, the app should be marketed in the afternoon and early evening in which most potential consumers are exercising. For example, ads can be broadcasted on TV or music streaming services to target new consumers.

Step 6: Act

Conclusion:

Now that the analysis has been done and the visualizations have been shown with the findings and implications, it is time to revisit the business questions and give recommendations to the executives.

1. What are some trends in smart device usage?

- Sedentary time makes up 81.33% of total time.
- As time in bed increases, total minutes in bed increases. However, two groups of outliers exist when time in bed is greater than time asleep.
- The number of burned calories peaked in the late afternoon.

2. How could these trends apply to Bellabeat customers?

- Customers remain sedentary most of the day.
- Some customers don't sleep well.
- Customers are mostly active during the afternoon and early evening.

3. How could these trends help influence Bellabeat marketing strategy?

- Bellabeat should market a point-based reinforcement system in its app to attract potential customers who are looking for self-management of their activities or wanting to reduce their sedentary time.
- The app should also include video service like meditation or breathing exercise videos for customers who are looking to improve their sleep.
- Bellabeat should advertise the fitness tracker and its app during late afternoon in which most people are exercising and burning calories. This would match the demand of potential customers looking to buy a fitness trackers with an app that has many beneficial features.