Case study

Jacob Pangilinan

2022-06-14

1.Summary

- Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company

2.Ask

2.1 Questions for the analysis

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

2.2 Business task

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

3. Prepare Phase

3.1 Dataset used

- The data source used for our case study is FitBit Fitness Tracker Data. This dataset is stored in Kaggle.

3.2 Accessibility and privacy of data

- Verifying the metadata of our dataset we can confirm it is open-source. Allowing us to copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

3.3 Information about our dataset

- This dataset generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Individual reports can be parsed by export session ID (column A) or timestamp (column B). Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences. .

3.4 Data Organization and verification

- The dataset includes 18 CSV documents. Each document represents different quantitative data tracked by Fitbit.

4. Process

4.1 Loading Packages

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.1 ──

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

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(readr)
```

4.2 Importing Datasets

```
daily_activity <- read.csv("C:\\Users\\Jacob\\Desktop\\coursera\\dailyActivit
y_merged.csv")
hourly_calories <- read.csv("C:\\Users\\Jacob\\Desktop\\coursera\\hourlyCalor
ies_merged.csv")
hourly_intensities <- read.csv("C:\\Users\\Jacob\\Desktop\\coursera\\hourlyIn
tensities_merged.csv")
sleep_day <- read.csv("C:\\Users\\Jacob\\Desktop\\coursera\\sleepDay_merged.c
sv")
```

4.3 Data Cleaning

- Upon investigation of the data I noticed that some of that dates in the datasets were formated differently so I converted them to the same date time format for merging later on.

```
hourly_intensities$ActivityHour=as.POSIXct(hourly_intensities$ActivityHour, f
ormat="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
hourly_intensities$time <- format(hourly_intensities$ActivityHour, format = "
%H:%M:%S")
hourly_intensities$date <- format(hourly_intensities$ActivityHour, format = "
%m/%d/%y")
```

```
hourly_calories$date <- as.Date(hourly_calories$ActivityHour)
hourly_calories$time <- format(hourly_calories$ActivityHour, format = "%H:%M:
%S")

daily_activity$ActivityHour=as.POSIXct(daily_activity$ActivityDate, format="%
m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
daily_activity$date <- format(daily_activity$ActivityDate, format = "%m/%d/%y
")

sleep_day$SleepDay=as.POSIXct(sleep_day$SleepDay, format="%m/%d/%Y %I:%M:%S %
p", tz=Sys.timezone())
sleep_day$date <- format(sleep_day$SleepDay, format = "%m/%d/%y")
sleep_day$timedate <- format(sleep_day$time, format = "%H:%M:%S")
library(dplyr)
sleep_day = select(sleep_day, -6)
```

- Now that the datasets have the same date columns it's time to check for duplicates within the data sets.To begin I determined the amount of distinct variables in each data set.

```
n_distinct(daily_activity$Id)

## [1] 33

n_distinct(sleep_day$Id)

## [1] 24

n_distinct(hourly_calories$Id)

## [1] 33

n_distinct(hourly_intensities$Id)

## [1] 33
```

- the findings from the initial exploration tell us that there are 33 participants in the activity, calories and intensities data sets, 24 in the sleep and 8 in the weight data set. Now that I know the amount of distinct variables I can determine the amount of duplicate variables.

```
sum(duplicated(daily_activity))

## [1] 0

sum(duplicated(sleep_day))

## [1] 3

sum(duplicated(hourly_calories))

## [1] 0

sum(duplicated(hourly_intensities))
```

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

- when checking for duplicates I found that the sleep_day data set had 3 duplicate values.

```
library(tidyverse)
sleep_day <- sleep_day %>%
  distinct() %>%
  drop_na()

sum(duplicated(sleep_day))

## [1] 0
```

- Now there are no duplicates in my data sets

5. Analyze/share

5.1 summarizing the datasets

```
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes) %>%
  summary()

##    TotalSteps    TotalDistance    SedentaryMinutes VeryActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :1057.5   Median :  4.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   : 21.16
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.: 32.00
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :  0.00      Min.   :  0.0
##  1st Qu.:  0.00      1st Qu.:127.0
##  Median :  6.00      Median :199.0
##  Mean   : 13.56      Mean   :192.8
##  3rd Qu.: 19.00      3rd Qu.:264.0
##  Max.   :143.00      Max.   :518.0

sleep_day %>%
  select(TotalMinutesAsleep,
         TotalSleepRecords,
         TotalTimeInBed) %>%
  summary()

##  TotalMinutesAsleep TotalSleepRecords TotalTimeInBed
##  Min.   : 58.0      Min.   :1.00      Min.   : 61.0
```

```
##  1st Qu.:361.0      1st Qu.:1.00      1st Qu.:403.8
##  Median :432.5      Median :1.00      Median :463.0
##  Mean   :419.2      Mean   :1.12      Mean   :458.5
##  3rd Qu.:490.0      3rd Qu.:1.00      3rd Qu.:526.0
##  Max.   :796.0      Max.   :3.00      Max.   :961.0
```

```
hourly_calories %>%
  select(Calories) %>%
  summary()
```

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

```
hourly_intensities %>%
  select(TotalIntensity,
         AverageIntensity) %>%
  summary()
```

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

Findings

- The average sedentary mintutes were around 991 minutes or 16hours per day. This is a large portion of the day that Bellabeat could try reducing. *Most users were lightly active which is lower then the recommendation set by the U.S. Department of Health and Human Services which recommends 150 minutes a week of moderate-intensity exercise.
- On the average, participants sleep 1 time for 7 hours.
- Average total steps per day are 7638 and roughly an average of 5.5 miles per day.

Identifying Target users

Identifying user who are less active will give us a good starting point on which demographic Bellabeat can market towards. I have categorize the activity rate by as follows based on the following article: Medicinenet

```
* Sedentary: Less than 5,000 steps daily
* Low active: About 5,000 to 7,499 steps daily
* Somewhat active: About 7,500 to 9,999 steps daily
```

```
* Active: More than 10,000 steps daily
* Highly active: More than 12,500 steps daily
```

- To start I will find the average total steps and calories from the daily activity data set.

```
daily_steps <- daily_activity %>%
  group_by(Id) %>%
  summarise(mean_daily_steps = mean(TotalSteps), mean_daily_calories = mean(C
alories))

head(daily_steps)
```

```
## # A tibble: 6 × 3
##           Id mean_daily_steps mean_daily_calories
##        <dbl>            <dbl>               <dbl>
## 1 1503960366           12117.               1816.
## 2 1624580081            5744.               1483.
## 3 1644430081            7283.               2811.
## 4 1844505072            2580.               1573.
## 5 1927972279             916.               2173.
## 6 2022484408           11371.               2510.
```

- Now that we have the average total steps for each unique user we can go ahead and classify each user into user types listed above.

```
library(dplyr)
user_type <- daily_steps %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7499 ~ "low active",
    mean_daily_steps >= 7500 & mean_daily_steps < 9999 ~ "somewhat active",
    mean_daily_steps >= 10000 ~ "active"
  ))

head(user_type)
```

```
## # A tibble: 6 × 4
##           Id mean_daily_steps mean_daily_calories user_type
##        <dbl>            <dbl>               <dbl> <chr>
## 1 1503960366           12117.               1816. active
## 2 1624580081            5744.               1483. low active
## 3 1644430081            7283.               2811. low active
## 4 1844505072            2580.               1573. sedentary
## 5 1927972279             916.               2173. sedentary
## 6 2022484408           11371.               2510. active
```

- once we have the user type we can calculate what percentage each user is of the entire data set by turning them into percentages.

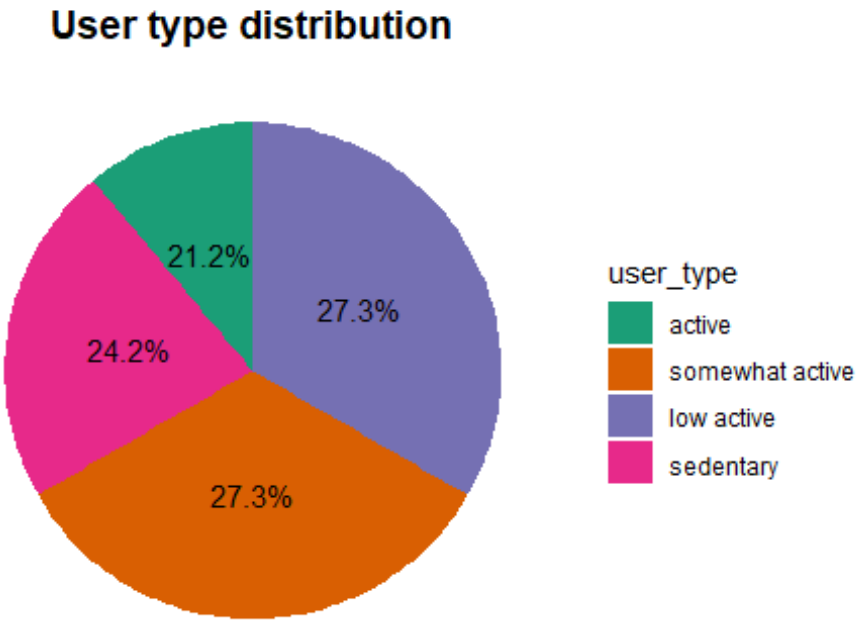
```
library(dplyr)
user_type_percent <- user_type %>%
  group_by(user_type) %>%
```

```
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

user_type_percent$user_type <- factor(user_type_percent$user_type , levels =
c("active","somewhat active", "low active", "sedentary"))
```

Visualization

```
library(ggplot2)
bp<- ggplot(user_type_percent, aes(x="", y=labels, fill=user_type))+
geom_bar(width = 1, stat = "identity")
pie <- bp + 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")) +
  scale_fill_brewer(palette="Dark2") +
  theme(axis.text.x=element_blank())+
  geom_text(aes(label = labels),position = position_stack(vjust = 0.5))+
  labs(title="User type distribution")
pie
```

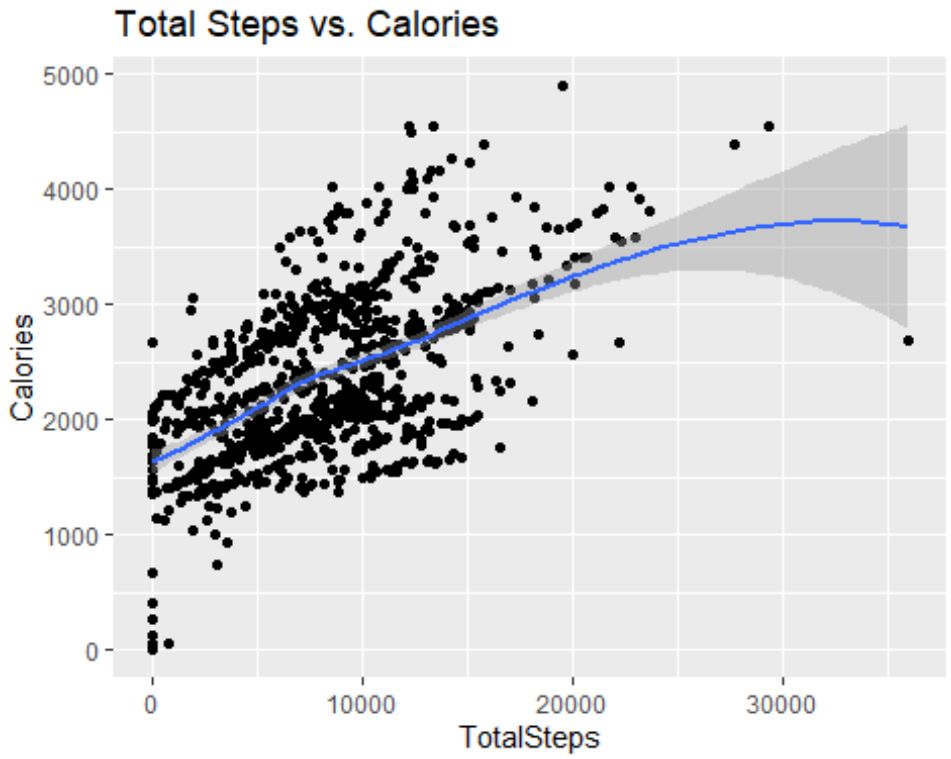


Findings

- According to the pie chart we can see that about 54% of the user types fall into a somewhat active or low active categories. while 24.2% fall into the sedentary category.
- If Bellabeat wanted more engagement on their app they could target the 24% of sedentary users and even the 27% of low active users.
- By targeting these two user types Bellabeat could increase their app usage as more people use the step tracker for longer periods of time.
- Now that we know what the target audience is, we can dive into the data to see how we might use the information to get these users to engage with the app longer.

```
library(ggplot2)
ggplot(daily_activity, mapping=aes(x=TotalSteps, y= Calories)) +
  geom_point() +geom_smooth() +labs(title="Total Steps vs. Calories")

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



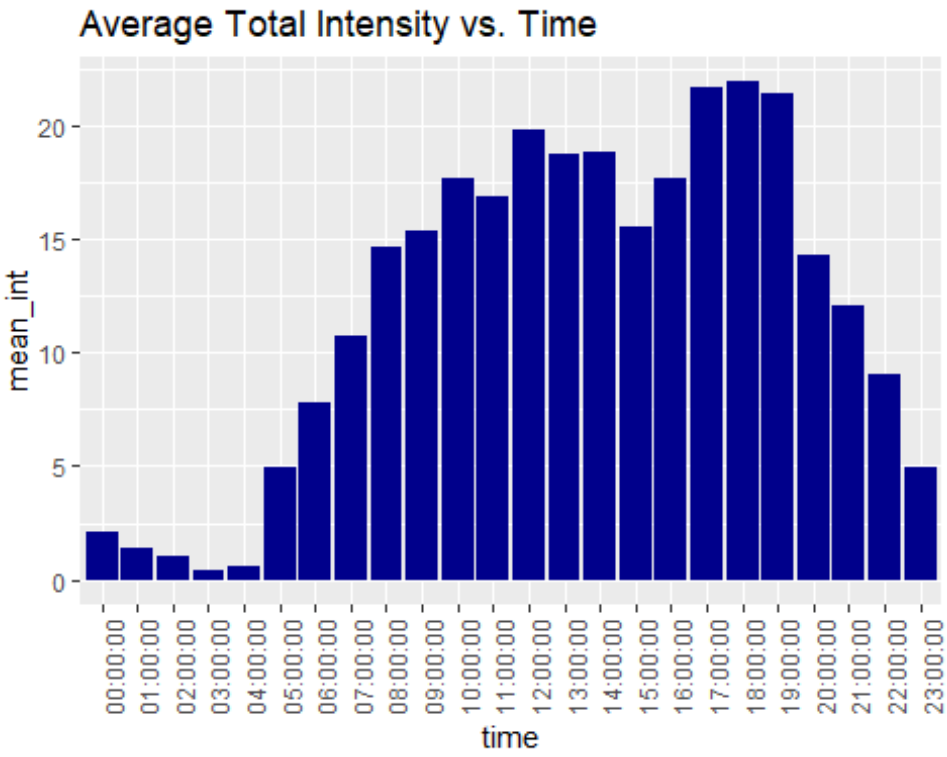
Findings

- In this graph we see that there is a positive correlation between the total steps and calories burned. This isn't a new discovery but Bellabeat can use this information to encourage people to walk for longer in order to burn more calories. This has a simultaneous effect of keeping people using the app longer.It's important to remember that correlation does not equal causation.
- Making a goal of 10,000 steps a day boosts your heart health, helps you focus better, provides more energy, and strengthens your lungs, bones, and muscles.
- Next we can look to see at what time people are working out the most. We can do that by using the hourly intensities data set

```
intensities_new <- hourly_intensities %>%
  group_by(time) %>%
  summarise(mean_int = mean(TotalIntensity))

ggplot(data=intensities_new, aes(x=time, y=mean_int)) + geom_histogram(stat =
"identity", fill='darkblue') + theme(axis.text.x = element_text(angle = 90))
+
  labs(title="Average Total Intensity vs. Time")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Findings

- In this graph we see that people are most active from 5pm to 7pm. This is most likely due to the fact that people get off work around this time and can exercise. IT would be a good idea to have the Bellabeat app remind and motivate users to go for a run or walk during these peak times.

6. Conclusions

- We found that most of the people work out between the times of 5pm to 7pm. Bellabeat could increase engagement of their app by reminding users during those peak times to go on a walk or run.
- We classified users into 4 categories and through our summary statistics found that the average of users walk more than 7,500 steps daily. Bellabeat can encourage customers to reach the recommend daily steps of 8,000 by the CDC. One way of doing this would be sending them alarms if they haven't reached the 8,000 daily steps.
- Bellabeat could also send out articles on the app that explain the health benefits of reaching that goal. As CDC explains the more steps you walk the lower is the mortality rate. We also saw a positive correlation between steps and calories.

6.1 Recommendations

- Based on the conclusions drawn from the data I recommend that Bellabeat:
    1. create a daily steps notifications starting at 5pm.

2. send out articles on the health benefits of being active.
3. Set up a rewards program for those who reach their step goals.