Bella.Beats_Case

Abdallah Zubedi

2023-05-22

Introduction

BellaBeats offers fitness tracker products such as wrist trackers and watches named Time. The task at hand is to answer the following questions:

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?

To accomplish this, we need data from BellaBeat or data that is like BellaBeat. in this case, Fitbit data was chosen for its similarity to BellaBeat, and it is publicly available to use.

Source of data: Fitbit Data

The Data

1. There may be sampling bias present as the method of participant selection is unclear. It is possible that participants who are willing to make their activity data public are more likely to be heavy users of FitBit.

2. Bellabeats target market is women.

3. The data is from 2016, which can be considered outdated.

4. The sample size is made up of 33 users. This is not a great sample of the population to locate significant patterns and analyze. The data for each user recurs multiple times depending on the dates, which provides us some value on analyzing how they use the products with date/time.

5. Fitbits data has a total of 10+ files. The main files used were dailyActivity_merged and SleepDay_merged.

Tools used:

- SQL to prepare, clean, organize and descriptively analyze data.
- Tableau for Visualization

Setting up

Packages

The following packages were used after organising data in SQL.

```
#install.packages("dplyr")
#install.packages("janitor")
#install.packages("tidyverse")
#install.packages("ggplot2")

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

library(janitor)

##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ ggplot2   3.4.1     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.1
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0

## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force
all conflicts to become errors

library(ggplot2)
```

Files

```
Processed <- read.csv("tired_table_test.csv")
```

Data cleaning

Removing Rows with null and duplicate values

Confirming if column LoggedActivitiesDistance has any data.

```
SELECT *

FROM bella_beats.dailyactivity_cleantable

WHERE LoggedActivitiesDistance > 0
```

No rows were returned.

Removing LoggedActivitiesDistance column

```
ALTER TABLE dailyactivity_cleantable

DROP COLUMN LoggedActivitiesDistance;
```

Removed Trackerdistance as it has the dupicate data as total distance.

```
ALTER TABLE daily_activity_new

DROP COLUMN Trackerdistance;
```

Order by Date and Id

```
SELECT *

FROM dailyactivity_cleantable

order by ActivityDate AND Id
```

Confirming len of column ID

```
SELECT MAX(LENGTH(Id)) AS max_length

FROM dailyactivity_cleantable;
```

10 was max meaning the Id was correct.

Counting distinct ID

```
SELECT COUNT(distinct id)

 FROM bella_beats.dailyactivity_merged;
```

I only got a count of 6. This was suspicious, so I used Excel to look at the count of unique values which were 33.

I changed the format of the Id column to text as the int format got an error when importing, stating it was out of range. The count of ids is now 33.

Changing date format from text to date.

```
UPDATE dailyactivity_merged

SET ActivityDate = STR_TO_DATE(ActivityDate, '%m/%d/%Y')
```

Added time_sleeping and time_in_bed columns to the dailyactivity.

```
ALTER TABLE daily_activity

ADD total_minutes_sleep INT,

    total_time_in_bed INT;
```

Added data from sleepday_merged to match id and activitydate in the dailyactivity table using JOIN.

```
UPDATE `bella_beats`.`dailyactivity` as temp1

JOIN sleepday_merged as temp2

ON temp1.id = temp2.id AND temp1.ActivityDate = temp2.SleepDay

SET temp1.total_minutes_sleep = temp2.TotalMinutesAsleep,

    temp1.total_time_in_bed = temp2.TotalTimeInBed;
```

Removed null values that did not have matching IDs. This was mostly in the sleepday column as it has 22 distinct ids while dailyactivity has 33.

```
CREATE TABLE daily_activity_filtered AS

SELECT *

FROM daily_activity

WHERE total_minutes_sleep IS NOT NULL AND total_time_in_bed IS NOT NULL;
```

Analyze

Descriptive analysis

```
## Analysis
descriptive_analysis <- Processed %>%

  group_by(Id) %>%

  summarise(

    max_TotalDistance = max(TotalDistance),

    min_TotalDistance = min(TotalDistance),

    max_Calories = max(Calories),

    min_Calories = min(Calories),

    max_diff = max(diff),

    min_diff = min(diff),

    day_of_week = first(day_of_week)

  ) %>%

  arrange (max_TotalDistance)

print(descriptive_analysis)
```

```
## # A tibble: 24 × 8
##        Id max_TotalDistance min_TotalDistance max_Calories min_Calories
max_diff
##     <dbl>             <dbl>             <dbl>        <int>        <int>
<int>
##  1 1.93e9              2.6               0.25         2638         2151
25
##  2 1.84e9              2.67              1.7          1763         1541
371
##  3 2.32e9              3.42              3.42         1804         1804
8
##  4 6.78e9              3.7               0.92         2507         2127
25
##  5 7.01e9              3.75              3.1          2225         2076
3
##  6 8.79e9              5.35              0.78         3101         1934
35
##  7 4.45e9              6.11              0.52         2499         1212
```

```
76
##  8 4.32e9              7.28              0.01         2367          257
41
##  9 2.03e9              7.71              0.16         1926         1141
61
## 10 4.02e9              8.43              1.42         3879         2704
58
## # ℹ 14 more rows
## # ℹ 2 more variables: min_diff <int>, day_of_week <chr>
```

From the descriptive analysis, you will notice that it does not matter if a user had more steps than the other, as calories differentiated. This is likely to do with their dietary plans and metabolism.

```
SELECT ROUND(AVG(TotalSteps)) AS mean_steps, ROUND(AVG(TotalDistance)) AS
mean_distance, ROUND(AVG(Calories)) AS mean_calories,

ROUND(MAX(TotalSteps)) AS max_steps, ROUND(MAX(TotalDistance)) AS
max_distance, ROUND(MAX(calories)) AS max_calories

FROM dailyactivity_merged
```

mean_steps	mean_distance	mean_calories	max_steps	max_distance	max_calories
7638	5	2304	36019	28	4900

Minimums started from 0,1,2... this has no value apart from showing that the users might have not recorded their steps/calories on that day.

Day of max steps

```
SELECT
    DAYNAME(ActivityDate) AS day_of_week,
    ROUND(AVG(TotalSteps)) AS mean_steps,
    ROUND(AVG(TotalDistance)) AS mean_distance,
    ROUND(AVG(Calories)) AS mean_calories,
    ROUND(MAX(TotalSteps)) AS max_steps,
    ROUND(MAX(TotalDistance)) AS max_distance,
    ROUND(MAX(Calories)) AS max_calories

FROM
    dailyactivity_merged
WHERE
    TotalSteps > 0
GROUP BY
    DAYNAME(ActivityDate)
```

```
HAVING

    MAX(TotalSteps) = (
        SELECT
            MAX(TotalSteps)
        FROM
            dailyactivity_merged
        WHERE
            TotalSteps > 0
    );
```

The maximum steps were on a Sunday.

Viewing totals by day of week for total steps, total calories, and total distance columns

```
SELECT DAYNAME(ActivityDate) AS Day_of_week, SUM(TotalSteps) AS total_steps,
SUM(calories) AS total_calories, ROUND(SUM(TotalDistance)) AS Total_distance

FROM dailyactivity_merged

GROUP BY Day_of_week

ORDER BY total_steps;
```

Results

Day_of_week	total_steps	total_calories	Total_distance
Sunday	838921	273823	608
Monday	933704	278905	666
Friday	938477	293805	669
Saturday	1010969	292016	726
Thursday	1088658	323337	781
Wednesday	1133906	345393	823
Tuesday	1235001	358114	886

It is noticeable that Sunday is the end of the week while Monday is the beginning of the work week, these two days have the least activity intensities. It can be due to the users trying to relax on Sunday and warming up on a Monday. (Assumption)

The frequency of day of the week users exercises most and least often.

```
WITH day_counts AS (

  SELECT day_of_week, COUNT(*) AS count
  FROM tired_table_test
  GROUP BY day_of_week
)
```

```sql
SELECT
  day_of_week AS mode_day,
  count AS mode_count
FROM day_counts

WHERE count = (
  SELECT MAX(count)
  FROM day_counts
)

UNION ALL
SELECT
  day_of_week AS least_day,
  count AS least_count
FROM day_counts
WHERE count = (
  SELECT MIN(count)
  FROM day_counts

)
```

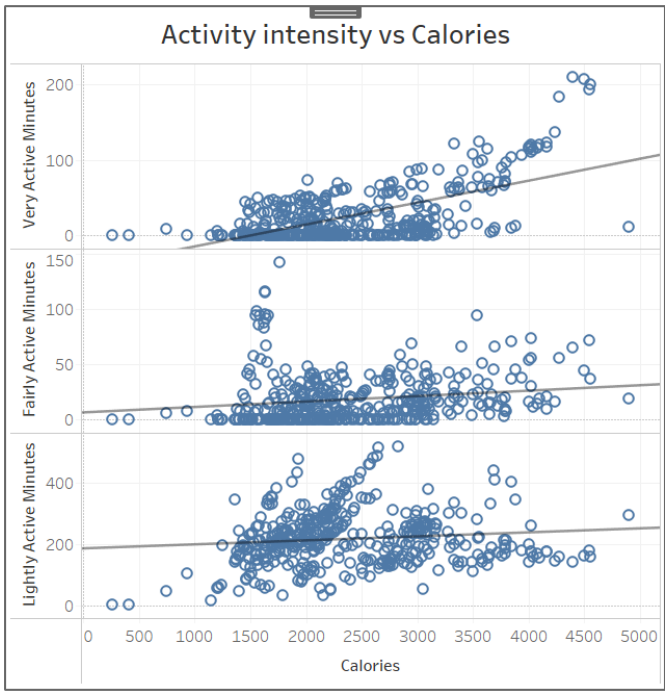
mode_day	mode_count
Wednesday	66
Monday	46

Wednesday is the mode with 66 counts, while Monday had the least count with 46.

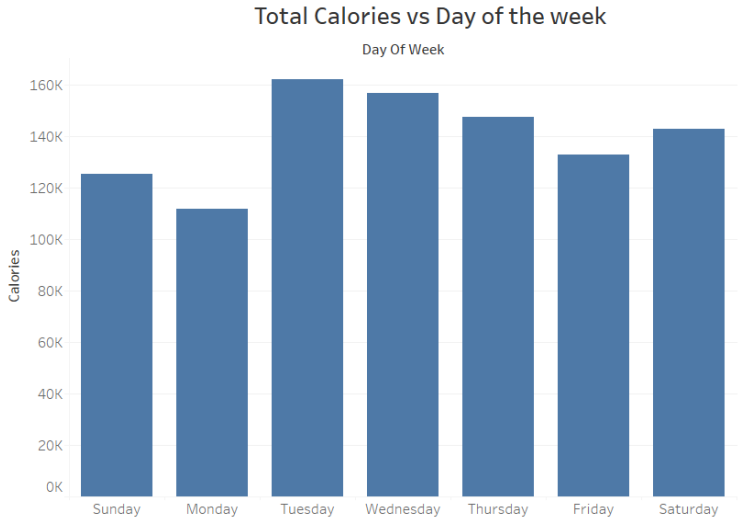
This shows that the users have more activity during the weekdays compared to the weekend and the start of the week. It does not result in more intensity levels, but the sample population is more likely to exercise on the weekdays than on weekends.

Visualization

I used Tableau for the visualizations.

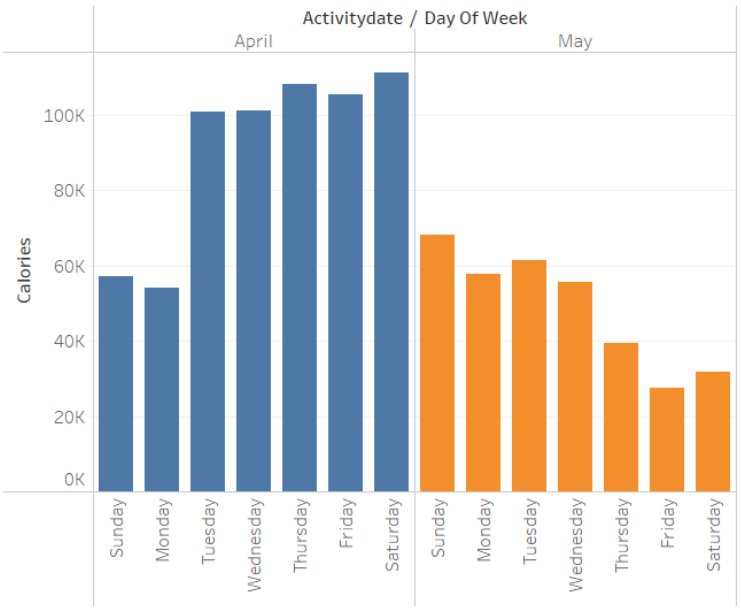


There is a close relationship with very active intensity and calories, with an R-square value of 0.37 while the others were lower gradually.



The bar graph clearly shows when most calories are burned, which is Tuesday Wedesday and a Thursday.

Bar graph Calories burned in day of the week and month



There are 2 months of data from the source files. May and April. As the bar graph shows, there are more calories burned in April compared to May, this may be due to summer being close and the workout culture when getting to summer is usually to burn as much calories as possible after the winter months which are usually for mass gaining. For May, most workouts are done more in the beginning of the week, to Wednesday. This is a difference in the days the users usually workout the most in April, which are the weekdays to Saturday.

Conclusion

With a small population size, it is usually harder to assess patterns and trends as accuracy is decreased. In this case. From the little data we have, some insights have appeared.

- Most user's workout on the weekdays instead of the weekends.
- Most user's workout way more during the winter season to prepare for summer.
- There are users who work out every day compared to others.
- Manual reporting is tedious for the users and can develop more dirty data that is unreliable.

Recommendations

- Marketing strategies can be targeted towards the seasons and days of the week, such as the weekdays in April and weekends in May.
- Automating tracking records should be a priority, if the ease of use is increased, there would be accurate data being recorded while the users would also be more inclined to use the products.

- There are two groups of users, ones who workout more frequently and others who do not. Marketing campaigns can be target for both these users separately, with users who workout everyday receiving notifications a premium package that can highlight more details into their vitals and workout sessions, while the users who do not workout everyday can receive prompts of encouragement such as a score to achieve at a certain date.

- Personalization would work great in terms of suggesting to the user the type of exercise and diet they would need to achieve a specific outcome. This can be segmented by understanding the metabolisms of the users. clustering these users would be fruitful for the future as there would be standardization on how to communicate with them inr regards to their workouts.

Limitations

- Data obtained was from Fitbit, this is usually a product that is put on the wrist. BellaBeats has multiple products meaning this data would only fit the BellaBeat products that are used while on the wrist.

- Finding data with the same geographic location is important, as they may have different seasons and workout culture.