‘Bellabeat’ Product Analysis: How Can A Wellness Company Play It Smart?

Jhermien Paul Alejandria
Data Analyst at Royal Caribbean • Transforming Big Data into Actionable Insights & Strategies PH

March 6, 2023

For this case study, I take on the role of a junior data analyst working in the marketing and analytics team at *Bellabeat*, a high-tech manufacturer of health-focused products for women. Its founder believes that analyzing user data could help unlock new growth opportunities for the company. Therefore, my team is directed to gain insight into how users are utilizing our products in order to present high-level recommendations for Bellabeat’s marketing strategy. I am specifically assigned to focus on our best-selling product, the Leaf smart device.

To accomplish my task, I will follow the steps of the data analysis process: **Ask, Prepare, Process, Analyze, Share**, and **Act.**

Tools: Spreadsheet (Microsoft Excel), R (RStudio), Tableau

Visualization: Go to my Tableau profile.

Documentation: Visit my GitHub profile.

Step 1: Ask

I. Context

In 2016, Bellabeat launched the following products that empower women with knowledge about their health:

- **Leaf** – a smart tracker worn as a bracelet/necklace which monitors users' activity and sleep;
- **Time** – a smart watch that tracks users' pulse rate; and
- **Spring** – a smart water bottle that traces users' daily water intake.

These smart devices are connected to the **Bellabeat app**, which provides users with their overall health data in real-time.

Bellabeat also has a subscription-based membership program that gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness, based on their lifestyle and goals.

II. Task

I will analyze how users are utilizing the Leaf smart device. This will involve finding relevant patterns and trends in whatever information can be found in the dataset.

Afterward, I will propose recommendations for Bellabeat's marketing strategy.

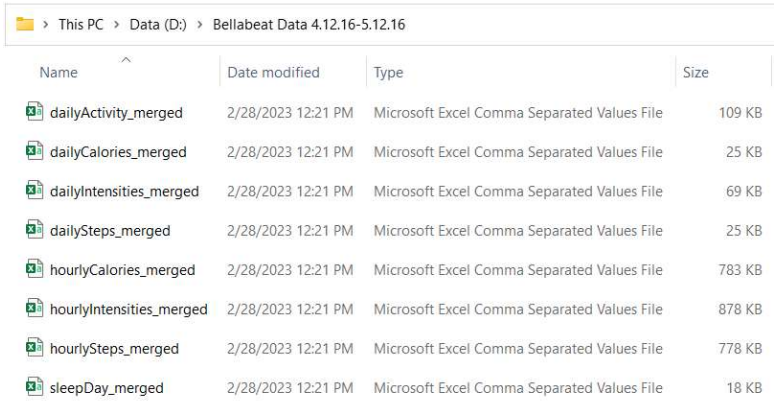
Step 2: Prepare

I. Access the data

My team was provided with Bellabeat's product user data from April 12 to May 12 of 2016. Access the dataset through this link.

II. Collect the data

The dataset consists of many CSV files with users' health data in various timeframes.



I only downloaded 8 files pertaining to activity and sleep data in daily and hourly intervals to concentrate on larger trends and patterns in Leaf device usage.

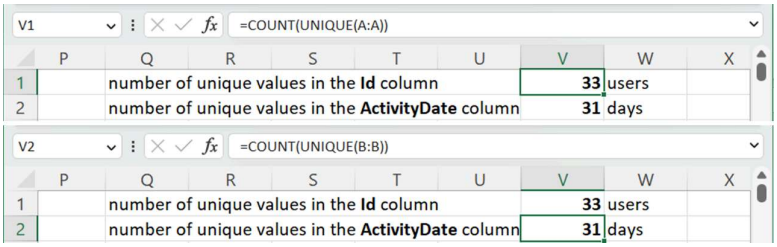
III. Examine the data

I inspected the files in Microsoft Excel and found that each file holds the following information:

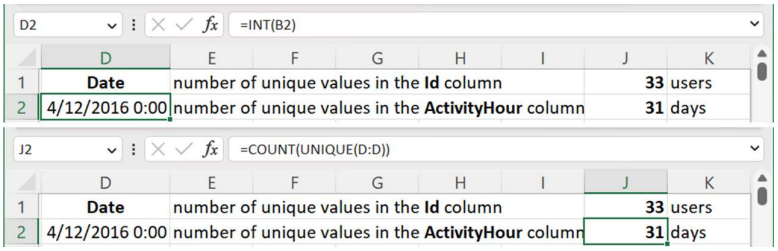
	Table	Attributes		Observations	Users	Days
1	dailyActivity_merged.csv	Id	Unique ID for each activity record	940	33	31
		ActivityDate	Date value in mm/dd/yyyy format			
		TotalSteps	Total number of steps taken			
		TotalDistance	Total kilometers tracked			
		TrackerDistance	Total kilometers tracked by the device			
		LoggedActivitiesDistance	Total kilometers from logged activities			
		VeryActiveDistance	Kilometers travelled during very active activity			
		ModeratelyActiveDistance	Kilometers travelled during moderate activity			
		LightActiveDistance	Kilometers travelled during light activity			
		SedentaryActiveDistance	Kilometers travelled during sedentary activity			
		VeryActiveMinutes	Total minutes spent in very active activity			
		FairlyActiveMinutes	Total minutes spent in moderate activity			
		LightlyActiveMinutes	Total minutes spent in light activity			
		SedentaryMinutes	Total minutes spent in sedentary activity			
		Calories	Total estimated energy expenditure (in kilocalories)			
2	dailyCalories_merged.csv	Id	Unique ID for each activity record	940	33	31
		ActivityDay	Date value in mm/dd/yyyy format			
		Calories	Total estimated energy expenditure (in kilocalories)			
3	dailyIntensities_merged.csv	Id	Unique ID for each activity record	940	33	31
		ActivityDay	Date value in mm/dd/yyyy format			
		VeryActiveMinutes	Total minutes spent in very active activity			
		FairlyActiveMinutes	Total minutes spent in moderate activity			
		LightlyActiveMinutes	Total minutes spent in light activity			
		SedentaryMinutes	Total minutes spent in sedentary activity			
		VeryActiveMinutes	Total minutes spent in very active activity			
		FairlyActiveMinutes	Total minutes spent in moderate activity			
		LightlyActiveMinutes	Total minutes spent in light activity			
		SedentaryMinutes	Total minutes spent in sedentary activity			
4	dailySteps_merged.csv	Id	Unique ID for each activity record	940	33	31
		ActivityDay	Date value in mm/dd/yyyy format			
		StepTotal	Total number of steps taken			
5	hourlyCalories_merged.csv	Id	Unique ID for each activity record	22,099	33	31
		ActivityHour	Date and hour value in mm/dd/yyyy hh:mm:ss format			
		Calories	Total estimated energy expenditure (in kilocalories)			
6	hourlyIntensities_merged.csv	Id	Unique ID for each activity record	22,099	33	31
		ActivityHour	Date and hour value in mm/dd/yyyy hh:mm:ss format			
		TotalIntensity	Value calculated by adding all the minute-level intensity values that occurred within the hour			
		AverageIntensity	Average intensity state exhibited during that hour (TotalIntensity for that ActivityHour divided by 60)			
7	hourlySteps_merged.csv	Id	Unique ID for each activity record	22,099	33	31
		ActivityHour	Date and hour value in mm/dd/yyyy hh:mm:ss format			
		StepTotal	Total number of steps taken			
8	sleepDay_merged.csv	Id	Unique ID for each sleep record	413	24	31
		SleepDay	Date value in mm/dd/yyyy format			
		TotalSleepRecords	Number of recorded sleep periods for that day, includes naps > 60min			
		TotalMinutesAsleep	Total number of minutes classified as being "asleep"			
		TotalTimeInBed	Total minutes spent in bed, including asleep, restless, and awake, that occurred during a defined sleep record			

The description for each attribute is based on this metadata.

The number of users and days were determined by counting the number of unique values in the **Id** and **ActivityDate/ActivityDay** columns, respectively.



For files with **ActivityHour** column (DATE-TIME format), I temporarily inserted a column named **Date** and used the INT function to extract only the DATE portion. Then, I counted the number of unique values in the **Date** column.



My manager confirmed that only 33 users consented to share their personal data for data analysis purposes.

It appears that **dailyCalories_merged.csv, dailyIntensities_merged.csv,** and **dailySteps_merged.csv** are mere subsets of **dailyActivity_merged.csv,** so I decided to delete them to avoid redundancy.

IV. Limitations of the data

- **Outdated data**: The data is from 2016, which is seven years ago. The usage patterns, user behaviors, and market trends might have significantly changed since then, and the findings may not be relevant to the current market and user base.
- **Narrow timeframe**: The data only covers a period of 31 days, which is a short period to make any significant conclusions about the usage of the product.
- **Small sample size**: The data only includes data from 33 users at most. This could limit the representativeness of the data, and the findings may not generalize to the wider population of Leaf users.
- **No demographic info**: The data lacks any demographic information about the users (such as age, race, income, etc.) which can provide additional insights into how different user groups are utilizing the product.

Step 3: Process

I. Import the data

I used R for data cleaning and wrangling. I started by installing and loading the required packages in RStudio.

```r
# install and load packages

install.packages(c("tidyverse", "lubridate", "dplyr", "tidyr",
                   "janitor","ggplot2", "waffle", "corrplot"))

library(tidyverse)  # for data manipulation and visualization
library(lubridate)  # for working with dates and times
library(dplyr)      # for data manipulation
library(tidyr)      # for data tidying and reshaping
library(janitor)    # for data cleaning and formatting
library(ggplot2)    # for data visualization
library(waffle)     # for creating waffle charts
library(corrplot)   # for creating correlation plots
```

Then, I imported the files as data frames.

```
# import files

daily_activity <- read_csv("D:/Bellabeat Data 4.12.16-
5.12.16/dailyActivity_merged.csv")
daily_sleep <- read_csv("D:/Bellabeat Data 4.12.16-
5.12.16/sleepDay_merged.csv")
hourly_calories <- read_csv("D:/Bellabeat Data 4.12.16-
5.12.16/hourlyCalories_merged.csv")
hourly_intensities <- read_csv("D:/Bellabeat Data 4.12.16-
5.12.16/hourlyIntensities_merged.csv")
hourly_steps <- read_csv("D:/Bellabeat Data 4.12.16-
5.12.16/hourlySteps_merged.csv")
```

Now that the data is ready, it's time to clean it!

II. Clean the data

1. I renamed the headers in each data frame using the **clean_names()** function to ensure they are syntactically valid (i.e., start with a letter/underscore and contain only letters/numbers/underscores).

```
# rename headers

daily_activity <- clean_names(daily_activity)
daily_sleep <- clean_names(daily_sleep)
hourly_calories <- clean_names(hourly_calories)
hourly_intensities <- clean_names(hourly_intensities)
hourly_steps <- clean_names(hourly_steps)
```

Look at the new headers:

```
# show headers

colnames(daily_activity)
colnames(daily_sleep)
colnames(hourly_calories)
colnames(hourly_intensities)
colnames(hourly_steps)
```

```
> colnames(daily_activity)
 [1] "id"                       "activity_date"
 [3] "total_steps"              "total_distance"
 [5] "tracker_distance"         "logged_activities_distance"
 [7] "very_active_distance"     "moderately_active_distance"
 [9] "light_active_distance"    "sedentary_active_distance"
[11] "very_active_minutes"      "fairly_active_minutes"
[13] "lightly_active_minutes"   "sedentary_minutes"
[15] "calories"
> colnames(daily_sleep)
[1] "id"                   "sleep_day"            "total_sleep_records"
[4] "total_minutes_asleep" "total_time_in_bed"
> colnames(hourly_calories)
[1] "id"            "activity_hour" "calories"
> colnames(hourly_intensities)
[1] "id"               "activity_hour"    "total_intensity"
[4] "average_intensity"
> colnames(hourly_steps)
[1] "id"            "activity_hour" "step_total"
```

2. I checked the data type and format of headers in each data frame using the **glimpse()** function to ensure they are consistent with the metadata.

```r
# show data type and format

glimpse(daily_activity)
glimpse(daily_sleep)
glimpse(hourly_calories)
glimpse(hourly_intensities)
glimpse(hourly_steps)
```

```
> glimpse(daily_activity)
Rows: 940
Columns: 15
$ id                        <dbl> 1503960366, 1503960366, 1503960366, 150396036…
$ activity_date             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/…
$ total_steps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019…
$ total_distance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8…
$ tracker_distance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8…
$ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ very_active_distance      <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5…
$ moderately_active_distance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3…
$ light_active_distance     <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0…
$ sedentary_active_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
$ very_active_minutes       <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4…
$ fairly_active_minutes     <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21…
$ lightly_active_minutes    <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, …
$ sedentary_minutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818…
$ calories                  <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203…
> glimpse(daily_sleep)
Rows: 413
Columns: 5
$ id                  <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150…
$ sleep_day           <chr> "4/12/16", "4/13/16", "4/15/16", "4/16/16", "4/17/1…
$ total_sleep_records <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ total_minutes_asleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2…
$ total_time_in_bed   <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3…
> glimpse(hourly_calories)
Rows: 22,099
Columns: 3
$ id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366,…
$ activity_hour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/2016…
$ calories     <dbl> 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73, 66, 11…
> glimpse(hourly_intensities)
Rows: 22,099
Columns: 4
$ id                <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960…
$ activity_hour     <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/…
$ total_intensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 58,…
$ average_intensity <dbl> 0.333333, 0.133333, 0.116667, 0.000000, 0.000000, 0.000…
> glimpse(hourly_steps)
Rows: 22,099
Columns: 3
$ id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366,…
$ activity_hour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/2016…
$ step_total   <dbl> 373, 160, 151, 0, 0, 0, 0, 0, 250, 1864, 676, 360, 253, 221…
```

It appears that **id** is read as FLOAT type and **activity_date, sleep_day,** and **activity_hour** are read as STRING type. Therefore, I converted these columns into their appropriate formats using the **mutate()** function.

```r
# typecast columns: daily_activity

daily_activity <- daily_activity %>%
  mutate(id = as.character(id)) %>%
  mutate(activity_date = mdy(activity_date))

# typecast columns: daily_sleep

daily_sleep <- daily_sleep %>%
  mutate(id = as.character(id)) %>%
  mutate(sleep_day = mdy(sleep_day))

# typecast columns: hourly_calories

hourly_calories <- hourly_calories %>%
  mutate(id = as.character(id)) %>%
  mutate(activity_hour = mdy_hms(activity_hour))

# typecast columns: hourly_intensities

hourly_intensities <- hourly_intensities %>%
  mutate(id = as.character(id)) %>%
  mutate(activity_hour = mdy_hms(activity_hour))

# typecast columns: hourly_steps

hourly_steps <- hourly_steps %>%
  mutate(id = as.character(id)) %>%
  mutate(activity_hour = mdy_hms(activity_hour))
```

3. For uniformity, I renamed the headers in DATE format as **date** and those in DATE-TIME format as **date_time** using the **rename()** function.

```r
# rename headers: daily_activity

daily_activity <- daily_activity %>%
  rename(date = activity_date)

# rename headers: daily_sleep

daily_sleep <- daily_sleep %>%
  rename(date = sleep_day)

# rename headers: hourly_calories

hourly_calories <- hourly_calories %>%
  rename(date_time = activity_hour)

# rename headers: hourly_intensities

hourly_intensities <- hourly_intensities %>%
  rename(date_time = activity_hour)

# rename headers: hourly_steps

hourly_steps <- hourly_steps %>%
  rename(date_time = activity_hour)
```

4. I checked for observations with missing or null values using the **!complete.cases()** function.

```r
# count incomplete observations

sum(!complete.cases(daily_activity))
sum(!complete.cases(daily_sleep))
sum(!complete.cases(hourly_calories))
sum(!complete.cases(hourly_intensities))
sum(!complete.cases(hourly_steps))
```

```
> sum(!complete.cases(daily_activity))
[1] 0
> sum(!complete.cases(daily_sleep))
[1] 0
> sum(!complete.cases(hourly_calories))
[1] 0
> sum(!complete.cases(hourly_intensities))
[1] 0
> sum(!complete.cases(hourly_steps))
[1] 0
```

No incomplete observations were found in all data frames.

5. I checked for duplicated observations using the **duplicated()** function.

```r
# count duplicated observations

sum(duplicated(daily_activity))
sum(duplicated(daily_sleep))
sum(duplicated(hourly_calories))
sum(duplicated(hourly_intensities))
sum(duplicated(hourly_steps))
```

```
> sum(duplicated(daily_activity))
[1] 0
> sum(duplicated(daily_sleep))
[1] 3
> sum(duplicated(hourly_calories))
[1] 0
> sum(duplicated(hourly_intensities))
[1] 0
> sum(duplicated(hourly_steps))
[1] 0
```

I found three duplicated observations in **daily_sleep** and removed them using the **distinct()** function.

```r
# remove duplicated observations

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

6. If the tracker is utilized, it should record at least one step in a day. Hence, I checked for observations with 0 steps in **daily_activity.**

```r
# count invalid observations

invalid_steps <- daily_activity %>%
  filter(total_steps==0)

nrow(invalid_steps)
```

```
> nrow(invalid_steps)
[1] 77
>
```

I found 77 observations with invalid steps value and removed them using the **filter()** function.

```r
# remove invalid observations

daily_activity <- daily_activity %>%
  filter(total_steps!=0)
```

7. If the tracker works properly, the total minutes spent in four intensity levels of daily activity should not exceed 1440 minutes (24 hours). Thus, I checked for observations with total activity time greater than 1440 minutes in **daily_activity** by temporarily adding the values from all intensity levels**.**

```r
# create a column for total activity time

daily_activity_time <- daily_activity %>%
  mutate(total_minutes = very_active_minutes +
         fairly_active_minutes +
         lightly_active_minutes +
         sedentary_minutes)

# count invalid observations

invalid_time <- daily_activity_time %>%
  filter(total_minutes > 1440)

nrow(invalid_time)
```

```
> nrow(invalid_time)
[1] 0
>
```

No observations with invalid time value were found.

III. Transform the data

1. I decided to combine the cleaned data frames with the same timeframe using the **inner_join()** function. This would return only the observations that have matching values of **id** and **date/date_time.**

I first merged **daily_activity** and **daily_sleep** into a new data frame named **daily_activity_sleep**. Then, I also merged **hourly_calories,**

hourly_intensities, and **hourly_steps** into a new data frame named **hourly_activity.**

```
# merge daily data frames

daily_activity_sleep <- inner_join(daily_activity,
daily_sleep, by = c("id", "date"))

# merge hourly data frames

hourly_activity <- inner_join(hourly_calories,
hourly_intensities, by = c("id", "date_time")) %>%
  inner_join(hourly_steps, by = c("id", "date_time"))
```

Let's check the number of rows and users (unique id) in each data frame.

```
# count rows and users

nrow(daily_activity_sleep)
n_distinct(daily_activity_sleep$id)

nrow(hourly_activity)
n_distinct(hourly_activity$id)
```

```
> nrow(daily_activity_sleep)
[1] 410
> n_distinct(daily_activity_sleep$id)
[1] 24
> nrow(hourly_activity)
[1] 22099
> n_distinct(hourly_activity$id)
[1] 33
```

With the sample size of **daily_activity_sleep** significantly reduced to match **daily_sleep**, I would prefer using **daily_activity** for finding activity trends due to its broad coverage. However, **daily_activity_sleep** would still be valuable for finding sleep trends and correlating activity and sleep variables. Meanwhile, since **hourly_activity** retained the sample size of the three hourly data frames, I chose to alternatively use the former.

In short, I would use three data frames for my analysis: **daily_activity,** **daily_activity_sleep,** and **hourly_activity.**

2. I added some columns that would be helpful in analyzing the data frames using the **mutate()** function.

I first inserted a column for the day of the week (Monday to Sunday) named **day** in all data frames. To do that, I first split the **date_time** column in **hourly_activity** and cast the divided columns in their appropriate data types.

```r
# separate date and time in hourly_activity

hourly_activity <- hourly_activity %>%
  separate(col = date_time, into = c("date", "time"),
           sep = " ", remove = FALSE) %>%
  mutate(date = ymd(date), time = hms(time))

# add a column for day

daily_activity <- daily_activity %>%
  mutate(day = weekdays(date))

daily_activity_sleep <- daily_activity_sleep %>%
  mutate(day = weekdays(date))

hourly_activity <- hourly_activity %>%
  mutate(day = weekdays(date))
```

Then, I inserted a column for time spent in non-sedentary activity named **total_active_minutes** in **daily_activity** and **daily_activity_sleep** computed as the sum of **very_active_minutes, fairly_active_minutes,** and **lightly_active_minutes.**

```r
# add a column for non-sedentary activity

daily_activity <- daily_activity %>%
  mutate(total_active_minutes = very_active_minutes +
         fairly_active_minutes + lightly_active_minutes)

daily_activity_sleep <- daily_activity_sleep %>%
  mutate(total_active_minutes = very_active_minutes +
         fairly_active_minutes + lightly_active_minutes)
```

Lastly, I inserted a column for awake time in bed named **total_minutes_awake** in **daily_activity_sleep** computed as **total_time_in_bed** minus total_minutes_asleep.

```r
# add a column for awake time in bed

daily_activity_sleep <- daily_activity_sleep %>%
  mutate(total_minutes_awake = total_time_in_bed -
         total_minutes_asleep)
```

3. In contrast, I removed a few columns that would unlikely help in my analysis. Such columns are related to distance, which I believe would provide minimal value in understanding device usage and even user health profile.

```r
# remove columns

daily_activity <- daily_activity [, -c(4:10)]

daily_activity_sleep <- daily_activity_sleep [, -c(4:10)]
```

4. I sorted the columns in each data frame in a custom order.

```r
# sort columns

daily_activity <- daily_activity [, c(1, 2, 9, 8, 3, 10, 4:7)]

daily_activity_sleep <- daily_activity_sleep [, c(1, 2, 12, 8,
3, 13, 4:7, 11, 10, 14, 9)]

hourly_activity <- hourly_activity [, c(1:3, 9, 4:8)]
```

5. I sorted the rows in each data frame first by **id** and then by **date/date_time.**

```r
# sort rows

daily_activity <- daily_activity %>%
  arrange(id, date)

daily_activity_sleep <- daily_activity_sleep %>%
  arrange(id, date)

hourly_activity <- hourly_activity %>%
  arrange(id, date_time)
```

These new versions of **daily_activity,** **daily_activity_sleep,** and **hourly_activity** shall be the final version of the dataset that will be used for analysis and visualization.

IV. View the data

Here's a preview of the **daily_activity** data frame:

```r
# preview daily_activity

head(daily_activity)
```

```
# A tibble: 6 × 10
  id       date       day   calor…¹ total…² total…³ very_…⁴ fairl…⁵ light…⁶ seden…⁷
  <chr>    <date>     <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 1503960… 2016-04-12 Tues…    1985   13162     366      25      13     328     728
2 1503960… 2016-04-13 Wedn…    1797   10735     257      21      19     217     776
3 1503960… 2016-04-14 Thur…    1776   10460     222      30      11     181    1218
4 1503960… 2016-04-15 Frid…    1745    9762     272      29      34     209     726
5 1503960… 2016-04-16 Satu…    1863   12669     267      36      10     221     773
6 1503960… 2016-04-17 Sund…    1728    9705     222      38      20     164     539
# … with abbreviated variable names ¹calories, ²total_steps,
#   ³total_active_minutes, ⁴very_active_minutes, ⁵fairly_active_minutes,
#   ⁶lightly_active_minutes, ⁷sedentary_minutes
```

Here's a preview of the **daily_activity_sleep** data frame:

```r
# preview daily_activity_sleep

head(daily_activity_sleep)
```

```
# A tibble: 6 × 14
  id       date       day   calor…¹ total…² total…³ very_…⁴ fairl…⁵ light…⁶ seden…⁷
  <chr>    <date>     <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 1503960… 2016-04-12 Tues…    1985   13162     366      25      13     328     728
2 1503960… 2016-04-13 Wedn…    1797   10735     257      21      19     217     776
3 1503960… 2016-04-15 Frid…    1745    9762     272      29      34     209     726
4 1503960… 2016-04-16 Satu…    1863   12669     267      36      10     221     773
5 1503960… 2016-04-17 Sund…    1728    9705     222      38      20     164     539
6 1503960… 2016-04-19 Tues…    2035   15506     345      50      31     264     775
# … with 4 more variables: total_time_in_bed <dbl>, total_minutes_asleep <dbl>,
#   total_minutes_awake <dbl>, total_sleep_records <dbl>, and abbreviated variable
#   names ¹calories, ²total_steps, ³total_active_minutes, ⁴very_active_minutes,
#   ⁵fairly_active_minutes, ⁶lightly_active_minutes, ⁷sedentary_minutes
# ℹ Use `colnames()` to see all variable names
```

Here's a preview of the **hourly_activity** data frame:

```r
# preview hourly_activity

head(hourly_activity)
```

```
# A tibble: 6 × 9
  id         date_time           date       day    time     calor…¹ total…² avera…³
  <chr>      <dttm>              <date>     <chr>  <Period>   <dbl>   <dbl>   <dbl>
1 1503960366 2016-04-12 00:00:00 2016-04-12 Tuesd… 0S            81      20   0.333
2 1503960366 2016-04-12 01:00:00 2016-04-12 Tuesd… 1H 0M 0S      61       8   0.133
3 1503960366 2016-04-12 02:00:00 2016-04-12 Tuesd… 2H 0M 0S      59       7   0.117
4 1503960366 2016-04-12 03:00:00 2016-04-12 Tuesd… 3H 0M 0S      47       0   0
5 1503960366 2016-04-12 04:00:00 2016-04-12 Tuesd… 4H 0M 0S      48       0   0
6 1503960366 2016-04-12 05:00:00 2016-04-12 Tuesd… 5H 0M 0S      48       0   0
# … with 1 more variable: step_total <dbl>, and abbreviated variable names
#   ¹calories, ²total_intensity, ³average_intensity
# ℹ Use `colnames()` to see all variable names
```

Step 4: Analyze and Share

DEVICE USAGE ANALYSIS

Based on the available information in the data frames, I opted to explore the following trends and patterns in Leaf device usage:

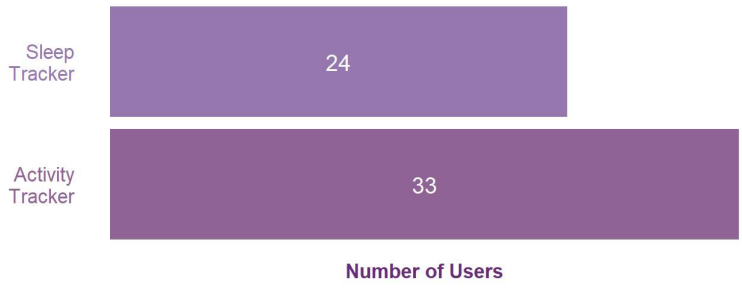
- Usage rate of device features
- Usage rate by day in a week
- Usage rate by hour in a day
- Usage rate by hour in a week

I once again used RStudio for data analysis and visualization. Then, I used Tableau to create a dashboard. Visit my Tableau profile to check the data viz for this project.

I. Usage rate of device features

I created a horizontal bar chart to identify which Leaf smart device feature (Activity or Sleep Tracker) is used more often.

```r
# create a data frame

usage_by_features <- data.frame(
  device_feature = c("Activity Tracker", "Sleep Tracker"),
  no_of_users = c(
    n_distinct(daily_activity$id),
    n_distinct(daily_activity_sleep$id)))

# create a horizontal bar chart

ggplot(usage_by_features, aes(x = no_of_users, y =
device_feature, fill = device_feature)) +
  geom_bar(stat = "identity") +
  geom_text(
    aes(label = no_of_users),
    position = position_stack(vjust = 0.5),
    color = "white", size = 10) +
  labs(x = "Number of Users", y = "Device Feature") +
  scale_fill_manual(values = c("#946597", "#9A7BB3")) +
  scale_y_discrete(
    labels = c("Activity\nTracker", "Sleep\nTracker")) +
  theme(
    panel.background = element_blank(),
    axis.title.x = element_text(
      size = 25, color = "#6B2D7C", face = "bold"),
    axis.title.y = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_text(
      size = 25, color = c("#946597", "#9A7BB3")),
    axis.ticks = element_blank(),
    legend.position = "none")
```



Based on this chart, the Leaf smart device's Activity Tracker feature (monitors calories, steps, and activity levels or intensities) appears to be more often used than its Sleep Tracker feature (monitors asleep and awake time in bed).

This finding suggests that users might be more interested in tracking their activity patterns, as well as their progress toward fitness goals, than monitoring their sleep patterns.

Users might also perceive the Sleep Tracker feature as less relevant to their health pursuits, believing sleep quality is not as critical as physical activity

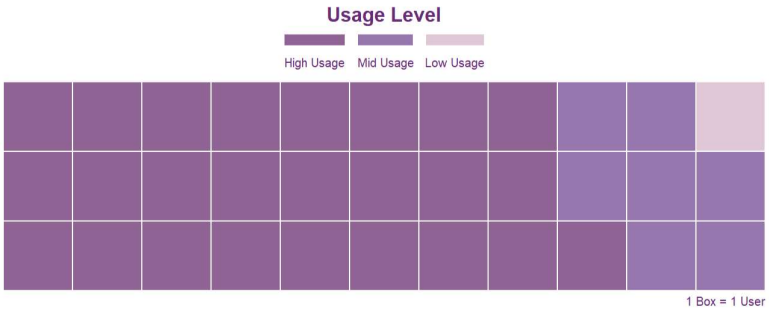
to their overall health.

Lastly, users might find the Activity Tracker feature more user-friendly or convenient to use since it is easier to activate and more reliable during the daytime.

II. User type distribution

I made a waffle chart to know how various Leaf users are distributed based on their usage level (high, mid, or low usage).

```r
# create a data frame

user_type <- daily_activity %>%
  count(id, name = "days_used") %>%
  mutate(usage_level = cut(
    days_used, c(0, 10, 20, 31),
    labels = c("Low Usage", "Mid Usage", "High Usage"))) %>%
  count(usage_level, name = "no_of_users") %>%
  pivot_wider(
    names_from = usage_level, values_from = no_of_users) %>%
  select("High Usage", "Mid Usage", "Low Usage")

# create a waffle chart

waffle(user_type, row = 3, size = 1, colors = c("#946597",
"#9A7BB3", "#E3CBDA"), legend = "left") +
  labs(caption = "1 Box = 1 User") +
  guides(fill = guide_legend(
    title = "Usage Level", title.position = "top",
    title.hjust = 0.5, label.position = "bottom")) +
  theme(
    plot.caption = element_text(size = 15, color = "#6B2D7C"),
    legend.position = "top",
    legend.title = element_text(
      size = 25, color = "#6B2D7C", face = "bold"),
    legend.text = element_text(
      size = 15, color = "#6B2D7C"))
```



Based on this diagram, the majority of Leaf smart device users (75%) demonstrated high-level device usage (21-31 days of usage per month), while 21% and 3% of the users showed mid-level (11-20 days) and low-level (0-10 days) device usage, respectively.

This finding indicates that users might be highly satisfied with the Leaf device's performance as it might be effectively fulfilling their health monitoring needs.

Frequent users might also have more specific fitness goals they want to achieve, and the device greatly helps them track their progress toward

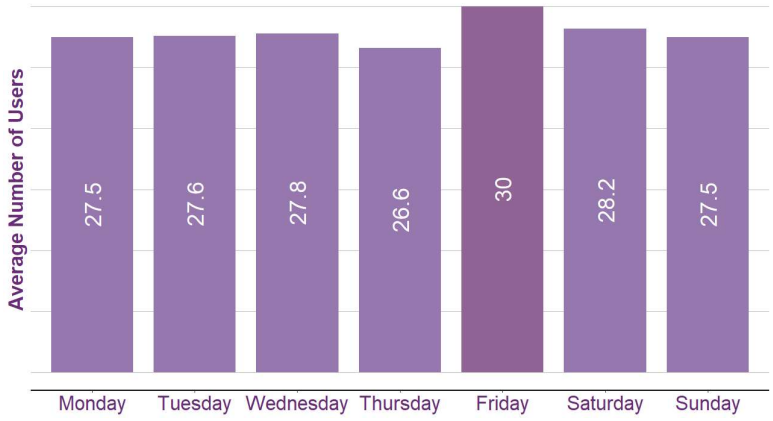
those goals.

Finally, frequent users might also have had a better onboarding experience, which leads to easier access and usage of device features and functionalities.

III. Usage rate by day of the week

I created a vertical bar chart to determine which day of the week is Leaf device usage highest and lowest.

```r
# create a data frame

usage_by_day <- daily_activity %>%
  group_by(date, day) %>%
  summarise(no_of_users = n_distinct(id)) %>%
  group_by(day) %>%
  summarise(avg_no_of_users = round(mean(no_of_users), 1))

# create a vertical bar chart

ggplot(usage_by_day, aes(x = day, y = avg_no_of_users)) +
  geom_bar(
    aes(fill = day %in% c("Friday")),
    stat = "identity", width = 0.8) +
  geom_text(
    aes(label = avg_no_of_users),
    position = position_stack(vjust = 0.5),
    color = "white", size = 10, angle = 90, hjust = 0.5)  +
  labs(x = "Day", y = "Average Number of Users") +
  scale_fill_manual(
    values = c("#9A7BB3", "#946597"), guide = FALSE) +
  scale_x_discrete(limits = c(
    "Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday", "Sunday")) +
  theme(
    panel.background = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.major.y = element_line(color = "grey"),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_line(color = "grey"),
    axis.title.x = element_blank(),
    axis.title.y = element_text(
      size = 25, color = "#6B2D7C", face = "bold"),
    axis.text.x = element_text(
      size = 25, color = "#6B2D7C"),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_line(size = 1))
```



Based on this graph, the Leaf device appears to be consistently used each day of the week. However, there is a slight variation in usage pattern as the device is least used on Thursdays and most used on Fridays.

This trend suggests that users might be less motivated to use the device on Thursdays since they have already been tracking their health for most of the week.

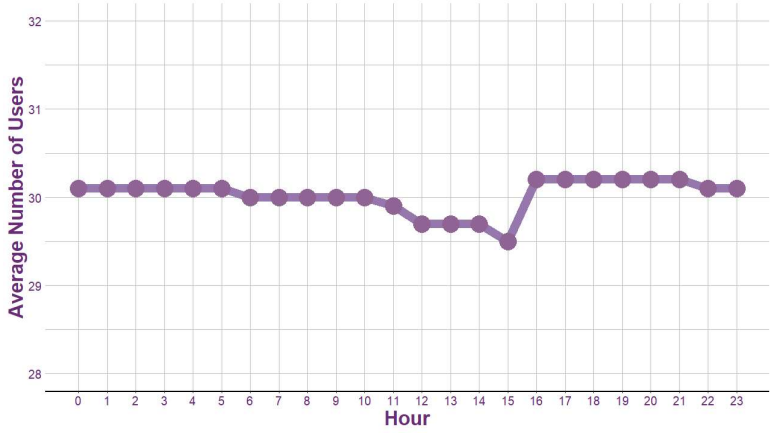
Conversely, users might be more motivated to use the device on Fridays because they want to ensure they are meeting their fitness goals before the week ends.

It is also possible that such a pattern could be influenced by users' work or social schedules. For example, users might have more commitments on Thursdays, while they might be less busy on Fridays.

IV. Usage rate by hour in a day

I made a line chart to track when Leaf device usage is highest and lowest throughout the day.

```r
# create a data frame

usage_by_hour <- hourly_activity %>%
  group_by(date_time, hour = hour(date_time)) %>%
  summarise(no_of_users = n_distinct(id)) %>%
  group_by(hour) %>%
  summarise(avg_no_of_users = round(mean(no_of_users), 1))

# create a vertical bar chart

ggplot(usage_by_hour, aes(x = hour, y = avg_no_of_users)) +
  labs(x = "Hour", y = "Average Number of Users") +
  geom_line(color="#9A7BB3", size=5) +
  geom_point(color="#946597", size=10) +
  scale_x_continuous(
    breaks = seq(0, 23), labels = paste0(seq(0, 23))) +
  ylim(28, 32) +
  theme(
    panel.background = element_blank(),
    panel.grid.major.x = element_line(color = "grey"),
    panel.grid.major.y = element_line(color = "grey"),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_line(color = "grey"),
    axis.title.x = element_text(
      size = 25, color = "#6B2D7C", face = "bold"),
    axis.title.y = element_text(
      size = 25, color = "#6B2D7C", face = "bold"),
    axis.text.x = element_text(
      size = 15, color = "#6B2D7C"),
    axis.text.y = element_text(
      size = 15, color = "#6B2D7C"),
    axis.ticks.y = element_blank(),
    axis.line.x = element_line(size = 1))
```



Based on this illustration, the Leaf device is found to be consistently utilized throughout the day, with a slight gradual decline in usage from midnight to 3 PM followed by a spike in usage after 3 PM.

This trend indicates that users might be less likely to use the device during regular working hours and more likely to use the device after work or during leisure or free time.

Users might also be more inclined to engage in certain types of activities, like walking and jogging, in the late afternoon or early evening when the weather is cooler.

Certainly, Bellabeat's reminder notifications could also be influencing this pattern as users might be receiving more notifications or alarms around 3 PM onwards.

USER WELLNESS PROFILE

Although my primary task is to only analyze how users are utilizing the Leaf device, I also decided to probe the following trends and patterns in user wellness status:

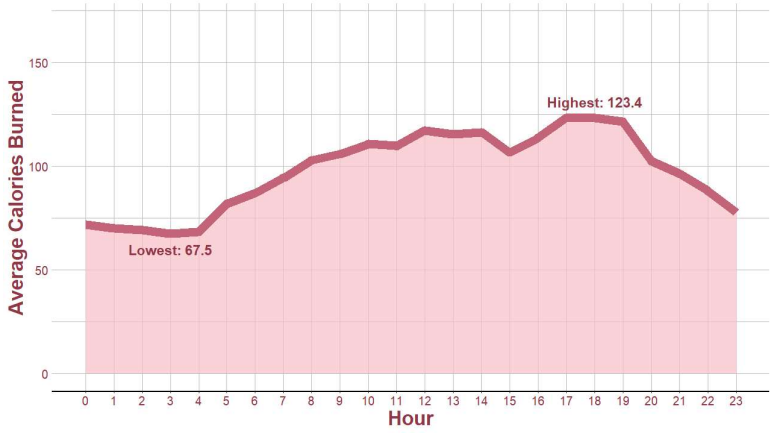
- Average calories burned by hour
- Average steps taken by hour
- Average active time by day
- Average time in bed by day

I believe understanding their wellness profile can offer additional insights to inform the company's marketing strategy.

I. Average calories burned by hour

I created an area chart to identify what time of the day Leaf users burn the most and least calories.

```r
# create a data frame

calories_by_hour <- hourly_activity %>%
  group_by(date_time, hour = hour(date_time)) %>%
  summarise(calories_burned = mean(calories)) %>%
  group_by(hour) %>%
  summarise(avg_calories_burned = mean(calories_burned))

# compute max and min values for annotation

max_row <- which.max(calories_by_hour$avg_calories_burned)
min_row <- which.min(calories_by_hour$avg_calories_burned)

# create a line chart

ggplot(calories_by_hour, aes(x = hour, y =
avg_calories_burned)) +
  geom_area(
    fill = "#FCC0C9", alpha = 0.7,
    color = "#C6687B", size = 5) +
  labs(x = "Hour", y = "Average Calories Burned") +
  scale_x_continuous(
    breaks = seq(0, 23), labels = paste0(seq(0, 23))) +
  ylim(0,170) +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "grey"),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_line(color = "grey"),
    axis.title = element_text(
      size = 25, color = "#94384A", face = "bold"),
    axis.text = element_text(
      size = 15, color = "#94384A"),
    axis.ticks.y = element_blank(),
    axis.line.x = element_line(size = 1),
    legend.position = "none") +

  # add annotation

  annotate(
    "text", label = paste0("Highest:", round(
      calories_by_hour$avg_calories_burned[max_row], 1)),
    x = calories_by_hour$hour[max_row],
    y = calories_by_hour$avg_calories_burned[max_row],
    vjust = -1, color = "#94384A",
    size = 6, fontface = "bold") +
  annotate(
    "text", label = paste0("Lowest: ", round(
      calories_by_hour$avg_calories_burned[min_row], 1)),
    x = calories_by_hour$hour[min_row],
    y = calories_by_hour$avg_calories_burned[min_row]-5,
    vjust = 1, color = "#94384A", size = 6, fontface = "bold")
```



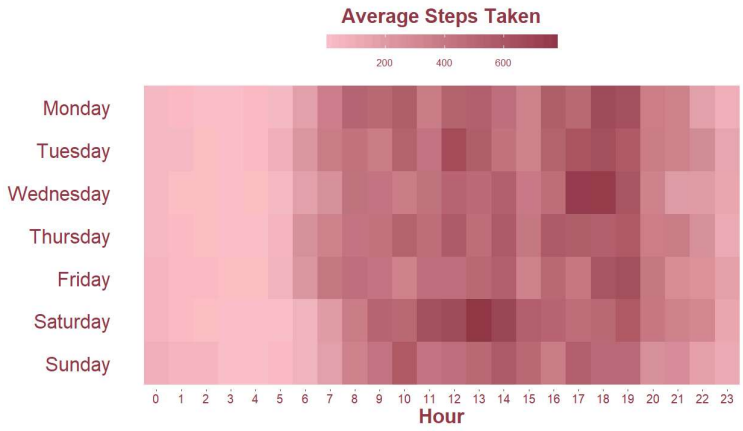
Based on this graph, Leaf users tend to burn fewer calories in the early morning hours and burn more calories in the early evening. This could be due to many factors, such as work schedules, leisure activities, or mealtimes.

Evidently, there is also a gradual increase in calorie expenditure throughout the day. This could be because users become more active as the day progresses brought by the accumulation of energy and motivation.

II. Average steps taken by hour

I made a heat map to see what time of the day Leaf users take the most and least number of steps.

```r
# create a data frame

steps_by_hour <- hourly_activity %>%
  group_by(hour = hour(date_time), day = weekdays(date)) %>%
  reframe(avg_no_of_steps = mean(step_total))

# create heat map

ggplot(steps_by_hour, aes(x = hour, y = day, fill =
avg_no_of_steps)) +
  geom_tile() +
  labs(x = "Hour", y = "Day", fill = "Average Steps Taken") +
  scale_x_continuous(
    breaks = seq(0, 23), labels = paste0(seq(0, 23))) +
  scale_y_discrete(limits = c(
    "Sunday", "Saturday", "Friday", "Thursday",
    "Wednesday", "Tuesday", "Monday")) +
  scale_fill_gradient(
    low = "#FCC0C9", high = "#94384A", limits = c(2, 780)) +
  guides(fill = guide_colorbar(
    barwidth = 20, title.position = "top",
    title.hjust = 0.5,label.position = "bottom")) +
  theme(
    panel.background = element_blank(),
    panel.grid = element_blank(),
    axis.title.x = element_text(
      size = 25, color = "#94384A", face = "bold"),
    axis.title.y = element_blank(),
    axis.text.x = element_text(
      size = 15, color = "#94384A"),
    axis.text.y = element_text(
      size = 25, color = "#94384A"),
    axis.ticks.y = element_blank(),
    legend.title = element_text(
      size = 25, color = "#94384A", face = "bold"),
    legend.text = element_text(
      size = 12, color = "#94384A"),
    legend.position = "top")
```



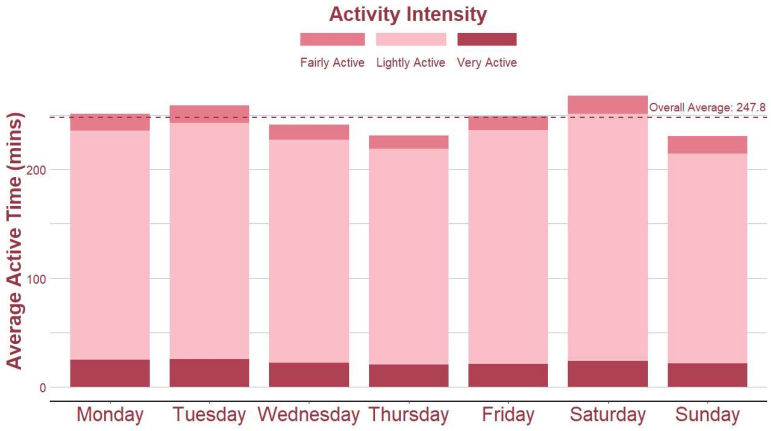
Based on this chart, Leaf users typically take the most steps between 8 AM and 7 PM and the least steps between 12 AM to 4 AM, proving they are more active during the daytime hours and less active during the nighttime hours. This is a common pattern among people who follow a typical workday schedule.

Furthermore, there is a notable increase in step count between 6 to 7 AM on weekdays compared to weekends, implying that users are more likely to engage in early physical activities, like commuting to work, on weekdays than on weekends when they have more leisure time.

III. Average active time by day

I created a vertical stacked bar chart to identify which day of the week Leaf users are most and least active (divided into very, fairly, and lightly active).

```r
# create a data frame

active_time_by_day <- daily_activity %>%
  group_by(date, day) %>%
  summarise(across(c(
    very_active_minutes, fairly_active_minutes,
    lightly_active_minutes), mean)) %>%
  group_by(day) %>%
  summarise(across(
    c(
      very_active_minutes, fairly_active_minutes,
      lightly_active_minutes),
    ~ round(mean(.), 1), .names = "avg_{.col}")) %>%
  pivot_longer(
    cols = starts_with("avg_"),
    names_to = "activity_intensity",
    values_to = "avg_active_minutes")

# compute overall average for annotation

overall_avg_active <- mean(
  daily_activity$total_active_minutes)

# create a vertical stacked bar chart

ggplot(active_time_by_day, aes(x = day, y =
avg_active_minutes, fill = activity_intensity)) +
  geom_bar(stat = "identity", width = 0.8) +
  labs(
    x = "Day", y = "Average Active Time (mins)",
    fill = "Activity Intensity") +
  scale_x_discrete(limits = c(
    "Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday", "Sunday")) +
  scale_fill_manual(
    values = c("#EA7F8E", "#FCC0C9", "#B24459"),
    labels = c(
      "Fairly Active", "Lightly Active", "Very Active")) +
  guides(fill = guide_legend(
    title.position = "top", title.hjust = 0.5,
    label.position = "bottom")) +
  theme(
    panel.background = element_blank(),
    panel.grid.major.x=element_blank(),
    panel.grid.major.y=element_line(color="grey"),
    panel.grid.minor.x=element_blank(),
    panel.grid.minor.y=element_line(color="grey"),
    axis.title.x = element_blank(),
    axis.title.y = element_text(
      size = 25, color = "#94384A", face = "bold"),
    axis.text.x = element_text(
      size = 25, color = "#94384A"),
    axis.text = element_text(
      size = 15, color = "#94384A"),
    axis.ticks.y = element_blank(),
    axis.line.x = element_line(size = 1),
    legend.position = "top",
    legend.title = element_text(
      size = 25, color = "#94384A", face = "bold"),
    legend.text = element_text(
      size = 15, color = "#94384A")) +

  # add annotation

  annotate(
    "text", label = paste0(
      "Overall Average: ", round(overall_avg_active, 1)),
    x = 7, y = overall_avg_active + 10,
    color = "#94384A", size = 5) +
  geom_hline(
    yintercept = overall_avg_active, color = "#94384A",
    linetype = "dashed", size = 1)
```



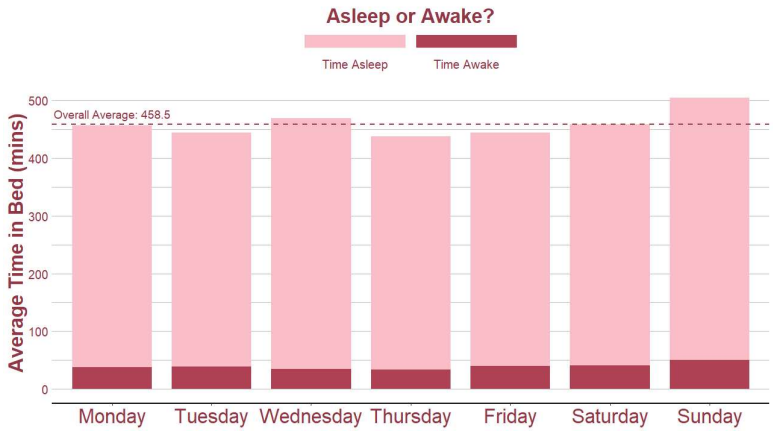
Based on this diagram, Leaf users appear to have a consistent activity pattern throughout the week with a slight variation since they have above-average activity levels on Mondays, Tuesdays, and Saturdays, signifying that users could have a specific routine or schedule during these days that influence their activity levels.

Additionally, users also tend to mostly engage in light-intensity activities every day, suggesting that the Leaf device is more suitable for tracking low-intensity activities, such as walking or light exercises, rather than intense workouts.

IV. Average time in bed by day

I created another vertical stacked bar chart to determine what day of the week Leaf users spend the most and least time in bed (including both awake and asleep time).

```r
# create a data frame

bed_time_by_day <- daily_activity_sleep %>%
  group_by(date, day) %>%
  summarise(across(c(
    total_minutes_asleep, total_minutes_awake), mean)) %>%
  group_by(day) %>%
  summarise(across(
    c(total_minutes_asleep, total_minutes_awake),
    ~ round(mean(.), 1), .names = "avg_{.col}")) %>%
  pivot_longer(
    cols = starts_with("avg_"), names_to = "asleep_awake",
    values_to = "avg_bedtime_minutes")

# compute overall average for annotation

overall_avg_bedtime <- mean(
  daily_activity_sleep$total_time_in_bed)

# create a vertical stacked bar chart

ggplot(bed_time_by_day, aes(x = day, y = avg_bedtime_minutes,
fill = asleep_awake)) +
  geom_bar(stat = "identity", width = 0.8) +
  labs(
    x = "Day", y = "Average Time in Bed (mins)",
    fill = "Asleep or Awake?") +
  scale_x_discrete(limits = c(
    "Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday", "Sunday")) +
  scale_fill_manual(
    values = c("#FCC0C9", "#B24459"),
    labels = c("Time Asleep", "Time Awake")) +
  guides(fill = guide_legend(
    title.position = "top", title.hjust = 0.5,
    label.position = "bottom")) +
  theme(
    panel.background = element_blank(),
    panel.grid.major.x=element_blank(),
    panel.grid.major.y=element_line(color="grey"),
    panel.grid.minor.x=element_blank(),
    panel.grid.minor.y=element_line(color="grey"),
    axis.title.x = element_blank(),
    axis.title.y = element_text(
      size = 25, color = "#94384A", face = "bold"),
    axis.text.x = element_text(
      size = 25, color = "#94384A"),
    axis.text = element_text(
      size = 15, color = "#94384A"),
    axis.ticks.y = element_blank(),
    axis.line.x = element_line(size = 1),
    legend.position = "top",
    legend.title = element_text(
      size = 25, color = "#94384A", face = "bold"),
    legend.text = element_text(
      size = 15, color = "#94384A")) +

  # add annotation

  annotate(
    "text", label = paste0(
      "Overall Average: ", round(overall_avg_bedtime, 1)),
    x = 1, y = overall_avg_bedtime + 18,
    color = "#94384A", size = 5) +
  geom_hline(
    yintercept = overall_avg_bedtime, color = "#94384A",
    linetype = "dashed", size = 1)
```



Based on this illustration, Leaf users appear to have a consistent sleep pattern throughout the week with a slight variation as they have longer bedtime and sleep durations on Wednesdays and Sundays, indicating that these are the days when they have fewer commitments or responsibilities allowing them to get more rest.

On the other hand, users have shorter bedtime and sleep durations on Tuesdays, Thursdays, and Fridays, suggesting that they may have busier schedules or may be prioritizing other activities over sleep during these days.

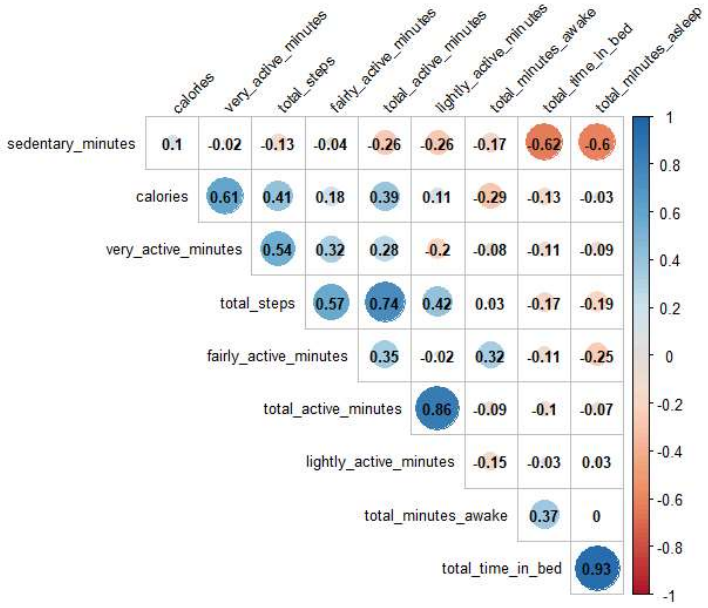
FINDING CORRELATIONS

Additionally, I decided to further investigate the relationships between various health indicators in the dataset to somewhat verify the Leaf device's precision in recording user health data.

To begin with, I performed a Pearson correlation test for the numeric variables in **daily_activity_sleep** and **hourly_activity** to identify the variables that exhibited a strong relationship. Then, I transformed the test results into a correlation plot for practical purposes.

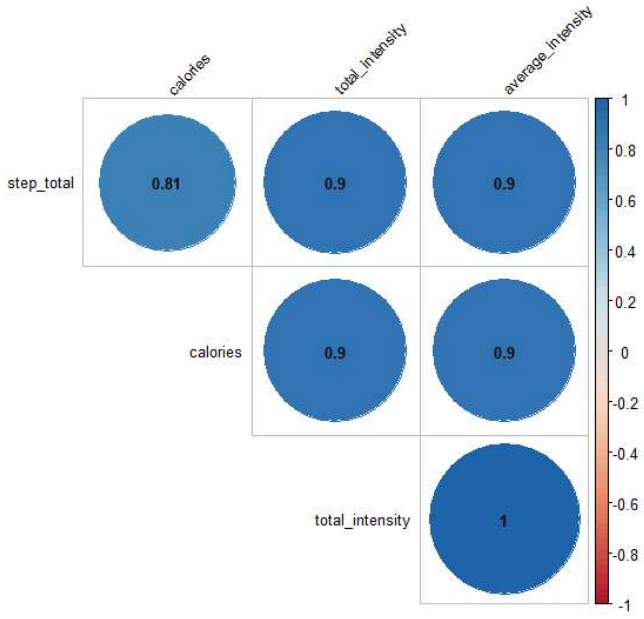
Here's the Pearson correlation test for **daily_activity_sleep:**

```r
# calculate correlation coefficients

corr_daily_activity_sleep <- daily_activity_sleep[, 4:13] %>%
  cor(method = "pearson")

# create a correlation plot

corrplot(
  corr_daily_activity_sleep, type = "upper", order = "hclust",
  col = colorRampPalette(c(
    "#b2182b", "#ef8a62", "#fddbc7", "#d1e5f0",
    "#67a9cf", "#2166ac"), space = "rgb")(100),
  tl.col = "black",
  tl.cex = 0.8,
  tl.srt = 45,
  addCoef.col = "black",
  number.cex = 0.8,
  diag = FALSE,
  method = "circle")
```



And here's the Pearson correlation test for **hourly_activity**:

```r
# calculate correlation coefficients

corr_hourly_activity <- hourly_activity[, 6:9] %>%
  cor(method = "pearson")

# create a correlation plot

corrplot(
  corr_daily_activity_sleep, type = "upper", order = "hclust",
  col = colorRampPalette(c(
    "#b2182b", "#ef8a62", "#fddbc7", "#d1e5f0",
    "#67a9cf", "#2166ac"), space = "rgb")(100),
  tl.col = "black",
  tl.cex = 0.8,
  tl.srt = 45,
  addCoef.col = "black",
  number.cex = 0.8,
  diag = FALSE,
  method = "circle")
```



The size and color of the circles reflect the strength and direction of the relationship between the health indicators. A large dark blue circle indicates a strong positive correlation (as one variable increases, the other variable tends to increase as well). Conversely, a large dark red circle indicates a strong negative correlation (as one variable increases, the other variable tends to decrease).

Based on the test results, I decided to map out the relationship between the following pairs of variables:

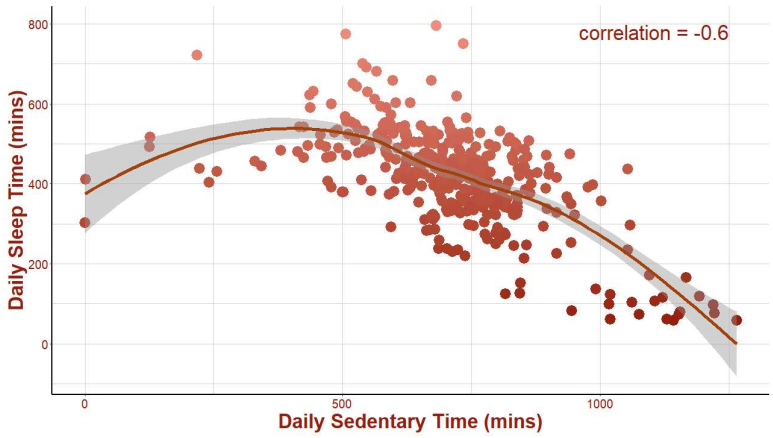
- Daily sedentary time and sleep time
- Daily active time and steps
- Hourly steps and calories
- Hourly intensity level and calories

These pairs of variables are chosen due to their relatively strong correlation.

I. Daily sedentary time and sleep time

The scatter plot below illustrates the moderate negative correlation (-0.6) between daily sedentary time and sleep time.

```r
# compute correlation coefficient

cor_coef_1 <- cor(
  daily_activity_sleep$sedentary_minutes,
  daily_activity_sleep$total_minutes_asleep)

# create a scatter plot

ggplot(data = daily_activity_sleep, aes(x = sedentary_minutes,
y = total_minutes_asleep, color = total_minutes_asleep)) +
  geom_point(size = 6) +
  geom_smooth(color = "#A5420B", size = 2) +
  labs(
    x = "Daily Sedentary Time (mins)",
    y = "Daily Sleep Time (mins)") +
  scale_color_gradient(low = "#962212", high = "#F09083") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "#D3D3D3"),
    panel.grid.minor = element_line(color="#D3D3D3"),
    axis.title = element_text(
      size = 25, color = "#962212", face = "bold"),
    axis.text = element_text(
      size = 15, color = "#962212"),
    axis.line = element_line(size = 1),
    legend.position = "none") +

  # add annotation

  annotate(
    geom = "text", label = paste0(
      "correlation = ", round(cor_coef_1, 2)),
    x = 1250, y = 800, hjust = 1, vjust = 1,
    color = "#962212", size = 9)
```



To determine if this correlation is significant, I performed a t-test with a significance level of 0.05 using the **cor.test()** function. Let the null (H0) and alternative (Ha) hypotheses be the following:

- **H0** – The population correlation coefficient is zero (no correlation).
- **Ha** – The population correlation coefficient is not zero (there is a correlation).

```r
# perform t-test

cor.test(
  daily_activity_sleep$sedentary_minutes,
  daily_activity_sleep$total_minutes_asleep,
  significance.level = 0.05)
```

The t-test would calculate the t-statistic and its corresponding p-value, which is important to ascertain if the correlation is significant.

This is the result of the t-test:

```
        Pearson's product-moment correlation

data:  daily_activity_sleep$sedentary_minutes and daily_activity_sleep$total_minutes_a
sleep
t = -15.192, df = 408, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6595278 -0.5353923
sample estimates:
       cor
-0.6010731
```

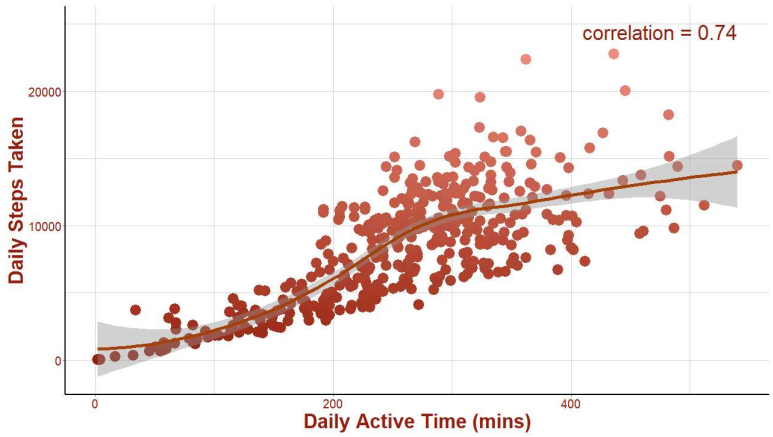
Since the p-value (< 2.2e-16) is less than the significance level, we can reject the null hypothesis and conclude that daily sedentary time and sleep time are significantly correlated, indicating that as sedentary time increases, sleep time tends to decrease.

This negative correlation is consistent with studies stating that a sedentary lifestyle is known to have detrimental effects on sleep quality and duration. Hence, it can be inferred that Leaf's tracking feature for daily sedentary time and sleep time is precise and can offer meaningful information for users who wish to enhance their health.

II. Daily active time and steps

The scatter plot below shows a strong positive correlation (0.74) between daily active time and steps.

```r
# compute correlation coefficient

cor_coef_2 <- cor(
  daily_activity_sleep$total_active_minutes,
  daily_activity_sleep$total_steps)

# create a scatter plot

ggplot(data = daily_activity_sleep, aes(x =
total_active_minutes, y = total_steps, color = total_steps)) +
  geom_point(size = 6) +
  geom_smooth(color = "#A5420B", size = 2) +
  labs(
    x = "Daily Active Time (mins)",
    y = "Daily Steps Taken") +
  scale_color_gradient(low = "#962212", high = "#F09083") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "#D3D3D3"),
    panel.grid.minor = element_line(color="#D3D3D3"),
    axis.title = element_text(
      size = 25, color = "#962212", face = "bold"),
    axis.text = element_text(
      size = 15, color = "#962212"),
    axis.line = element_line(size = 1),
    legend.position = "none") +

  # add annotation

  annotate(
    geom = "text", label = paste0(
      "correlation = ", round(cor_coef_2, 2)),
    x = 540, y = 25000, hjust = 1, vjust = 1,
    color = "#962212", size = 9)
```



To determine if this correlation is significant, I performed the same procedure above.

```r
# perform t-test

cor.test(
  daily_activity_sleep$total_active_minutes,
  daily_activity_sleep$total_steps,
  significance.level = 0.05)
```

This is the result of the t-test:

```
        Pearson's product-moment correlation

data:  daily_activity_sleep$total_active_minutes and daily_activity_sleep$total_steps
t = 22.535, df = 408, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6981498 0.7848916
sample estimates:
      cor
0.7446485
```

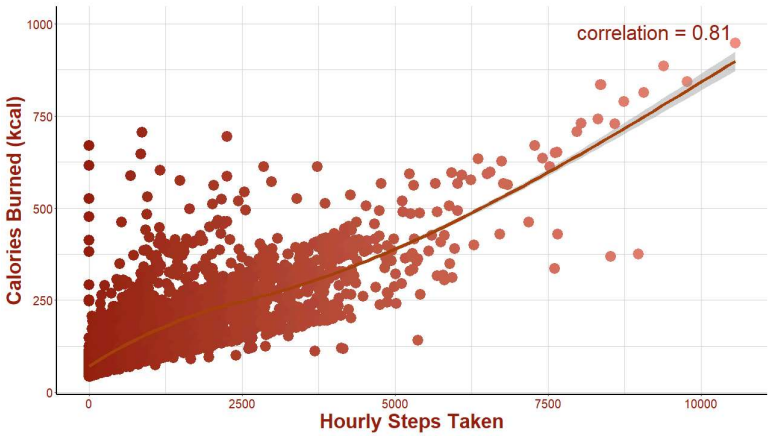
Since the p-value (< 2.2e-16) is less than the significance level, we can reject the null hypothesis and conclude that daily active time and steps are significantly associated, hinting that as active time increases, the number of steps taken also tends to increase.

This positive association is expected as activities like walking, jogging, or running, which are typical forms of physical activity, require more steps to be taken. This proves the precision of the Leaf device's measurement of daily active time and steps.

III. Hourly steps and calories

The scatter plot below displays the strong positive correlation (0.81) between hourly steps and calories.

```r
# compute correlation coefficient

cor_coef_3 <- cor(
  hourly_activity$step_total,
  hourly_activity$calories)

# create a scatter plot

ggplot(data = hourly_activity, aes(x = step_total, y =
calories, color = step_total)) +
  geom_point(size = 6) +
  geom_smooth(color = "#A5420B", size = 2) +
  labs(
    x = "Hourly Steps Taken",
    y = "Calories Burned (kcal)") +
  scale_color_gradient(low = "#962212", high = "#F09083") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "#D3D3D3"),
    panel.grid.minor = element_line(color="#D3D3D3"),
    axis.title = element_text(
      size = 25, color = "#962212", face = "bold"),
    axis.text = element_text(
      size = 15, color = "#962212"),
    axis.line = element_line(size = 1),
    legend.position = "none") +

  # add annotation

  annotate(
    geom = "text", label = paste0(
      "correlation = ", round(cor_coef_3, 2)),
    x = 10500, y = 1000, hjust = 1, vjust = 1,
    color = "#962212", size = 9)
```



To determine if this correlation is significant, I performed the same procedure above.

```r
# perform t-test

cor.test(
  hourly_activity$step_total,
  hourly_activity$calories,
  significance.level = 0.05)
```

This is the result of the t-test:

```
        Pearson's product-moment correlation

data:  hourly_activity$step_total and hourly_activity$calories
t = 209.05, df = 22097, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8104921 0.8193486
sample estimates:
      cor
0.814968
```

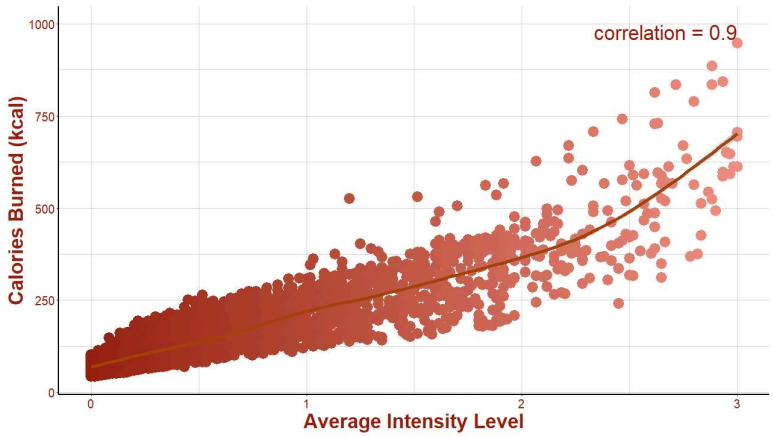
Since the p-value (< 2.2e-16) is less than the significance level, we can reject the null hypothesis and conclude that hourly steps and calories are significantly related, implying that as the number of steps taken increases, the number of calories burned also tends to increase.

This positive relationship is not surprising given that the number of steps taken is a key indicator of physical activity, and calorie expenditure is directly related to the amount of physical activity. This likewise confirms the precision of the Leaf device's recording of hourly steps and calories.

IV. Hourly intensity level and calories

The scatter plot below portrays the strong positive correlation (0.9) between hourly average intensity level and calories.

```r
# compute correlation coefficient

cor_coef_4 <- cor(
  hourly_activity$average_intensity,
  hourly_activity$calories)

# create a scatter plot

ggplot(data = hourly_activity, aes(x = average_intensity, y =
calories, color = average_intensity)) +
  geom_point(size = 6) +
  geom_smooth(color = "#A5420B", size = 2) +
  labs(
    x = "Average Intensity Level",
    y = "Calories Burned (kcal)") +
  scale_color_gradient(low = "#962212", high = "#F09083") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "#D3D3D3"),
    panel.grid.minor = element_line(color="#D3D3D3"),
    axis.title = element_text(
      size = 25, color = "#962212", face = "bold"),
    axis.text = element_text(
      size = 15, color = "#962212"),
    axis.line = element_line(size = 1),
    legend.position = "none") +

  # add annotation

  annotate(
    geom = "text", label = paste0(
      "correlation = ", round(cor_coef_4, 2)),
    x = 3, y = 1000, hjust = 1, vjust = 1,
    color = "#962212", size = 9)
```



To determine if this correlation is significant, I performed the same procedure above.

```r
# perform t-test

cor.test(
  hourly_activity$average_intensity,
  hourly_activity$calories,
  significance.level = 0.05)
```

This is the result of the t-test:

```
        Pearson's product-moment correlation

data:  hourly_activity$average_intensity and hourly_activity$calories
t = 300.99, df = 22097, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8939999 0.8991711
sample estimates:
      cor
0.8966161
```

Since the p-value (< 2.2e-16) is less than the significance level, we can reject the null hypothesis and conclude that hourly intensity level and calories are significantly connected, indicating that as the average intensity level of physical activity increases, the number of calories burned also tends to increase.

This positive connection is also anticipated because the body demands more energy to carry out activities at a higher intensity level. This could mean that the Leaf device is correctly estimating the hourly intensity level and calories.

NOTE: It is important to understand that correlation does not necessarily mean causation and further analysis is needed to determine the exact nature of the relationship between these variables.

Step 5: Act

Based on these insights, I suggest the following approaches to help Bellabeat improve the Leaf smart device and expand its market reach:

I. Focus on the Sleep Tracker feature

Bellabeat should focus on promoting the Sleep Tracker feature. Although the Activity Tracker feature is used more often, Leaf users should not overlook the importance of monitoring sleep patterns to overall health. Bellabeat can consider adding new functionalities, such as sleep tips, sleep quality analysis, or smart alarms, to motivate more users to utilize the Sleep Tracker feature.

II. Improve onboarding experience

Bellabeat should continue to provide product guidance to users. However, to ensure a better onboarding experience for all, it should also consider offering more personalized instructions tailored to match the learning curves of various users. It could also consider adding more interactive features and customized fitness challenges to make device usage more enjoyable.

III. Implement a targeted reward system

Bellabeat should implement a reward system in the form of badges, points, discounts, or other perks for users who consistently use the device throughout the week. It should also focus on marketing the device on Thursdays, when the device usage is low, through conducting surprise airdrops or challenges with huge prizes held exclusively on Thursdays.

IV. Develop stimulating features and notifications

Bellabeat should create features that encourage device usage even during regular working hours. For instance, it could consider developing a feature that reminds users to take short breaks or recommends quick stretching exercises to do at work. It could also send notifications to give users

inspiration and remind them to stay healthy and focused throughout the day.

V. Promote morning activity and calorie burn

Bellabeat should develop more features that enable users to engage in physical activity during the early morning hours, such as morning workout routines or challenges to inspire users to start the day with more energy and motivation. Bellabeat could also consider designing features that encourage users to stay active throughout the day, such as friendly calorie-burn contests.

VI. Encourage early weekend walks and jogs

Bellabeat should motivate users to start their weekends with at least a brisk walk or jog to have consistent early physical activity. It could consider sending push notifications to users or creating weekend steps competitions for users and their network. It could also partner with fitness or wellness influencers as virtual coaches to give users more excitement and motivation.

VII. Market Leaf as a low-impact tracker

Bellabeat should market the Leaf device as a low-impact fitness tracker suitable for people who prefer less strenuous activities. This approach could appeal to a broader audience, including individuals who are just starting with fitness tracking, who might not have a strenuous workout routine, or who want to improve their health by gradually increasing their physical activity levels.

VIII. Offer personalized sleep guidance

Bellabeat should deliver personalized guidance to users to improve their sleep routine. For example, the Bellabeat app could remind users to prioritize their sleep on days when they have a shorter bedtime and sleep duration. It could also provide tips and advice on how to balance busy schedules with sufficient sleep to help users maintain a consistent sleep pattern throughout the week.

DISCLAIMER: My findings and recommendations are based on Bellabeat's available data, which is subject to several limitations I indicated earlier. The stakeholders are advised to interpret these with caution and consider the limitations when making business decisions.

---

*This project was completed as part of the requirements for the **Google Data Analytics Professional Certificate** program.*

Add a comment...

Most relevant

Jhermien Paul Alejandria Author (edited) 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Act...

A huge shoutout to all of you who have been sending me kind messages and comments about my previous case study! It means the world to me and I can't thank you enough. Some of you have asked if I've done any other case studies, specifically the Bellabeat one, or if I've used R before. Well, the good news is that I've finally had the time to put together an article about my experience creating the Bellabeat case study with R! I'd love to hear your thoughts on it, so feel free to send me your feedback. Once again, thank you all so much for your amazing support! 💜

Like · 1 | Reply | 901 impressions

Samarth Singhal • 1st 2y
Data Analyst Apprentice at Google

Great visualization this is Jhermien Paul Alejandria

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Hi Samarth! Thank you very much for your kind words! 😊

Like | Reply | 222 impressions

Ryan Johnson • 1st 2y
Planning Coordinator | Developing Field Service Consultant

Currently working on this one myself. You have set the bar HIGH! 👏

Love · 2 | Reply · 1 reply

Jhermien Paul Alejandria Author 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Thanks Ryan! I am humbled and grateful to read your feedback. I'm very excited about your project, and I'm sure you'll do an amazing job! If you need any help or guidance along the way, feel free to reach out to me. Best of luck! 💙

Like | Reply | 325 impressions

Daina Ulmer • 1st (edited) 2y
Aspiring Web Developer

Excellent work! I love how neat, organized, and easy to understand this is. The colors are not over the top and confusing either. Keep it! Great work!

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Hi Daina! Thanks for taking the time to leave such a thoughtful comment! I'm thrilled to hear that you appreciate how I created my dashboard. It means a lot to me. 😍

Like · 1 | Reply | 65 impressions

M Luthfi Al Ghifari • 1st 2y
Lead Full-Stack Developer | WordPress & Technical SEO Specialist

Awesome work,very impressive 🙌

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Wow! Thank you so much, M Luthfi Al Ghifari! 😍

Like | Reply | 153 impressions

Johannes Moon • 1st
Reporting Engineer | Banking industry | Process Engineer | ad-hoc | SQL | ...
2y

Your work is simply top notch, Jhermien! 👏

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...
2y

Hi there, Johannes! Here you go again. 😂 I can't thank you enough for your continuous support. Your feedback always inspires me to come up with better projects. 😍

Like · 1 | Reply | 316 impressions

Bryan Ubalde • 1st
Quality Assurance Specialist at MicroSource Inc.
2y

Your efforts to produce this excellent work are to be commended. May you maintain on as you have been doing for the duration of your profession. I must admit that I am jealous of your abilities. But because it's an incredible privilege to have you as my colleague and to collaborate with you at some point, I'd like to take this opportunity to wish you the very best of luck. Kudos!

Support · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author
Data Analyst at Royal Caribbean • Transforming Big Data i...
(edited) 2y

Thank you so much for your kind words and support, Bryan Ubalde! I appreciate your encouragement and am grateful to have you as my colleague. I believe that we can all learn from each other and grow together. I look forward to collaborating with you again in the future. Thanks and wishing you all the best! 🤗

Like · 1 | Reply | 280 impressions

Taras Khamardiuk, MS, MBA • 1st
BI & Analytics Specialist | Python Automations for Reporting & Workflows...
2y

Super! Great job!

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...
2y

Hello Taras! Thank you so much! Hoping you and your family are safe there in Ukraine. 😊

Like | Reply | 181 impressions

Jatila Molegoda • 1st
Freelance Data Analyst | Turning Complex Data into Actionable Insights | ...
2y

Amazing work

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...
2y

Thank you Jatila! I'm glad you liked my work. I sincerely appreciate it! 😊

Like | Reply | 247 impressions

Laurence Camillo • 1st
Data Analyst
2y

Super awesome! ✨ 💯

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...
2y

Thank you so much Laurence! This comment means a lot! Good luck on your data analytics journey. 😊

Like | Reply | 205 impressions

Vishwas Kshirsagar • Following 2y
Data Analytics & Science | I Help You Land Your Dream Data Job

Keep up the good work Jhermien Paul Alejandria

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author 2y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Hi there, Vishwas! Thank you very much. 😊

Like · 1 | Reply | 141 impressions

Joe Unyah • 3rd+ 1y
Certified QuickBooks ProAdvisor | Mentoring | Accounting | Tax Enthusias...

It;s really nice

Love · 1 | Reply · 1 reply

Jhermien Paul Alejandria Author 1y
Data Analyst at Royal Caribbean • Transforming Big Data into Actio...

Thanks Joe!

Like | Reply | 33 impressions

Jhermien Paul Alejandria

Data Analyst at Royal Caribbean • Transforming Big Data into Actionable Insights & Strategies PH

More articles for you

UK Telephonic Lifestyle Survey: Understanding the Nation's Habits and Preferences

Vanggharamaa BPO

3

A Knowledgeable Employee Sells More Product

CNHR Magazine

4

How Can A Wellness Technology Company Play it Smart (With R)

Sharon Mateka

2