Scenario

You are a junior data analyst working on the marketing analyst team at 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, co-founder and Chief Creative Officer of Bellabeat believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide the company's marketing strategy. You will present your analysis to the Bellabeat executive team along with your high-level recommendations for Bellabeat's marketing strategy.

The analysis Process is broken down into 6 steps

- Ask
- Prepare
- Process
- Analyze
- Share
- Act

1. Ask

During this phase, key tasks are identifying the business task and considering key stakeholders.

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

Stakeholders:

- Urška Sršen - Bellabeat co-founder and Chief Creative Officer
- Sando Mur - Bellabeat cofounder and a key member of Bellabeat executive team
- Bellabeat Marketing Analytics team

2. Prepare

This phase is where we identify the type of data we require and then collect it. After this, we check the format of the data, verify its credibility, and then store it.

2.1 Dataset Used

We are using data of Fitbit users obtained from Kaggle. The dataset was 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.

2.1 More Information about Dataset

This according to the author is open-source data that can be copied, modified, and distributed. There are 15 csv tables within the files and I have used 5 tables for this analysis. The tables used are

- dailyActivity_merged
- hourlyCalories_merged
- hourlySteps_merged
- sleepDay_merged
- weightloggingsfo_merged

Details about columns within these tables have been compiled into this sheets.

2.2 Credibility of the Data

Due to the limitation of size (30 users) and not having any demographic information we could encounter a sampling bias. Although the metadata indicates that 30 individuals participated in this survey, the dataset surprisingly contains information on 33 individuals. Additionally, the dataset lacks clear descriptions for each column, posing a challenge for data interpretation

2.3 Assumption

We have made the assumption during analysis that the column sedentaryminutes indicates the time the watch was not worn or the person is idle.

3. Process

This is the phase where we Check the data for errors, choose the tools we are going to use for analysis, transform the data so you can work with it effectively, and document the cleaning process.

The tools that I have used for this analysis are

- PostgreSQL
- Microsoft Power BI.

On initially viewing the table in Excel I noticed that the date column requires formatting.

3.1 Importing the data

I imported all the tables that I am using into PostgreSQL using the following queries.

```
-- importing daily activities table
drop table if exists daily_activities;
CREATE TABLE daily_activities (
  Id BIGINT ,
  ActivityDate DATE,
  TotalSteps INT,
  TotalDistance FLOAT,
  TrackerDistance FLOAT,
  LoggedActivitiesDistance FLOAT,
  VeryActiveDistance FLOAT,
  ModeratelyActiveDistance FLOAT,
  LightActiveDistance FLOAT,
  SedentaryActiveDistance FLOAT,
  VeryActiveMinutes INT,
  FairlyActiveMinutes INT,
  LightlyActiveMinutes INT,
  SedentaryMinutes INT,
  Calories INT
);
copy daily_activities(Id, ActivityDate, TotalSteps, TotalDistance,
TrackerDistance,
       LoggedActivitiesDistance, VeryActiveDistance,
       ModeratelyActiveDistance, LightActiveDistance,
       SedentaryActiveDistance, VeryActiveMinutes, FairlyActiveMinutes,
       LightlyActiveMinutes, SedentaryMinutes, Calories)
from 'C:\Datasets\Fitabase Data 4.12.16-5.12.16\Daily_Activites.csv'
delimiter ','
csv header
select * from daily_activities
--importing sleepDay_merged table
drop table if exists Sleep;
create table Sleep (
Id BIGINT,
Day TIMESTAMP,
Total_Sleep_Records INT,
Total_Minutes_Asleep INT,
Total_Time_inbed INT
);

copy Sleep( Id, Day, Total_Sleep_Records,Total_Minutes_Asleep,
```

```
Total_Time_inbed)
from 'C:\Datasets\Fitabase Data 4.12.16-5.12.16\sleepDay_merged.csv'
delimiter ','
csv header;
select * from Sleep
-- importing weightLog_info table
drop table if exists weight;
create table weight(
id bigint,
date timestamp,
Weight_kg float,
weight_pounds float,
Fat float,
BMI float,
is_manual_report boolean,
logid bigint);

copy weight(id, date, weight_kg, weight_pounds, fat, bmi,
is_manual_report, logid)
from 'C:\Datasets\Fitabase Data
4.12.16-5.12.16\weightloggingsfo_merged.csv'
delimiter ','
csv header
-- importing hourlysteps_merged table
Create table hourly_steps(
Id bigint,
Activity_hour timestamp,
total_steps int);

copy hourly_steps(Id, Activity_hour, total_steps)
from 'C:\Datasets\Fitabase Data 4.12.16-5.12.16\hourlySteps_merged.csv'
delimiter ','
csv header

select * from hourly_steps

-- importing hourlyCalories_merged table
Create table hourly_calories(
Id bigint,
Activity_hour timestamp,
calories int);

copy hourly_calories(Id, Activity_hour, calories)
from 'C:\Datasets\Fitabase Data
4.12.16-5.12.16\hourlyCalories_merged.csv'
delimiter ','
```

```
csv header;
select * from hourly_calories
```

3.2 Cleaning and Transforming the Data

After importing the dataset the next step is to analyze and clean the data. On analyzing It was clear that there are 33 distinct IDs in the daily_activities table but there were only 24 and 4 distinct IDs in sleep and weight data respectively. We will be coming to this point later in the analysis.

The Weight table had a column that had 97% null values due to this reason it was decided to drop that column. None of the other tables had any null values.

The sleep table had Duplicates which were verified and then removed. No other tables had any duplicate values.

All these queries are provided below

```
Select Distinct(id) from activities
Select Distinct(id) from sleep
Select Distinct(id) from weight


--Cleaning the weight table
Select Id, date, weight_kg, weight_pounds, bmi, is_manula_report, logid
from weight

-- Cleaning sleep table
-- Identify duplicates using a CTE
WITH CTE AS (
    SELECT id, day, total_sleep_records, total_minutes_asleep,
total_time_inbed,
           ROW_NUMBER() OVER (PARTITION BY id, day ORDER BY id) AS
row_num
    FROM sleep
)
-- Delete rows where row_num is greater than 1
DELETE FROM sleep_cleaned
WHERE (id, day, total_sleep_records, total_minutes_asleep,
total_time_inbed) IN (
    SELECT id, day, total_sleep_records, total_minutes_asleep,
total_time_inbed
    FROM CTE
    WHERE row_num > 1
```

```
);

-- Verify that duplicates are removed
SELECT id, day, COUNT(*) as count_rows
FROM sleep_cleaned
GROUP BY id, day
HAVING COUNT(*) > 1;
```

3.3 Combining Tables

On analysis, it was clear that having 2 tables that contain hourly information about calories spent and steps taken is redundant. So it was decided to combine these tables for ease of analysis.

The queries used were

```
SELECT hs.id, hs.activity_hour, hs.total_steps, hc.calories
    FROM hourly_steps AS hs
    INNER JOIN hourly_calories AS hc
    ON hs.id = hc.id AND hs.activity_hour = hc.activity_hour
```

4. Analyze

This is the phase of analysis where we aggregate the data, organize and format it, create plots and visualizations, perform calculations, and finally find patterns and relationships within the data.

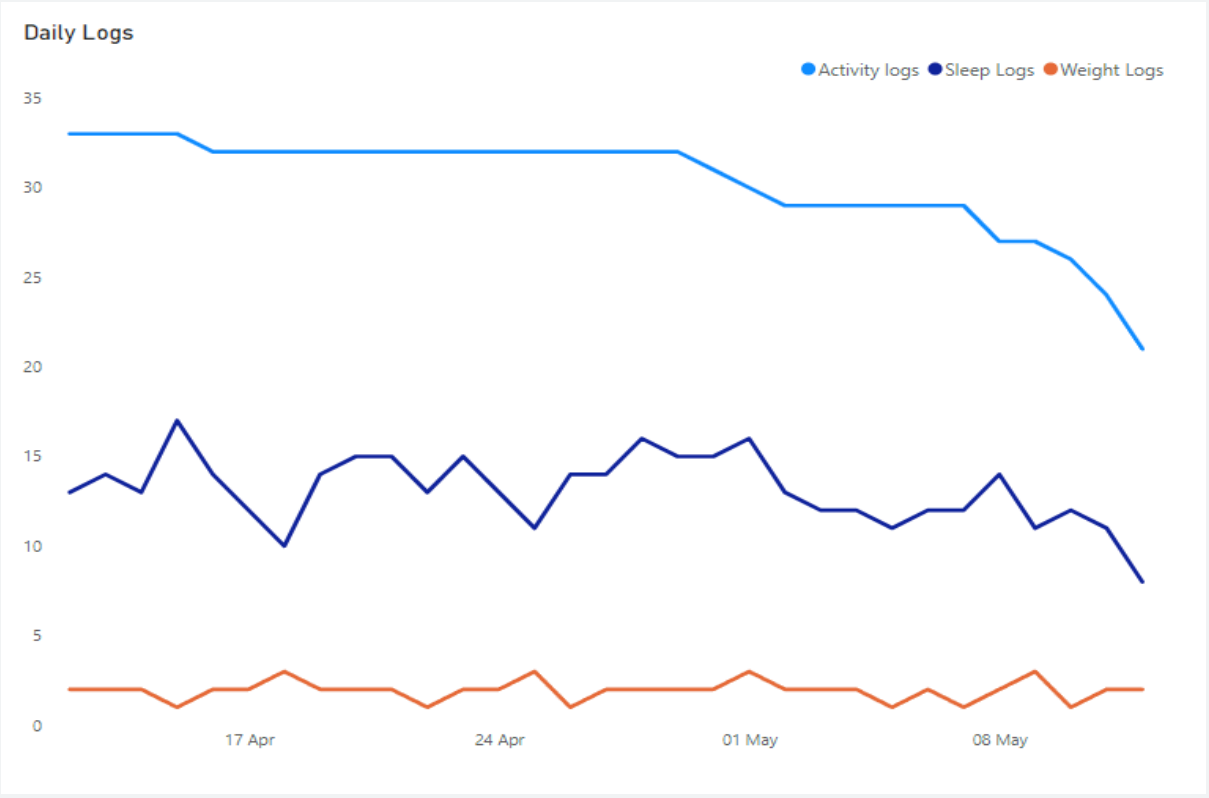
This part of the report unfolds like a journey, with each segment revealing new insights. Each discovery is backed by thorough calculations and vibrant visuals, all powered by the intricate web of code running beneath the surface.

4.1 Daily Loggings

So my first step was to find the number of times each individual tracker logged in to measure different values. So I created a table that provides us with information about the average, maximum, and minimum number of loggings for each individual as per the activities, sleep, and weight tables.

The plot comparing the loggings for each table during the period of the survey (one month) is also provided.

Table Name	Average no.of loggings	Maximum no. of loggingss	Minimum no.of loggingss
Activities table	30.32	33	21
Sleep Table	13.13	17	8
Weight Table	2	3	1



- There is a clear discrepancy in the no of loggings for daily activities, sleep, and weight. Of the 33 individuals, only 13 individuals on average have sleep loggings, and only 2 individuals have logged daily on average to log their weights.
- Only 4 individuals have logged in their weight. We should find a way to integrate the scale into the Bellabeat app so that every time they weigh themselves it gets registered.
- There is a dip in daily activities logging and sleep logging towards the end of the month.

The query used to obtain the table used for the above visualization is

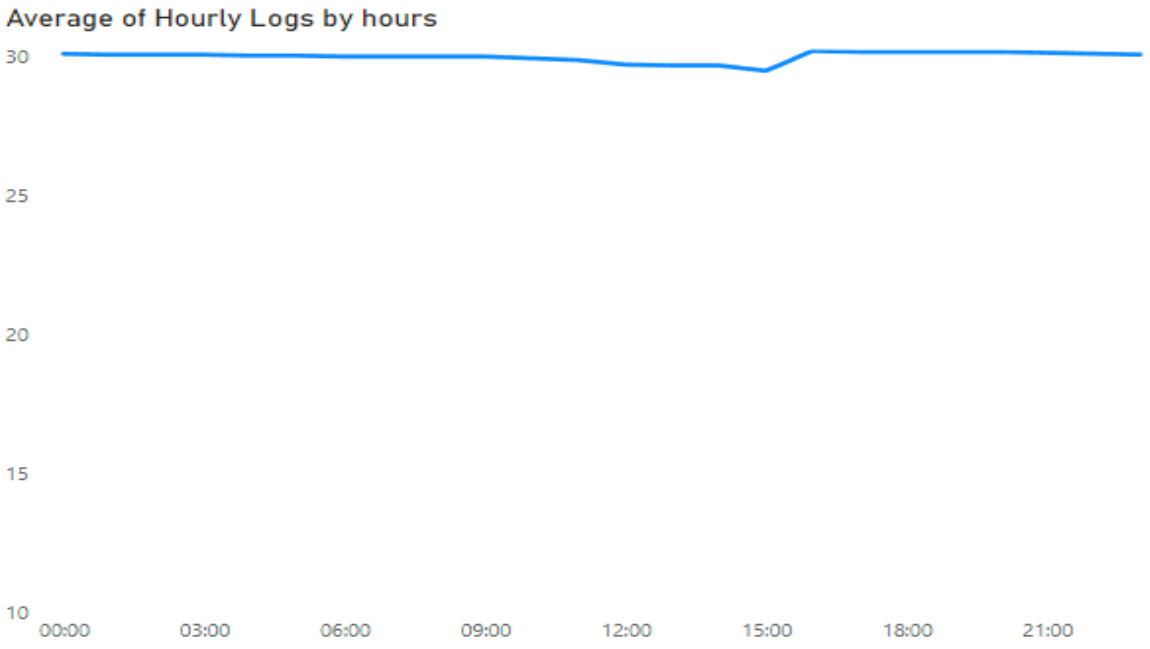
```
-- logs for daily activities
with cte as (select activitydate, count(row_counts) as no_of_logs
      from(select *, row_number() over(partition by id, activitydate)
```

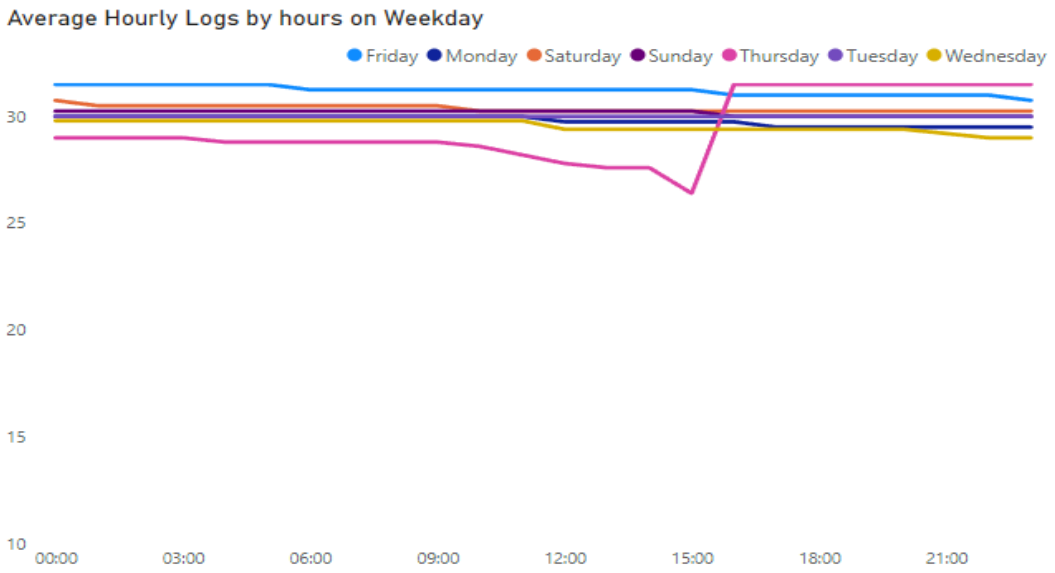
```
row_counts
             from daily_activities)
group by activitydate)

select round(avg(no_of_logs),2) as average_logs, min(no_of_logs) as
min_logs, max(no_of_logs) as max_logs
from cte
-- logs of sleep
SELECT
    round(AVG(num_of_logs),2) AS average_loggingss,
    MAX(num_of_logs) AS max_loggingss,
    MIN(num_of_logs) AS min_loggingss
FROM (
    SELECT
        day,
        COUNT(row_counts) AS num_of_logs
    FROM (
        SELECT
            id,
            day::date AS day,
            total_minutes_asleep,
            ROW_NUMBER() OVER(PARTITION BY id, day) AS row_counts
        FROM
            sleep_cleaned
    ) a
    GROUP BY
        day
) b;
-- No of weight logs per day
select round(avg(num_of_logs), 1), min(num_of_logs), max(num_of_logs)
from (
    select date, count(row_no) as num_of_logs
    from (
        select id, date, row_number() over(partition by id, date) as
row_no
        from weight_cleaned
    ) as subquery1
    group by date
) as subquery2;
```

4.2 Hourly loggings

Here we uncover the natural cadence of individuals' steps and the energetic pulse of calories burned, hour by hour. We've crafted two insightful plots: one illuminates the peak logging

hours, offering a glimpse into when activity surges. The second, enriched by a novel weekday column extracted from our dates dataset, delves deeper into the hourly logging trends across each day of the week.





- From the first plot, it is clear that the average no of loggings is almost the same almost all the time except for a slight dip at 15:00 hours.

- The second plot shows except for Thursdays all days have almost the same number of hourly loggings. This dip in hourly loggings on Thursday at 15:00 hrs might be the reason for the dip in overall loggings at 15:00 hrs

The queries used for generating the table that gave us the above plots are

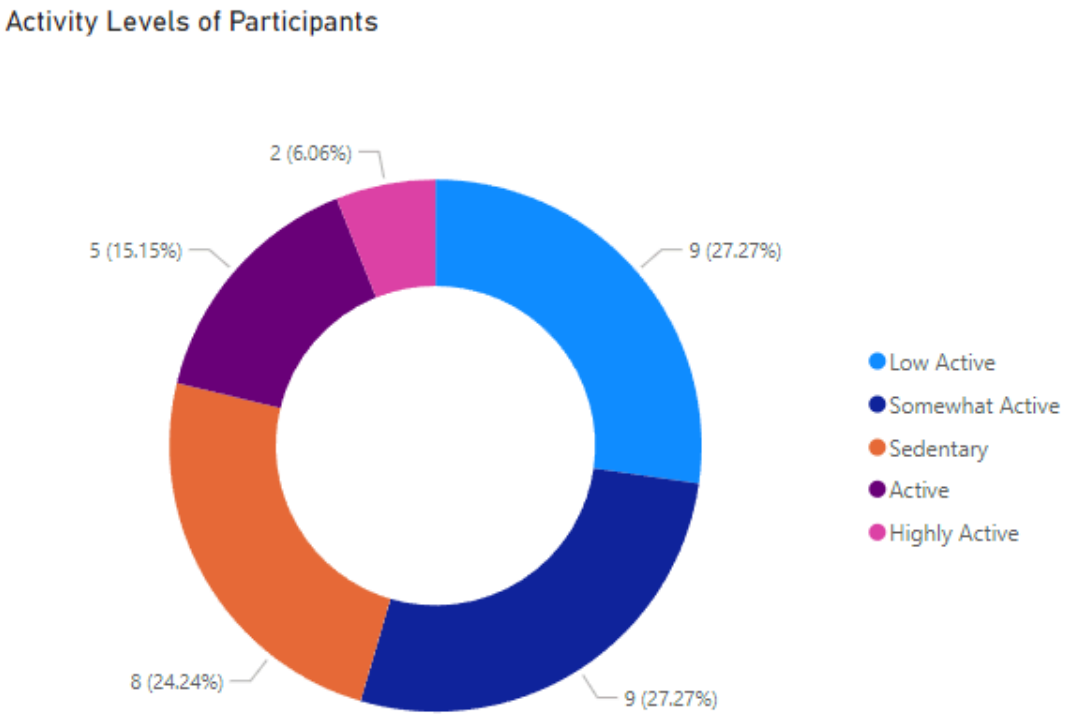
```
WITH cte AS (
    SELECT hs.id, hs.activity_hour, hs.total_steps, hc.calories
    FROM hourly_steps AS hs
    INNER JOIN hourly_calories AS hc
    ON hs.id = hc.id AND hs.activity_hour = hc.activity_hour
),
cte1 AS (
    SELECT activity_hour, COUNT(row_no) AS count
    FROM (
        SELECT id, activity_hour, total_steps, calories,
               ROW_NUMBER() OVER (PARTITION BY id, activity_hour ORDER
BY id) AS row_no
        FROM cte
    ) subquery
    GROUP BY activity_hour
)

SELECT date,
       CASE
           WHEN weekday = 0 THEN 'Sunday'
           WHEN weekday = 1 THEN 'Monday'
           WHEN weekday = 2 THEN 'Tuesday'
           WHEN weekday = 3 THEN 'Wednesday'
           WHEN weekday = 4 THEN 'Thursday'
           WHEN weekday = 5 THEN 'Friday'
           WHEN weekday = 6 THEN 'Saturday'
       END AS weekday,
       Hours,
       hourly_logs
FROM (
    SELECT activity_hour::date AS date,
           activity_hour::time AS Hours,
           EXTRACT(DOW FROM activity_hour) AS weekday,
           count AS hourly_logs
    FROM cte1
) subquery
ORDER BY date;
```

4.3 Segmenting users on activity

According to studies by Tudor-Locke and Bassett, and Tudor-Locke et al. (2011), individuals can be classified into different activity levels based on the number of steps they take daily. These classifications are:

- **Sedentary:** Fewer than 5,000 steps per day
- **Low Active:** 5,000 to 7,499 steps per day
- **Somewhat Active:** 7,500 to 9,999 steps per day
- **Active**: 10,000 - 12,499 steps per day
- **Highly Active:** 12,500 steps or more per day

Now using this information I segmented the table and then plot a visualization using the below query



- According to the plot, a significant 51.5% of individuals fall into the sedentary category, taking fewer than 5,000 steps per day. This indicates that more than half of the population is not engaging in sufficient physical activity, potentially increasing their risk for various health issues related to inactivity.
- In contrast, only 21% of individuals achieve highly active status by completing more than 10,000 steps daily. This minority demonstrates a commendable level of physical activity that is associated with better health outcomes, as highlighted in the studies by Tudor-Locke et al. (2011).

```sql
select id, round(avg(totalsteps), 2) as avg_daily_steps,
case
    when avg(totalsteps) < 5000 then 'Sedentary'
    when avg(totalsteps) between 5000 and 7499 then 'Low Active'
    when avg(totalsteps) between 7500 and 9999 then 'Somewhat Active'
    when avg(totalsteps) between 10000 and 12499 then 'Active'
    else 'Highly Active'
end as activity_level
from daily_activities
group by id
order by id;
```

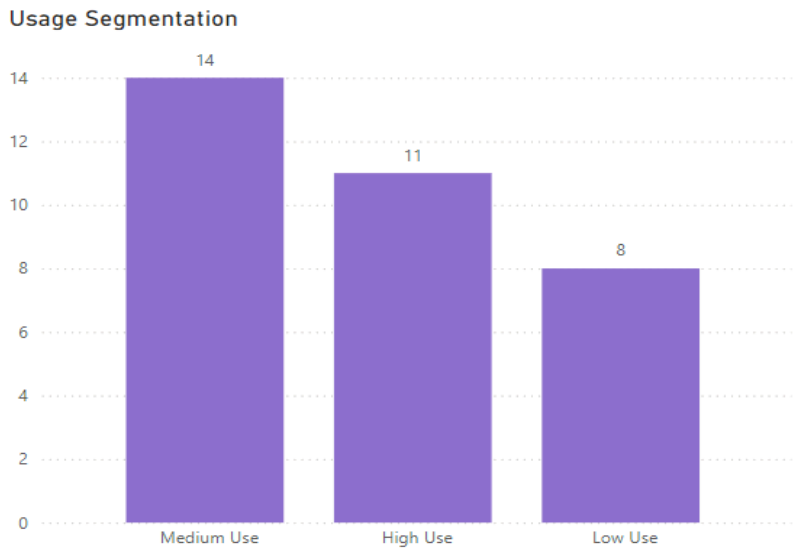
4.4 Segmenting Users on Use of Device

n the next step of our analysis, we aimed to determine the duration of device usage for each individual. To achieve this, we introduced a new column that calculated the total minutes the device was actively used. This was based on the assumption that sedentary minutes represented the time the device was not in use.

After calculating the total active minutes, we segmented the data into four distinct usage categories:

- **Low Use:** Less than 170 minutes
- **Medium Use:** 170 to 270 minutes
- **High Use:** More than 270 minutes

These thresholds were carefully chosen after a thorough examination of the distribution of total active minutes. This categorization allows us to clearly distinguish between different levels of device engagement among users, providing valuable insights into user behavior and engagement patterns.



- An impressive **42%** of users fall into the "Medium Use" category, utilizing the device between 170 and 270 minutes. This highlights a substantial engagement level among a significant portion of users.
- Only **18%** of users are in the "Low Use" category, with usage below 170 minutes. This small percentage suggests that the Wearable effectively engages the vast majority of users beyond minimal interaction.
- The point ot be remembered here is that Users were segmented using the 33rd and 67th percentile values of `minutes_worn`, a method designed to ensure an even and insightful distribution of usage categories.

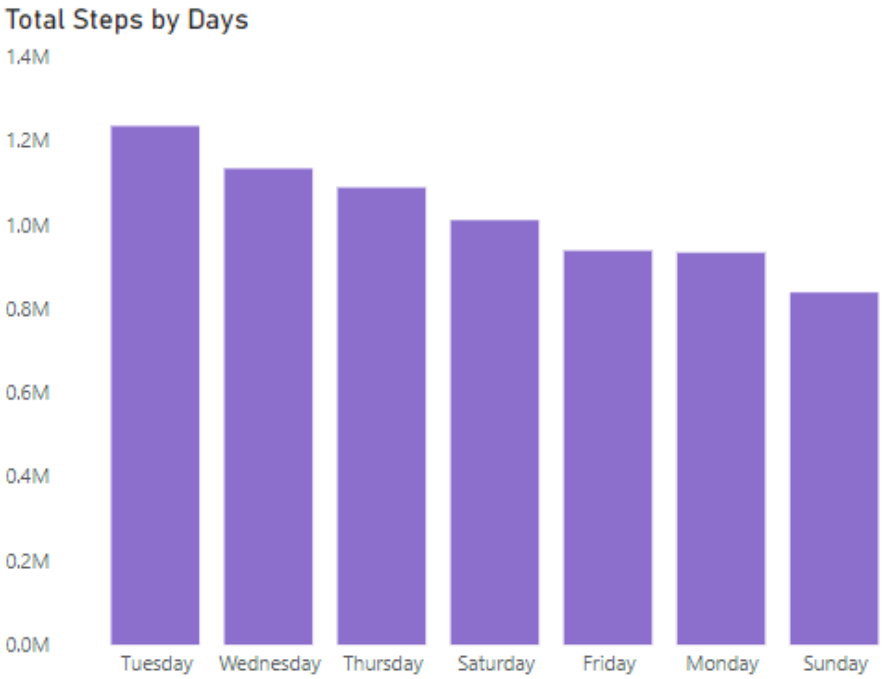
```
The query used for generating the table is
WITH cte AS (
    SELECT *,
           veryactiveminutes + fairlyactiveminutes +
lightlyactiveminutes AS minutes_worn
    FROM daily_activities
),
cte2 AS (
    SELECT id,
           ROUND(AVG(minutes_worn), 2) AS avg_minutes_worn,
           MAX(minutes_worn) AS max_minutes_worn,
           MIN(minutes_worn) AS min_minutes_worn
    FROM cte
    GROUP BY id
    ORDER BY id
)
SELECT
    id,
    avg_minutes_worn,
    max_minutes_worn,
    min_minutes_worn,
    CASE
        WHEN avg_minutes_worn <= 170 THEN 'Low Use'
        WHEN avg_minutes_worn > 170 AND avg_minutes_worn <= 270 THEN
'Medium Use'
        ELSE 'High Use'
    END AS use_of_wearable
FROM cte2;
```

4.5 Relation Between Variables

a. Steps vs weekdays

To analyze patterns in the total steps taken by people on different weekdays, we will start by creating a new column in the daily activities table. This column will extract the day of the week from the date column. Next, we will group the data by this new weekday column to summarize the total steps taken on each day. Finally, we'll use this grouped data to build a

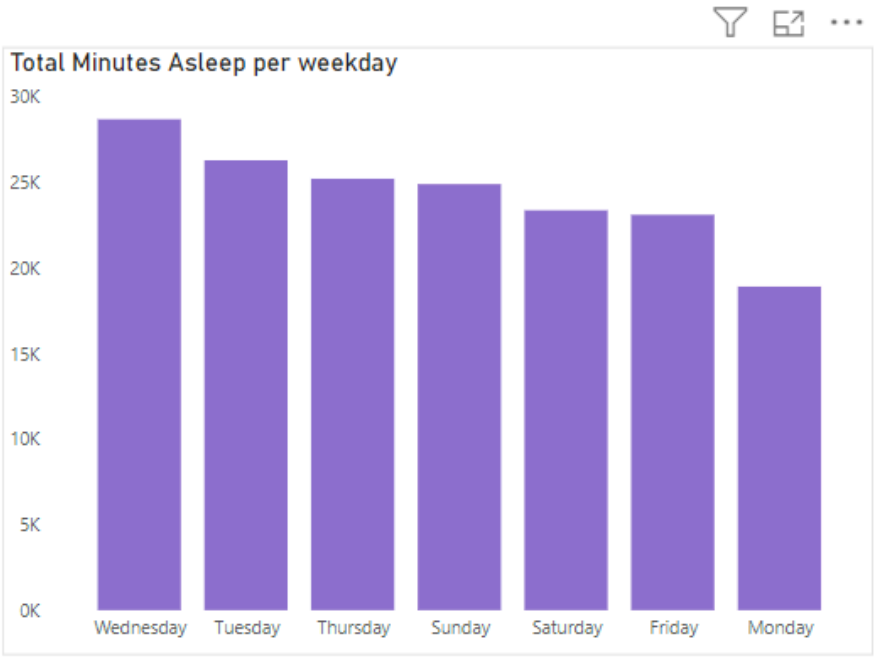
plot in Power BI, allowing us to visually identify any trends or patterns in step activity throughout the week.



- Individuals exhibit significantly higher step counts in the middle of the week, with **Tuesdays and Wednesdays** showing the most activity. This trend suggests that people are most active during these days, likely due to midweek routines and work commitments.
- **Sundays** consistently record the lowest number of steps, indicating that many people use this day as a rest day. This sharp decline in activity underscores the importance of rest and recovery in weekly routines.

b. Sleep vs Weekdays

Similar to what we did previously we use the already generated weekday column for this analysis. Grouping the table with respect to id and then importing the table back into power bi to plot a graph.



- Similar to activity patterns, sleep duration peaks midweek, with **Wednesdays and Tuesdays** showing the highest average sleep times. This indicates that people may prioritize rest during these days to recharge for the rest of the week.
- **Mondays** consistently show the least amount of sleep, suggesting that the start of the week is particularly challenging for maintaining adequate rest, possibly due to the transition from the weekend or the demands of starting a new week.

The queries for both the tables are provided below

```
- for activity table
with cte as
      (select id, activitydate, totalsteps, totaldistance, calories,
extract(DOW from activitydate) as weekday
      from daily_activities)

      select id, activitydate, totalsteps, totaldistance, calories,
                  case
                        when weekday = 0 then 'Sunday'
                        when weekday = 1 then 'Monday'
                        when weekday = 2 then 'Tuesday'
                        when weekday = 3 then 'Wednesday'
                        when weekday = 4 then 'Thursday'
                        when weekday = 5 then 'Friday'
                        when weekday = 6 then 'Saturday'
                  end as weekday
      from cte
- for activity and sleep minutes
```

```sql
WITH cte AS (
    SELECT *
    FROM daily_activities
    WHERE id IN (SELECT DISTINCT id FROM sleep_cleaned)
),
cte2 AS (
    SELECT cte.id, cte.activitydate, cte.totalsteps, cte.calories,
s.total_minutes_asleep
    FROM cte
    INNER JOIN sleep_cleaned s ON cte.id = s.id AND cte.activitydate =
s.day
),
cte3 AS (
    SELECT id, activitydate, total_minutes_asleep, EXTRACT(DOW FROM
activitydate) AS weekday
    FROM cte2
)
SELECT id,
    CASE
        WHEN weekday = 0 THEN 'Sunday'
        WHEN weekday = 1 THEN 'Monday'
        WHEN weekday = 2 THEN 'Tuesday'
        WHEN weekday = 3 THEN 'Wednesday'
        WHEN weekday = 4 THEN 'Thursday'
        WHEN weekday = 5 THEN 'Friday'
        WHEN weekday = 6 THEN 'Saturday'
    END AS weekday,
    AVG(total_minutes_asleep) AS minutes_slept
FROM cte3
GROUP BY id, weekday
ORDER BY id, weekday;
```

5. Share Phase

This is the phase where we share our recommendation to the stake holders on the basis of the analysis we did.

The dataset we used for this analysis has a lot of limitations in a lot of cases it dosent give all the information about different variables present in the data. Also the dataset has a very small sample size.Considering all these limitations these are the recommandations we to make to improve **Bellabeat app** and the **Leaf wearable.**

- To enhance user experience and accuracy in tracking daily weight, we recommend integrating the scale directly with the Bellabeat app. This seamless connection will

allow customers to automatically log and monitor their weight data in real-time. Our analysis revealed significant discrepancies in daily weight records, highlighting the need for a more reliable and consistent tracking system.

- Make the Leaf wearable more comfortable to wear so people can track their sleep as well.
- Implement a rewarding system within the Bellabeat app to incentivize users to achieve their daily step goals. This gamification element will motivate users to stay active and engaged with the app and their health goals.
- There is **18%** of customers who have a low daily usage, we can increase this by making these changes in to the Leaf wearable
    - Make it more comfortable to wear
    - Increase its water and dust resistance
    - Increase battery life
    - Make the UI accessible to all customers

By implementing these measures, Bellabeat can significantly enhance the **Leaf wearable's** and the **Bellabeat app's** appeal and usability.