Introduction

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

We are going 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 we discover will then help guide marketing strategy for the company.

We will follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and act.

1. Ask:

1.1 Business Task:

Analyze non-Bellabeat smart device data to identify trends and patterns in-order to gain insights and help the marketing team make the right decisions and plan strategies.

Stakeholders:

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer.
- Sando Mur: Bellabeat's cofounder; key member of the Bellabeat executive team.
- Bellabeat marketing analytics team.

2. Prepare:

1.1 Data source:

We are using a dataset from Kaggle, made available through Mobius. And it is titled 'Fitbit Fitness Tracker Data'.

1.2 Accessibility and privacy of data:

The dataset we are using is confirmed to be open-source, as it is licensed as 'CC0: Public Domain', which gives all rights to the extent allowed by law, such as copying, modifying and distributing, even for commercial purposes.

1.3 Metadata:

Our dataset is organized by several parts, in which all of them are stored in a CSV format, with the following each file's name and it's corresponding description:

- dailyCalories_merged: Daily calories of 33 users, over 31 days.
- dailyIntensities_merged: Daily logs of intensity performed by 33 users over 31 days, with the type of intensity divided by 4 groups, and measured in time and distance.
- dailySteps_merged: Records of 33 user daily steps taken, over 31 days.
- dailyActivity_merged: Merged version of the above datasets.
- heartrate_seconds_merged: Records of heart rate from 7 users, measured in date and time in minutes and seconds. Value column unrecognizable.
- hourlyCalories_merged: Hourly calories burned logs of 33 users, over 31 days.
- hourlyIntensities_merged: Hourly logs of intensity performed by 33 users over 31 days, with the type of intensity by time and distance. Alongside with the average intensity.
- hourlySteps_merged: Records of 33 user hourly steps taken, over 31 days.
- minuteCaloriesNarrow_merged: Hourly calories burned of 33 users, over 31 days. Stored in a long format.
- minuteCaloriesWide_merged: Hourly calories burned of 33 users, over 31 days. Stored in a wide format.
- minuteIntensitiesNarrow_merged: Hourly logs of intensity performed by 33 users over 31 days, with the type of intensity by time and distance. Stored in a long format.
- minuteIntensitiesWide_merged: Hourly logs of intensity performed by 33 users over 31 days, with the type of intensity by time and distance. Stored in a wide format.
- minuteMETsNarrow_merged: Ratio of energy used on a physical activity, compared to energy used on rest time of 33 users, counted by minute, over 31 days.

- minuteSleep_merged: Sleep logs of 24 users by minute, over 31 days. Value column unknown.
- minuteStepsNarrow_merged: 33 user steps taken by minute. Stored in a long format.
- minuteStepsWide_merged: 33 user steps taken by minute. Stored in a wide format.
- sleepDay_merged: Each user sleep logs, such as sleep time, total count of sleep records, total minutes slept and total minutes in bed.
- weightLogInfo_merged: Each user weight logs by date and time, such as weight in KG, weight in pounds, fat percentage, body mass index, and a column with inputs indicating how each log was recorded ("Manual", "Not manual").

1.4 Data integrity:

The dataset we are using is original, comprehensive and cited. However, it's not current, since it was collected on 2016. Also, it's sample size is small (30 users), which could result to a sampling bias. But we can still use it to gain insights and apply them on a further analysis.

3. Process

3.1 Tools we are using:

For this case study, we are going to use mostly SQL and a bit of Excel, and that's because our data size is relatively big.

3.2 Cleaning data:

Before importing our data into SQL servers, first we are going to do some cleaning on Excel, so we are going to open our csv files that we need on Excel, and they are as following:

- Daily_activity.
- Daily_sleep
- Hourly_steps
- Weight_log

After that, we shall use ‘Remove Duplicates’ , and so we found that there are no duplicates, Except for the daily_sleep file where we found 3 of them. Then, we will do some basic filtering to check if any of the columns are corrupt, all of them are looking good. We would also clean our data from additional spaces on any of the records with the TRIM() functions. Additionally, there was a problem with importing the date column into SQL later on, as it had ‘PM’ and ‘AM’ labels on Excel which refused to be rid of only after applying the TEXT() function to extract the date and time without ‘PM’ or ‘AM’. We would also change column names to make them appear in the same format later on when we are working on SQL.

3.3 Importing datasets:

We are going to open our SQL RDBMS and for that we chose postegresql, we create a schema named ‘project’ then proceed with creating tables as following:

```
DROP TABLE IF EXISTS project.daily_activity;


CREATE TABLE IF NOT EXISTS project.daily_activity
(
    id bigint,
    activity_date date,
    total_steps integer,
    total_distance numeric,
    tracker_distance numeric,
    logged_activities_distance numeric,
    very_active_distance numeric,
    moderately_active_distance numeric,
    light_active_distance numeric,
    sedentary_active_distance numeric,
    very_active_minutes integer,
    fairly_active_minutes integer,
    lightly_active_minutes integer,
    sedentary_minutes integer,
```

```
    calories integer
);

-- Table 2: project.hourly_steps

DROP TABLE IF EXISTS project.hourly_steps;

CREATE TABLE IF NOT EXISTS project.hourly_steps
(
    id bigint,
    activity_hour timestamp without time zone,
    step_total integer
);

-- Table 3: project.sleep_day

DROP TABLE IF EXISTS project.sleep_day;

CREATE TABLE IF NOT EXISTS project.sleep_day
(
    id bigint,
    sleep_day date,
    total_sleep_records integer,
    total_minutes_asleep integer,
    total_time_in_bed integer
);
```

```
-- Table 4: project.weight_log

DROP TABLE IF EXISTS project.weight_log;

CREATE TABLE IF NOT EXISTS project.weight_log
(
    id bigint,
    date date,
    weight_kg numeric,
    weight_pounds numeric,
    fat numeric,
    bmi numeric,
    ismanual boolean
);
```

After that we can import our csv files into the tables.

```
COPY project.daily_activity FROM 'C:\FILES\dailyActivity_merged.csv' DELIMITER ',' CSV HEADER;

COPY project.hourly_steps FROM 'C:\FILES\hourlySteps_merged.csv' DELIMITER ',' CSV HEADER;

COPY project.sleep_day FROM 'C:\FILES\sleepDay_merged.csv' DELIMITER ',' CSV HEADER;

COPY project.weight_log FROM 'C:\FILES\weightLogInfo_merged.csv' DELIMITER ',' CSV HEADER;
```

4. Analyze:

4.1 Number of users tracking their sleep/weight:

Number of unique users tracking their sleep:

```
SELECT
    COUNT(DISTINCT id)
FROM
    project.sleep_day
```

Output = 24

Number of users tracking their weight:

```
SELECT
    COUNT(DISTINCT id)
FROM
    project.weight_log
```

Output = 8

4.2 Classify each user by the amount of daily device usage:

We will classify our users per the number of days they used the devices for, By that we can determine whether most of our sample are using their devices on a daily basis or not.

First, we write a query to count each user's the number of days they used the device as following:

```
SELECT
        id,
        count(DISTINCT activity_date) days_used
FROM
        project.daily_activity
GROUP BY 1
```

Then, we write another query to classify users based on the previous query we wrote, this is where we classify each user into either "High Use" or "Low Use" based on the number of days they used the device. If a user used the device for 20 or more days, they are classified as "High Use". Otherwise, they are classified as "Low Use".

```
CREATE VIEW project.device_usage AS
( SELECT
        id,
        days_used,
        CASE WHEN days_used >= 20 THEN 'High Use'
        ELSE 'Low Use'
        END AS usage_type
   FROM
 (
        SELECT
                id,
                count(DISTINCT activity_date) days_used
        FROM
                project.daily_activity
        GROUP BY 1
 ) q
)
```

4.3 Type of users per activity level:

We want to determine each user's activity lifestyle by counting the average number of steps they take daily, we are going to classify each user per the article:

www.medicinenet.com/how_many_steps_a_day_is_considered_active/article.htm

And as following the activity level by average daily steps:

- Sedentary: Less than 5,000 steps daily

- Low active: About 5,000 to 7,499 steps daily
- Somewhat active: About 7,500 to 9,999 steps daily
- Active: More than 10,000 steps daily
- Highly active: More than 12,500 steps daily

Firstly, we write a query to fetch the average steps taken by each user.

```
SELECT
        id,
        AVG(total_steps) AS average_steps
    FROM
        project.daily_activity
    GROUP BY 1
```

Secondly, we use that query to create a view to classify each user by the amount of average steps taken.

```
CREATE VIEW project.activity_level AS
    (SELECT
    id,
    CASE WHEN avg < 5000 THEN 'Sedentary'
        WHEN average_steps BETWEEN 5000 AND 7499 THEN 'Low Active'
        WHEN average_steps BETWEEN 7500 AND 9999 THEN 'Somewhat Active'
        WHEN average_steps BETWEEN 10000 AND 12499 THEN 'Active'
        WHEN average_steps > 12500 THEN 'Highly Active'
      END AS activity_level
FROM (SELECT
        id,
        AVG(total_steps) AS average_steps
    FROM
```

```
        project.daily_activity
    GROUP BY 1) a )
```

After that, We can perform calculations as following:

```
SELECT
    activity_level,
    count(*)                                    AS user_count,,
    COUNT(*) / SUM(COUNT(*)) OVER() * 100 AS percentage
FROM project.activity_level
GROUP BY 1
ORDER BY 2 DESC
```

We can see that most of our users fall under the sedentary to somewhat active lifestyle, however that will be shown more clearly on a visualization that we will create later.

4.4 Activity and sleep amount by day of the week:

We would like to know how active and how much total sleep each user gets on each day of the week, so we write the following code:

```
CREATE TABLE project.average_SleepAndSteps_ByDay AS
(
    SELECT
        TO_CHAR(activity_date, 'Day') AS day_name,
        AVG(total_steps)                    AS average_steps,
        AVG(total_minutes_asleep)       AS average_sleep
    FROM
        project.daily_activity a
    INNER JOIN project.sleep_day b
    ON a.activity_date=b.sleep_day AND a.id=b.id
    GROUP BY 1
)
```

4.5 Top 10 users by monthly total steps:

We would like to create a monthly leaderboard to determine the top 10 users with the highest total steps taken, and potentially give a reward to these users.

First, we create a temporary table containing the leaderboard for ranks of each user by monthly total steps taken.

```
with leaderboard AS
(
        SELECT
                EXTRACT('month' FROM activity_date) AS month,
                id,
                RANK () OVER(PARTITION BY EXTRACT('month' FROM activity_date)
ORDER BY SUM(total_steps) DESC) AS rank
        FROM
                project.daily_activity
        GROUP BY 1,2
)
```

Now we select only ranks from 1 to 10.

```
SELECT
        a.id,
        b.month,
        b.rank,
        SUM(a.total_steps)
FROM
        project.daily_activity a
JOIN leaderboard b ON a.id=b.id
HAVING RANK <= 10
ORDER BY 2,3
```

4.6 Merging tables:

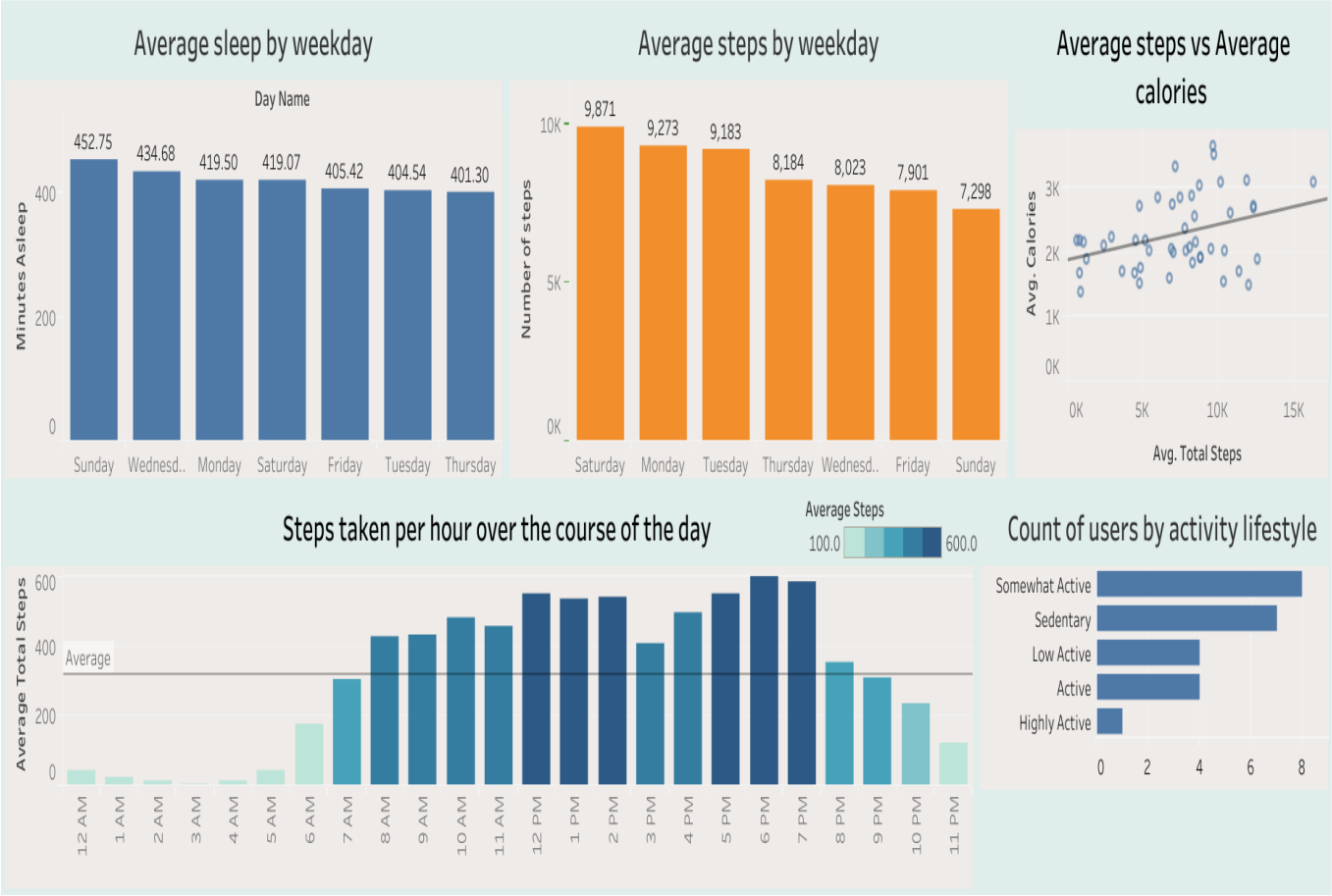
We would like to merge all the code we wrote into one main table if possible, and that's to make it easier for us when importing our work into Tableau for visualizing.

```
CREATE TABLE project.newdaily_activity AS
(
        SELECT
                a.*,
                b.usage_type,
                c.activity_level
        FROM
                project.daily_activity a
      LEFT JOIN
                project.device_usage b ON a.id=b.id
        LEFT JOIN
                project.activity_level c ON a.id=c.id
)
```

5. Share

After exporting our data into Tableau visualization tool, we created a dashboard considering the following:

- Average sleep by weekday.
- Average steps by weekday.
- Average steps vs Average calories.
- Steps taken per hour over the course of the day.
- Count of users by activity lifestyle.



Full link to the dashboard: https://tabsoft.co/3JCueVI

And from that we were able to gain the following insights:

- Average sleep by weekday: We found that average sleep is higher on Sunday and Wednesday, so it seems like day of the week is not a factor for the change of the average total sleep.
- Average steps by weekday: We can notice that average steps are higher on Saturday, as it is considered a day of the weekend, people tend to walk more on that day, probably because they got more free time as it is usual for most people to be off work on Saturday.
- Correlation between average steps and average calories: There is a positive relationship between the two variables, as it is known that when more steps are taken, higher calories are burned along the way.
- Steps taken per hour over the course of the day: We notice that users tend to be active from 12:00 to 14:00 and even more active from 18:00 to 19:00.

- Count of users by activity lifestyle: It is also clear that most of our users fall under the “Sedentary” to “Somewhat active” lifestyle which indicates that they are generally less active than they need to be.

6. Act

As we reached the near end of our case study, we would like to present a few recommendations that may help improve our Bellabeat app, and they are as following:

- As we classified each user by the activity level they performed, we found that most of them are less active than they must be, therefore Bellabeat team should add a reward system to give back to those who are more active. So, to start with that, we created a monthly leaderboard ranking each user by their total steps taken. Each month, we can give a reward to the top 10 or top 5 ranking users, as a good motive to be more active.
- We can add a notification system to send tips and guides every day to all users who are less active and/or tend to get less sleep. These guidelines may include topics like: Health benefits from being more active, Sleep techniques, Healthy habits, etc...

Based on our results, and from what we mentioned before, this case study is based on a database with limitations, as it has a small sample which could lead to some sort of bias. That being said, I would advise bellabeat team to use their own data for further analysis.