# Documentation Bellabeat

## 1 Summary

This document presents the data analysis process for the Bellabeat company. It is the final project for the Google Data Analysis course. In this project, I applied the six steps of data analysis—ask, prepare, process, analyze, share, and act—to examine smart device usage data and gain insights into how consumers interact with their devices.

## 2 Ask Phase

The final project Will have to answer the following questions:

1. What are some trends in smart device usage?

2. How could these trends apply to Bellabeat customers?

3. How could these trends help influence Bellabeat marketing strategy?

In this part, I focused on thinking about how to answer those questions and learning more about the company.

### *2.1 Company information*

First, I read the case study, which gives a summary of the company. While I was reading, I had another important question: Which of Bellabeat's products benefits the most from data analysis?

Then, I searched for information on the internet. The company's website and its social media accounts gave me more details about its products and marketing strategy.

### *2.2 How to answer those questions*

First, I identified the product's features and the trends that could affect it.

| Product | Functions |
|---|---|
| App | Activity Tracking<br>Sleep Monitoring<br>Hydration Tracking<br>Meditation<br>Menstrual Cycle<br>Pregnancy<br>Postpartum<br>Coaching |
| Leaf | Activity Tracking<br>Sleep Monitoring<br>Stress Prediction<br>Meditation<br>Menstrual Cycle<br>Alarms and reminders |
| Time | |
| Spring | Automatic Hydration Tracking |
| Membership | Personalized Wellness Plans<br>Hormonal Cycle & Reproductive Health Tracking<br>Wellness Content Library<br>Monthly Health Reports |

I will analyze trends related to these functions, then I'm going to identify the products that benefit the most from these trends.

Thereafter, I will analyze how these trends apply to Bellabeat customers, and how trends help influence the Bellabeat marketing strategy.

## 3 Prepare Phase

The recommended dataset is "FitBit Fitness Tracker Data." is published on Kaggle in 2016 under the CC0: Public Domain license, with a usability rating of 8.75. It was published by Möbius, a user with the "Dataset Master" title.

This dataset is divided into two sets, each containing data for one month. Each set includes the following CSV files:

| Table Name | Description | In two tables? |
|---|---|---|
| dailyActivity_merged | Traking daily, Steps, Distance, Intensities, Calories | Yes |

| Table Name | Description | In two tables? |
|---|---|---|
| dailyCalories_merged | dailyCalories | No |
| dailyIntensities_merged | Daily Calories dividing in 4 categories | No |
| dailySteps_merged | Daily Steps | No |
| heartrate_seconds_merged | Heartrate | Yes |
| hourlyCalories_merged | Hourly Calories | Yes |
| hourlyIntensities_merged | Total and average intensity | Yes |
| hourlySteps_merged | Hourly Steps | Yes |
| minuteCaloriesNarrow_merged | Calories burned every minute | Yes |
| minuteCaloriesWide_merged | Calories burned every minute | No |
| minuteIntensitiesNarrow_merged | Intensity counted by minute | Yes |
| minuteIntensitiesWide_merged | Intensity counted by minute | No |
| minuteMETsNarrow_merged | Ratio of the energy you are using in a physical activity compared to the energy you would use at rest | Yes |
| minuteSleep_merged | Log Sleep by Minute | Yes |
| minuteStepsNarrow_merged | Steps tracked every minute | Yes |
| minuteStepsWide_merged | Steps tracked every minute | No |
| sleepDay_merged | Daily sleep | No |
| weightLogInfo_merged | Weight track | Yes |

Certain tables are not present in the March–April dataset. First, the daily data is missing, but we can calculate it using the minute-level data. Second, the "Wide" data is missing, so I will use only the "Narrow" data instead.

To verify that the daily data and hourly data match, pivot tables in Excel were used. The dates were set as rows, and calories and steps were set as values. From this, the average percentage differences were calculated. The summary table is shown below.

| | Calories | Steps |
|---|---|---|
| Max % Difference | 4,719% | 9,303% |
| Min % Difference | 0,003% | 0,013% |
| Avg % Difference | 0,708% | 1,629% |
| % Difference of the Total | 0,613% | 1,463% |

As seen in the table, the differences are not significant enough to conduct a separate analysis for each data set. For this reason, only the data from the hourly tables will be used, as they are present in both files.

In the case of the intensity table, it represents different data in the daily data table and the hourly data table, which is why separate analyses will be conducted. However, the daily table only exists for the May and April files, so only this data will be analyzed.

It was also checked whether the Narrow and Wide formats contain the same data. An attempt was made to apply the same analysis, but in this case, the percentage differences turned out to be significantly higher. In other contexts, it would be advisable to consult the data source to understand the reason behind these discrepancies. However, for the sake of practicality and alignment with the project's goals, only the Narrow format data will be used, as it is available for both months.

Additionally, the daily activity table summarizes the information from the daily calories, daily intensity, and daily steps tables, so it was decided to work exclusively with the daily activity table.

On the other hand, I believe that minute-level data does not provide particularly relevant information for a marketing campaign, except in certain specific cases. For instance, intensity data offers different insights compared to the hourly data. Additionally, the METs and Sleep tables are only available in minute-level format. Finally, the CSV files to be used are presented below.

| Table Name | Observation |
|---|---|
| dailyActivity_merged | |
| hourlyCalories_merged | |
| hourlyIntensities_merged | |
| hourlySteps_merged | |
| minuteIntensitiesNarrow_merged | |
| minuteMETsNarrow_merged | |
| minuteSleep_merged | |
| sleepDay_merged | Only with April-May data |
| weightLogInfo_merged | |

There were issues when importing the time-based tables into BigQuery because the data was in a 12-hour format, which BigQuery did not recognize. To resolve

this, the format was converted to 24-hour time in Excel. This was done using functions such as splitting columns, CONCATENATE, and TEXT.

An attempt was made to find additional information from other datasets, particularly regarding water consumption and menstrual cycle tracking, which are not covered in this dataset. However, no free and sufficiently reliable datasets were found that collect this type of data from smart devices.

# 4 Process Phase

## *4.1 Prepare the data*

First, all the necessary files mentioned during the preparation phase were downloaded.

Then, I created the calories and steps tables for the April and May files. I used pivot tables in Excel due to the volume of data. I also used the "Text to Columns" tool and the "Concatenate" function to organize and combine the information more efficiently.

After gathering all the necessary files, each file was renamed. The new naming convention starts with the unit of measurement (day, hour, or minute), followed by the type of data recorded, and finally the date (based on the latest date found in the dataset).

The data processing and analysis stage will be carried out in BigQuery, aiming to use the three main tools of the course: Excel for inspecting and reviewing simple documents, SQL for the processing and analysis phase, and R for the visualization phase.

## *4.2 Data cleaning process*

First, I verified that the data in each file fell within the date range specified in the documentation. To do this, I sorted the rows both in ascending and descending order by the date column, and then performed a visual check. An example of the query used for this review is as follows:

```
SELECT *
FROM (
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_041116`
  UNION ALL
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_051216`
)
ORDER BY
  ActivityDay ASC
```

The maximum and minimum values were also identified, and then it was checked whether these values were realistic. Below is an example of the code used for this review:

```
SELECT
  MAX(Calories) AS max_calories,
  MIN(Calories) AS min_calories
FROM (
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_041116`
  UNION ALL
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_051216`
)
```

In addition, the presence of null values was checked, and below is an example of the code used for this verification:

```
SELECT *
FROM (
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_041116`
  UNION ALL
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyCalories_051216`
)
WHERE
  Calories IS NULL
```

It was verified that all IDs were valid by checking that they contained only numeric values and by counting the number of characters. The query used was as follows:

```
SELECT DISTINCT
  LENGTH(CAST(Id AS string)) AS id_lenght
FROM (
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyActivity_051216`
  UNION ALL
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyActivity_041116`
)
```

In some cases, the number of records with extremely low or high values was counted. The query used was as follows:

```
SELECT
  TotalSteps,
  COUNT(TotalSteps) AS records
FROM (
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyActivity_051216`
  UNION ALL
  SELECT * FROM `bellabeat-456023.fitbit_fitness_tracker.dailyActivity_041116`
)
GROUP BY
  TotalSteps
ORDER BY
  TotalSteps ASC
```

It was also verified that the sum of the different distance fields in the daily activity table matched the total distance. Since exact matches were not found, the percentage difference was used to assess consistency. The code used is shown below:

```
SELECT
  TotalDistance,
  TrackerDistance,
  LoggedActivitiesDistance + VeryActiveDistance +
```

```
  ModeratelyActiveDistance + LightActiveDistance +
  SedentaryActiveDistance AS calculateDistance,
  100*ABS(TotalDistance - (
    LoggedActivitiesDistance + VeryActiveDistance +
    ModeratelyActiveDistance + LightActiveDistance +
    SedentaryActiveDistance
  )) / TotalDistance AS PorcentualDifference
FROM (
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity_051216`
  UNION ALL
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity_041116`
)
WHERE
  TotalDistance != 0 AND
  100*ABS(TotalDistance - (
    LoggedActivitiesDistance + VeryActiveDistance +
    ModeratelyActiveDistance + LightActiveDistance +
    SedentaryActiveDistance
  )) / TotalDistance > 30
```

## *4.3 Data cleaning results*

In the daily activity table, several data-cleaning actions were performed to ensure consistency and analytical relevance. A total of 138 records with 0 steps were removed. Although a value of 0 is not inherently implausible—such as in cases of complete rest—the number of such records was excessive and could interfere with trend analysis.

Similarly, 141 records were removed where the total distance traveled was 0, as they provide limited insight and may suggest tracking issues. In addition, 64 records were discarded due to a percentage discrepancy greater than 30% between the total distance and the sum of the individual distance components.

Finally, 9 entries were excluded where no calories were recorded as burned throughout the day. The code used for these filtering operations is shown below:

```
SELECT *
FROM (
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity_051216`
  UNION ALL
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity_041116`
)
WHERE
  (TotalSteps != 0 OR
  TotalDistance != 0 OR
  Calories != 0) AND
  (TotalDistance != 0 AND
  100*ABS(TotalDistance - (
    LoggedActivitiesDistance + VeryActiveDistance +
    ModeratelyActiveDistance + LightActiveDistance +
    SedentaryActiveDistance
  )) / TotalDistance < 30)
```

Finally, the daily activity table contained 1,192 records after the data cleaning process.

In the hourly intensity table, 20,143 records were found with an intensity value of 0. This is a realistic value, as it is possible that no movement was recorded during certain hours. However, the number of records with a value of 0 represents a significant portion—nearly half of the total. In this case, it was decided to keep these data, as identifying periods of low activity will be important for detecting patterns and trends in physical behavior.

Records with METs values equal to 0 were removed, resulting in the exclusion of 13 entries. Consequently, the METs table now contains a total of 2,770,607 records.

Several issues were found in the weight table related to missing decimal points in the values for weight in kilograms, weight in pounds, and BMI. The first step in correcting these problems was to adjust the WeightPounds column since all the values appeared to lack a decimal point. This was done using the CONCAT and SUBSTR functions to insert the decimal point after the third digit from the left

Once the weight in pounds was corrected, it was used to recalculate the weight in kilograms using the standard conversion factor. These newly calculated values were then compared with the original WeightKg values to verify their consistency

For the BMI column, a similar approach was applied. However, a CASE statement was added to ensure that only values with obvious errors—specifically those greater than 100—were adjusted by inserting the decimal point

Lastly, the Fat column was not included in the analysis because it contained only four non-null entries

The SQL code used for all corrections and validations is presented below

```
SELECT
  Id,
  Date,
  ROUND(CAST(
    CONCAT(
      SUBSTR(CAST(WeightPounds AS STRING), 1, 3),
      '.',
      SUBSTR(CAST(WeightPounds AS STRING), 4)
    ) AS FLOAT64
  )*0.45359237, 2) AS WeightKg,
  CAST(
    CONCAT(
      SUBSTR(CAST(WeightPounds AS STRING), 1, 3),
      '.',
      SUBSTR(CAST(WeightPounds AS STRING), 4)
    ) AS FLOAT64
```

```
  ) AS WeightPounds,
  CASE
    WHEN BMI > 100 THEN CAST(
      CONCAT(
        SUBSTR(CAST(BMI AS STRING), 1, 2),
        '.',
        SUBSTR(CAST(BMI AS STRING), 3)
      ) AS FLOAT64
    )
    ELSE BMI
  END AS BMI,
  IsManualReport,
  LogId
FROM (
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.weightLogInfo_041116`
  UNION ALL
  SELECT * FROM `bellabeat-
456023.fitbit_fitness_tracker.weightLogInfo_051216`)
ORDER BY Id, Date
```

## 5 Analyze

I started by working with the daily data tables, beginning with the daily activity table. I selected the calories and steps columns, grouped the data by day, and then calculated statistics such as the average, maximum value, minimum value, and total sum. To do this, I used the following query:

```
SELECT
  FORMAT_TIMESTAMP("%A", Activity.ActivityDate)
AS day,
  SUM(TotalSteps) AS sum_steps,
  MIN(TotalSteps) AS min_steps,
  MAX(TotalSteps) AS max_steps,
  AVG(TotalSteps) AS avg_steps,
  SUM(Calories) AS sum_calories,
  MIN(Calories) AS min_calories,
  MAX(Calories) AS max_calories,
  AVG(Calories) AS avg_calories
FROM
  `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity` AS
Activity
GROUP BY
  FORMAT_TIMESTAMP("%A", Activity.ActivityDate)
ORDER BY
  sum_calories DESC
```

The result of this query produced the following table. The main finding is that Saturdays are the most active days, both in terms of steps taken and calories burned.

| day | Saturday | Tuesday | Friday | Wednesday | Sunday | Monday | Thursday |
|---|---|---|---|---|---|---|---|
| sum_steps | 1520997 | 1449779 | 1346789 | 1397353 | 1254616 | 1310275 | 1309953 |
| min_steps | 14 | 8 | 42 | 8 | 16 | 8 | 7 |
| max_steps | 29326 | 23186 | 23014 | 24136 | 36019 | 20779 | 21129 |
| avg_steps | 8.843 | 7.966 | 7.696 | 8.077 | 7.468 | 8.241 | 8.037 |
| sum_calories | 415357 | 406847 | 406258 | 406212 | 383773 | 369234 | 367727 |
| min_calories | 787 | 50 | 403 | 52 | 489 | 1223 | 257 |
| max_calories | 4547 | 4286 | 4196 | 4079 | 4552 | 4234 | 4236 |
| avg_calories | 2.415 | 2.235 | 23.215 | 2.348 | 2.284 | 2.322 | 2.256 |

Next, I decided to perform the same analysis, but this time distinguishing between weekdays and weekends. The query used for this is shown below.

```
SELECT
  CASE
    WHEN FORMAT_TIMESTAMP("%A",
Activity.ActivityDate) = "Saturday"
      OR FORMAT_TIMESTAMP("%A",
Activity.ActivityDate) = "Sunday" THEN "Weekend"
    ELSE "Weekday"
  END AS dayType,
  SUM(TotalSteps) AS sum_steps,
  MIN(TotalSteps) AS min_steps,
  MAX(TotalSteps) AS max_steps,
  AVG(TotalSteps) AS avg_steps,
  SUM(Calories) AS sum_calories,
  MIN(Calories) AS min_calories,
  MAX(Calories) AS max_calories,
  AVG(Calories) AS avg_calories
FROM
  `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity` AS
Activity
GROUP BY
  CASE
    WHEN FORMAT_TIMESTAMP("%A",
Activity.ActivityDate) = "Saturday"
      OR FORMAT_TIMESTAMP("%A",
Activity.ActivityDate) = "Sunday" THEN "Weekend"
    ELSE "Weekday"
  END
ORDER BY
  avg_calories DESC
```

The results of the query are presented in the following table, where it can be seen that, on average, more steps are taken and more calories are burned on weekends compared to weekdays.

| dayType | Weekend | Weekday |
|---|---|---|
| sum_steps | 2775613 | 6814149 |
| min_steps | 14 | 7 |
| max_steps | 36019 | 24136 |
| avg_steps | 8.164 | 7.998 |
| sum_calories | 799130 | 1956278 |
| min_calories | 489 | 50 |
| max_calories | 4552 | 4286 |
| avg_calories | 2.350 | 2.296 |

Finally, for the daily activity table, the times and distances for the different activity levels were aggregated. The goal was to analyze the users' level of sedentary behavior. The query used is shown below.

```
SELECT
  SUM(TotalDistance) AS totalDistance,
  SUM(TrackerDistance) AS trackerDistance,
  SUM(LoggedActivitiesDistance) AS
loggedActivitiesDistance,
  SUM(VeryActiveDistance) AS veryActiveDistance,
  SUM(ModeratelyActiveDistance) AS
moderatelyActiveDistance,
  SUM(LightActiveDistance) AS lightActiveDistance,
```

```
  SUM(SedentaryActiveDistance) AS
sedentaryActiveDistance,
  SUM(VeryActiveMinutes) AS veryActiveMinutes,
  SUM(FairlyActiveMinutes) AS fairlyActiveMinutes,
  SUM(LightlyActiveMinutes) AS lightlyActiveMinutes,
  SUM(SedentaryMinutes) AS sedentaryMinutes

FROM
  `bellabeat-
456023.fitbit_fitness_tracker.dailyActivity` AS
Activity
```

The results of the query are shown below, highlighting that the greatest distance is covered during light activity and that users spend most of their time in a sedentary state.

| | |
|---|---|
| totalDistance | 6.855,63 |
| trackerDistance | 6.854,31 |
| loggedActivitiesDistance | 37,56 |
| veryActiveDistance | 1.835,34 |
| moderatelyActiveDistance | 721,83 |
| lightActiveDistance | 4.286,61 |
| sedentaryActiveDistance | 2,16 |
| veryActiveMinutes | 25.442,00 |
| fairlyActiveMinutes | 17.372,00 |
| lightlyActiveMinutes | 248.471,00 |
| sedentaryMinutes | 1.124.191,00 |

For the sleep table, the data was grouped by each sleep record in order to analyze users' sleep quality and the amount of time they spend either sleeping or lying in bed. The query used is shown below:

```
SELECT
  TotalSleepRecords,
  COUNT(TotalSleepRecords) AS count_SleepRecords,
  SUM(TotalMinutesAsleep) AS minutesAsleepByRecord,
  SUM(TotalTimeInBed) AS timeInBedByRecord,
  AVG(TotalMinutesAsleep) AS avgSleepByRecord,
  AVG(TotalTimeInBed) AS avgInBedByRecord
FROM
  `bellabeat-456023.fitbit_fitness_tracker.sleepDay`
GROUP BY
  TotalSleepRecords
ORDER BY
  avgSleepByRecord DESC
```

The results are shown in the following table, where it is notable that very few users have three sleep sessions in a day, while the majority have only one sleep session. Users with three sleep sessions sleep more than 8 hours, those with two records sleep around 8 hours, and those with only one sleep session sleep less than 8 hours.

| | | | |
|---|---|---|---|
| TotalSleepRecords | 3 | 2 | 1 |
| count_SleepRecords | 3 | 43 | 367 |
| minutesAsleepByRecord | 1932,00 | 19485,00 | 151823,00 |
| timeInBedByRecord | 2049,00 | 21504,00 | 165865,00 |
| avgSleepByRecord | 644,00 | 453,14 | 413,69 |
| avgInBedByRecord | 683,00 | 500,09 | 451,95 |

For the hourly data analysis, all tables were merged and analyzed together. The day was divided into four time intervals: morning, afternoon, evening, and night. The average of each metric was then calculated for each time period. The query used is shown below.

```
DECLARE
  MORNING_START,
  MORNING_END,
  AFTERNOON_END,
  EVENING_END INT64;
SET
  MORNING_START = 6;
SET
  MORNING_END = 12;
SET
  AFTERNOON_END = 18;
SET
  EVENING_END = 21;
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM calories.ActivityHour) >
MORNING_START AND
      EXTRACT(HOUR FROM calories.ActivityHour) <=
MORNING_END THEN "Morning"
    WHEN EXTRACT(HOUR FROM calories.ActivityHour) >
MORNING_END AND
      EXTRACT(HOUR FROM calories.ActivityHour) <=
AFTERNOON_END THEN "Afternoon"
    WHEN EXTRACT(HOUR FROM calories.ActivityHour) >
AFTERNOON_END AND
      EXTRACT(HOUR FROM calories.ActivityHour) <=
EVENING_END THEN "Evening"
    ELSE "Night"
  END AS time_of_day,
  AVG(calories.Calories) AS avg_calories,
  AVG(steps.StepTotal) AS avg_steps,
  AVG(intensities.TotalIntensity) AS avg_intensity
FROM
  `bellabeat-
456023.fitbit_fitness_tracker.hourlyCalories` AS
calories
  INNER JOIN
  `bellabeat-
456023.fitbit_fitness_tracker.hourlyIntensities` AS
intensities
  ON calories.Id = intensities.Id AND
calories.ActivityHour = intensities.ActivityHour
  INNER JOIN
  `bellabeat-456023.fitbit_fitness_tracker.hourlySteps`
AS steps
  ON calories.Id = steps.Id AND calories.ActivityHour =
steps.ActivityHour
GROUP BY
  time_of_day
```

The results of the query are shown below. It is evident that the afternoon is the most active time of day, while the night is the least active in terms of steps, calories burned, and intensity.

| time_of_day | avg_calories | avg_steps | avg_intensity |
|---|---|---|---|
| Night | 74,66 | 63,98 | 3,23 |
| Morning | 105,37 | 432,57 | 15,39 |
| Evening | 105,46 | 405,41 | 15,36 |
| Afternoon | 113,26 | 486,83 | 17,91 |

The minute-level sleep table was used to analyze when during the day users are asleep. To do this, the data was grouped by each hour of the day, and using the "value" column, it was determined whether the user was awake, restless, or asleep. Then, the number of minutes spent in each of these states was counted for every hour.

```
SELECT
  EXTRACT(HOUR FROM date) AS hour,
  COUNT(CASE WHEN value = 1 THEN 1 END) AS
minutes_asleep,
  COUNT(CASE WHEN value = 2 THEN 1 END) AS
minutes_restless,
  COUNT(CASE WHEN value = 3 THEN 1 END) AS
minutes_awake
FROM
  `bellabeat-456023.fitbit_fitness_tracker.minuteSleep`
GROUP BY
  hour
ORDER BY
  hour
```

The results of the query are shown below, where it is observed that users are generally more awake around midday. However, the data seems to contain some inconsistencies: at all hours there are more records of people sleeping than awake. This could be due to an issue in data collection or a malfunction of the Fitbit device. It may also stem from the interpretation of the "value" column, which was obtained from external sources rather than the official dataset documentation.

| hour | minutes_asleep | minutes_restless | minutes_awake |
|---|---|---|---|
| 0 | 31990 | 1830 | 357 |
| 1 | 36017 | 2255 | 242 |
| 2 | 39923 | 2458 | 366 |
| 3 | 43214 | 2382 | 224 |
| 4 | 41694 | 2567 | 256 |
| 5 | 38435 | 2501 | 173 |
| 6 | 28587 | 2703 | 255 |
| 7 | 18542 | 2025 | 212 |
| 8 | 11208 | 1391 | 118 |
| 9 | 7190 | 564 | 68 |
| 10 | 4212 | 354 | 90 |
| 11 | 1906 | 211 | 79 |
| 12 | 938 | 126 | 41 |
| 13 | 1035 | 112 | 33 |
| 14 | 994 | 111 | 9 |
| 15 | 1137 | 145 | 47 |
| 16 | 1067 | 141 | 29 |
| 17 | 728 | 94 | 18 |
| 18 | 508 | 71 | 75 |
| 19 | 677 | 81 | 63 |
| 20 | 1485 | 354 | 166 |
| 21 | 5721 | 842 | 336 |
| 22 | 13998 | 1707 | 439 |
| 23 | 24800 | 2008 | 345 |

Finally, the weight tables were analyzed, using measurements in kilograms, as they are the most common in my region. First, basic statistics related to BMI (Body Mass Index) and users' weight were obtained. The query used is shown below.

```
SELECT
  MAX(WeightKg) AS max_weight,
  MIN(WeightKg) AS min_weight,
  AVG(WeightKg) AS avg_weight,
  MAX(BMI) AS max_BMI,
  MIN(BMI) AS min_BMI,
  AVG(BMI) AS avg_BMI
FROM
  `bellabeat-
456023.fitbit_fitness_tracker.weightLogInfo`
```

The results of the query are shown below, where users appear to belong to different weight categories based on their BMI.

| | |
|---|---|
| max_weight | 133,50 |
| min_weight | 52,60 |
| avg_weight | 72,50 |
| max_BMI | 47,54 |
| min_BMI | 21,45 |
| avg_BMI | 25,37 |

Finally, based on the BMI (Body Mass Index) data, users were grouped into four categories: Underweight (less than 18.5), Normal weight (18.5 – 24.9), Overweight (25.0 – 29.9), and Obese (30.0 or higher). This classification was used to analyze the types of users present in the Fitbit dataset. The query used is shown below.

```
SELECT
```

```
  COUNT(CASE WHEN BMI < 18.5 THEN 1 END) AS
Underweight,
  COUNT(CASE WHEN BMI >= 18.5 AND BMI < 25.0 THEN 1
END) AS NormalWeight,
  COUNT(CASE WHEN BMI >= 25.0 AND BMI < 30.0 THEN 1
END) AS Overweight,
  COUNT(CASE WHEN BMI >= 30.0 THEN 1 END) AS Obese
FROM
  `bellabeat-
456023.fitbit_fitness_tracker.weightLogInfo`
```

The results are shown in the following table, with a notable number of users falling into the overweight category.

| | |
|---|---|
| Underweight | 0 |
| NormalWeight | 52 |
| Overweight | 45 |
| Obese | 3 |

## 6 Share

For the visualization section, I started by working with the daily tables data. In general, the charts I aimed to create could be developed using Excel along with the summary tables generated from SQL. However, there was an additional metric I wanted to explore — the potential relationship between sleep data and intensity data. To analyze this, I used R.

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

# Load the data
calories <- read_csv("dailyCalories.csv")
Sleep <- read_csv("sleepDay.csv")

# Summarize the calories data
calories_summary <- calories %>%
 group_by(Id) %>%
 summarize(
  total_calories = sum(Calories, na.rm = TRUE)
 )

# Summarize the sleep data
sleep_summary <- Sleep %>%
 group_by(Id) %>%
 summarize(
  total_sleep = sum(TotalMinutesAsleep, na.rm = TRUE)  # Sum
TotalMinutesAsleep
 )

# Combine both datasets based on 'Id'
combined_data <- calories_summary %>%
 inner_join(sleep_summary, by = "Id")

# Create the plot
ggplot(combined_data, aes(x = total_sleep, y = total_calories)) +
 geom_point(color = "blue") +
 labs(
  title = "Relationship between Calories Burned and Sleep Time",
  x = "Sleep Time (in minutes)",
  y = "Calories Burned"
 ) +
 theme_minimal()
```

However, no relationship was found between activity and sleep duration, meaning that no trend can be leveraged for a marketing campaign.

## 7 Act

The following are the final conclusions based on the analysis of user behavior. These insights will help guide advertising and targeting strategies more effectively.

1. Overall, users engage mostly in light physical activity, so it would be advisable to use images showing people involved in such activities in advertising campaigns.
2. Users tend to be slightly more active on weekends, especially on Saturdays, making those days ideal for launching advertisements.
3. User activity peaks in the afternoon, suggesting that advertising efforts should be concentrated during that time slot. Conversely, activity levels drop at night, so it is best to avoid advertising during late hours.
4. Most users take only one nap a day and sleep less than 8 hours. There is no clear correlation between physical activity and sleep duration, so no specific trend can be leveraged in that regard. However, users tend to be more alert in the morning hours.
5. The majority of users fall within the normal or overweight categories, making this the primary target group for advertising efforts.