Google Data Analytics Capstone Project

Case Study 2:

**How Can a Wellness Technology Company Play It Smart?**

**Business: “Bellabeat”**

# Summary:

# Business Task:

* How do consumers use non-Bellabeat smart devices?
* What are the patterns or trends of usage?
* How can this be applied to enhance Bellabeat’s marketing strategy and attract more consumers?
* How can Bellabeat attract new customers by analyzing the smart devices’ usage patterns?

# Stakeholders:

Primary Stakeholders:

* Bellabeat Marketing Analytics Team
* Ms. Urška Sršen (Cofounder & Chief Creative Officer of Bellabeat)

Secondary Stakeholders:

* Mr. Sando Mur (Mathematician – Cofounder & CEO at Bellabeat)

# Exploring Data:

The datasets contain information about the number of daily steps, the sleeping times, the calories burned, and the activity rate. We’ll analyze the patterns of usage through time, consistency through date streaks & average daily user activity to gain insights about their preferences that will help Bellabeat approach better to potential customers. We’ll deal with another dataset as the one under our hands has some limitations that will be mentioned in the upcoming section.

# Preparing Data:

The first dataset was obtained from Kaggle, discussing people log activities for other non-Bellabeat smart health devices. However, it’s not completely ROCCC. It’s collected in the time span of one month, from almost mid-April to mid-May 2016, and it’s worth mentioning that we are now in 2023, we have better health technologies. In addition, people’s lifestyle has significantly changed due to the Covid-19 pandemic. We’ll consider this dataset as well as another, more “ROCCC” dataset.

Edit: We’ll use our first dataset supplement to be: [[Fitness tracker data (2016 - present) [2450+days]](https://www.kaggle.com/datasets/damirgadylyaev/more-than-4-years-of-steps-and-sleep-data-mi-band)] collected by Damir Gadylyaev on Kaggle. This dataset lists the daily steps & sleep tracking for mi band from 2016 till January 2023 for one user (2450+ days), living in Europe, which is more “ROCCC”. The dataset is CC0 licensed (Public Domain use).

Edit 2: we’ll refer to old dataset as D1, and the new dataset as D1\_S

Since we are dealing with relatively large datasets. I’ll use Excel for cleaning & exploring, R for analyzing data, visualization & documentation.

D1 contains 18 files, including hourly & minutely recording of data. However, we believe dealing with daily records will help us obtain actionable overall insights & better analysis. While D1\_S contains only 2 files, both are recorded daily.

D1 files are long from the first glance, since ID is a primary key representing the user, and it’s repeating by day. Similarly, D1\_S is collected for one user. Thus, it’s definitely long dataset.

After having an overall look on both datasets, we’ll consider only 4 files in D1 out of total 18 files, files are “dailyActivity\_merged.csv”, “hourySteps\_merged.csv”, “weightLogInfo\_merged.csv” & “sleepDay\_merged.csv” and both files from D1\_S: “01\_Steps.csv” & “01\_Sleep.csv”.

# Processing Data:

The first challenge to understand our dataset is the units of D1, especially distance is it in km or miles? No clue is given about this. After a long search and asking the Kaggle community, it’s in km.

We will first clean our data using Excel, firstly by making sure every cell is in its correct format. In both datasets, dates are date, distances are in numbers (rounded to 3 decimal places), while BMI (Body Mass Index) is rounded to only 2 places. Finally, minutes, steps & calories are integer numbers. We changed all fields names in all working files to match camelCase convention for more consistency, we also change the type of “id” field into text in all tables as it won’t be involved in any calculations or analyses.

In “dailyActivity\_merged.csv” file of D1, we observed some dates were in the format of mm/dd/yyyy (right aligned) & some others were in the format of dd/mm/yyyy (left aligned). This was observed by trying to change the data type from general to date for the whole field. However, they look the same from the first glance (prefer to figure 1). We cleaned the dates and unified the format to dd/mm/yyyy. We observed the field “loggedActivitiesDistance” is almost full of zeros, after filtering, it seems not. We’ll keep that in mind for analysis. We also found the “totalDistance” & “trackerDistance” fields are almost equal, we set a filter to check that and understand what it indicates. Finally, we saved a copy of the table in a new name “activity.csv” for ease.

In “hourlySteps\_merged,csv” file, we split the date time into a date and a time fields and in the same time kept the original field for reference, we renamed the original field “activityDateHour” & the splitted fields “activityDate” & “activityHour” respectively. We also assign the data types. Fortunately, all the data was clean in this file. We saved a copy from the original file and renamed it “activityH.csv” for ease.

Similarly, in “sleepDay\_merged.csv”, we split date and time into 2 fields, everything seems clean and acceptable. We saved a copy of the table then renamed it as “sleep.csv”.

In the “weightLogInfo\_merged.csv” file, we split date and time again. We also have a column named “Fat” and it has lots of null values (66 null values out of 68 total values), we kept it, but we will not rely on it in our analysis. We found a repeated record for the id “1503960366” in the file. Finally, we renamed our copied table as “weight.csv”.

In “01\_Steps.csv” file in D1\_S, distance is measured in a different way, for example: in D1, 13162 steps are equivalent to 8.5 km. While in D1\_S, 13164 steps are equivalent to 10246, so most probably it’s in meters or yards. Since D1\_S owner lives in Europe, where they follow SI Unit System. The chances are higher it’s in meters. Thus, we’ll assume it in meters and unify it to kilometers in both datasets. We added two new columns “distanceKm” & “runDistanceKm”. We renamed our file “uSteps.csv”.

The last file we have is “02\_Sleep.csv” in D1\_S, we observed the difference between start and stop times isn’t equal to the total number of sleeping minutes recorded. Which will be kept in consideration in analysis. We renamed this file to “uSleep.csv”. Also, we observed the start and end of sleeping dates’ formatting changed fluctuatingly. Each six months approximately the format was changing from a uniform date to a Unix timestamp[[1]](#footnote-1) alternately. In order to convert it we have to divide the Unix timestamp and convert it into days, months and years and then add it to the start point (which is January 1, 1970). In addition, there are too many null Unix time stamp values. So, we cleaned the dates and converted them into the original uniform format using the equation:

After that, we stored the new dates in separate columns.  
Another observed issue in the same fields is even the formatted dates have an appended “+0000” in the end of the date. Which is actually preventing Excel from recognizing the fields as a date-time type. We trimmed this part using the Excel “RIGHT” function.  
Again, we observed many records with null date values, so we deleted them Finally, we renamed our work file copy “uSleep.csv”.

Finally, we’ll name our project “Bellabeat\_R”. We were considering doing the analysis on SQL BigQuery, but after creating the queries, it seems Google Cloud is laggy. Thus, we’ll analyze data on RStudio instead.

# Data Analysis

We first prepared our setup in Rstudio, this was done by installing and loading the packages “dyplyr”, “readr”, “ggplot2”, “ggsci”, and only loading “lubridate” package.

# Data Analysis:

We will run 4 queries to understand the data of D1. As mentioned earlier, we’ll not be concerned about sleep in this analysis. Our first query is to create a table for those who record their weight as well as workout progress.

SELECT \*

FROM patterns\_fitbit.activity

INNER JOIN patterns\_fitbit.weight

  ON activity.id = weight.id

We’ll count them using the COUNT (DISTINCT) function:

SELECT COUNT(DISTINCT activity.id)

FROM patterns\_fitbit.activity

INNER JOIN patterns\_fitbit.weight

  ON activity.id = weight.id

  /\* 8 people only record their weight and activity \*/

We’ll filter them further by counting those recorded their weight and activity on the same day:

SELECT COUNT(DISTINCT activity.id)

FROM patterns\_fitbit.activity

INNER JOIN patterns\_fitbit.weight

  ON activity.id = weight.id AND activity.activityDate = weight.date

  /\* 7 people only record their weight and activity on the same day \*/

Then, we’ll count how many people did any of both, total people who recorded their activity are 33 people, and total people who recorded their weight are 8 people. We can summarize our findings into the following venn:

1. Unix timestamp is the total number of seconds starting from January 1, 1970, until a particular moment. [↑](#footnote-ref-1)