**Bellabeat**

**Case Study: How Can a Wellness Technology Company Play It Smart?**

***Bellabeat product: Urban Leaf***

**Logo

Description automatically generated**

**ASK:**

* What are some trends in smart device usage?
* How could these trends apply to Bellabeat customers?
* How could these trends help influence Bellabeat marketing strategy?

**Business task:**  Analyze smart device usage data in order to gain insights into how consumers use non-Bellabeat smart devices. Then take one Bellabeat product and apply it to these insights.

**Prepare**

* Data was stored on the computer.
* Most of the data is in a long format. Some files are in a wide format, but it makes it difficult to read and it provides the same information as data in a long format.
* The data I am working with, is broken down in 18 different files. Some files contain the same columns you can find in other files.
* There is a lot of Data replication with columns which contain the same information. I will delete each replication to ensure data integrity.
* Then I tried to remove all duplicates. But didn’t find any.

Select all data -> Data -> Remove duplicates.

* Data was downloaded on [FitBit Fitness Tracker Data | Kaggle](https://www.kaggle.com/arashnic/fitbit). The data was made available through Mobius.

**Process**

* The tool I will be using is the R programming language.
* To ensure I will be working with clean data I removed all the duplicates. I also removed all the columns and rows which contained the same information through multiple files.
* I use an appropriate file naming convention. The first file for instance I named “Daily\_activity”. The second name is Heartrate\_seconds. I keep the file names similar, to ensure clarity.
* Now I import all the data into R. First, I install the tidyverse package and put it into the library. Second, I do the same with ggplot2.

install.packages(“ggplot2”)

library(ggplot2)

install.packages(“tidyverse”)

library(tidyverse)

* Now I import all the files I’ll be working on into R.

Select ‘Upload’ -> Choose file -> select a zip file so I can upload all files at once -> ok ->

Select folder from project -> import dataset.

* To start my cleaning process, I’ll install and load the packages ‘here’, ‘skimr’, ‘janitor’ and ‘dplyr’.

Install.packages(“here”)

library(here)

install.packages(“janitor”)

library(janitor)

install.packages(“skimr”)

library(skimr)

install.packages(“dplyr”)

library(dplyr)

* Next, I am going to use three functions which will help me to see the data in more comprehensive and precise way.

skim\_without\_charts(Daily\_activity)

|  |
| --- |
| Data Summary ──────────────────────── |
| Values |
| Name Daily\_activity |
| Number of rows 940 |
| Number of columns 15 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Column type frequency: |
| character 1 |
| numeric 14 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Group variables None |
|  |
| ── Variable type: character ───────────────────────────────────────────────────────── |
| skim\_variable n\_missing complete\_rate min max empty n\_unique whitespace |
| 1 ActivityDate 0 1 8 9 0 31 0 |
|  |
| ── Variable type: numeric ─────────────────────────────────────────────────────────── |
| skim\_variable n\_missing complete\_rate mean sd p0 |
| 1 Id 0 1 4.86e+9 2.42e+9 1503960366 |
| 2 TotalSteps 0 1 7.64e+3 5.09e+3 0 |
| 3 TotalDistance 0 1 5.49e+0 3.92e+0 0 |
| 4 TrackerDistance 0 1 5.48e+0 3.91e+0 0 |
| 5 LoggedActivitiesDistance 0 1 1.08e-1 6.20e-1 0 |
| 6 VeryActiveDistance 0 1 1.50e+0 2.66e+0 0 |
| 7 ModeratelyActiveDistance 0 1 5.68e-1 8.84e-1 0 |
| 8 LightActiveDistance 0 1 3.34e+0 2.04e+0 0 |
| 9 SedentaryActiveDistance 0 1 1.61e-3 7.35e-3 0 |
| 10 VeryActiveMinutes 0 1 2.12e+1 3.28e+1 0 |
| 11 FairlyActiveMinutes 0 1 1.36e+1 2.00e+1 0 |
| 12 LightlyActiveMinutes 0 1 1.93e+2 1.09e+2 0 |
| 13 SedentaryMinutes 0 1 9.91e+2 3.01e+2 0 |
| 14 Calories 0 1 2.30e+3 7.18e+2 0 |
| p25 p50 p75 p100 |
| 1 2320127002 4.45e+9 6.96e+9 8.88e+9 |
| 2 3790. 7.41e+3 1.07e+4 3.60e+4 |
| 3 2.62 5.24e+0 7.71e+0 2.80e+1 |
| 4 2.62 5.24e+0 7.71e+0 2.80e+1 |
| 5 0 0 0 4.94e+0 |
| 6 0 2.10e-1 2.05e+0 2.19e+1 |
| 7 0 2.40e-1 8.00e-1 6.48e+0 |
| 8 1.95 3.36e+0 4.78e+0 1.07e+1 |
| 9 0 0 0 1.10e-1 |
| 10 0 4 e+0 3.2 e+1 2.1 e+2 |
| 11 0 6 e+0 1.9 e+1 1.43e+2 |
| 12 127 1.99e+2 2.64e+2 5.18e+2 |
| 13 730. 1.06e+3 1.23e+3 1.44e+3 |
| 14 1828. 2.13e+3 2.79e+3 4.9 e+3 |

head(Daily\_activity)

|  |
| --- |
| # A tibble: 6 × 15 |
| Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitiesD… |
| *<dbl>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* |
| 1 1503960366 4/12/2016 13162 8.5 8.5 0 |
| 2 1503960366 4/13/2016 10735 6.97 6.97 0 |
| 3 1503960366 4/14/2016 10460 6.74 6.74 0 |
| 4 1503960366 4/15/2016 9762 6.28 6.28 0 |
| 5 1503960366 4/16/2016 12669 8.16 8.16 0 |
| 6 1503960366 4/17/2016 9705 6.48 6.48 0 |

glimpse(Daily\_activity)

|  |
| --- |
| Rows: 940 |
| Columns: 15 |
| $ Id *<dbl>* 1503960366, 1503960366, 1503960366, 1503960366, 15… |
| $ ActivityDate *<chr>* "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016"… |
| $ TotalSteps *<dbl>* 13162, 10735, 10460, 9762, 12669, 9705, 13019, 155… |
| $ TotalDistance *<dbl>* 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.… |
| $ TrackerDistance *<dbl>* 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.88, 6.… |
| $ LoggedActivitiesDistance *<dbl>* 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,… |
| $ VeryActiveDistance *<dbl>* 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.53, 1.… |
| $ ModeratelyActiveDistance *<dbl>* 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.32, 0.… |
| $ LightActiveDistance *<dbl>* 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.03, 4.… |
| $ SedentaryActiveDistance *<dbl>* 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,… |
| $ VeryActiveMinutes *<dbl>* 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 41, 39… |
| $ FairlyActiveMinutes *<dbl>* 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21, 5, … |
| $ LightlyActiveMinutes *<dbl>* 328, 217, 181, 209, 221, 164, 233, 264, 205, 211, … |
| $ SedentaryMinutes *<dbl>* 728, 776, 1218, 726, 773, 539, 1149, 775, 818, 838… |
| $ Calories *<dbl>* 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 17… |

* For the next step, I will use a function to look at just one column instead of having to look at multiple columns at the same time

Daily\_activity %>%

select(TotalSteps)

|  |
| --- |
| # A tibble: 940 × 1 |
| TotalSteps |
| *<dbl>* |
| 1 13162 |
| 2 10735 |
| 3 10460 |
| 4 9762 |
| 5 12669 |
| 6 9705 |
| 7 13019 |
| 8 15506 |
| 9 10544 |
| 10 9819 |
| # … with 930 more rows |

* Now I want to rename the column ‘ActivityDate’, so it doesn’t get confused with ‘Daily\_activity’.

Daily\_activity %>%

rename(Date=ActivityDate)

|  |
| --- |
| # A tibble: 940 × 15 |
| Id Date TotalSteps TotalDistance TrackerDistance LoggedActivitiesDis… |
| *<dbl>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* |
| 1 1503960366 4/12/2016 13162 8.5 8.5 0 |
| 2 1503960366 4/13/2016 10735 6.97 6.97 0 |
| 3 1503960366 4/14/2016 10460 6.74 6.74 0 |
| 4 1503960366 4/15/2016 9762 6.28 6.28 0 |
| 5 1503960366 4/16/2016 12669 8.16 8.16 0 |
| 6 1503960366 4/17/2016 9705 6.48 6.48 0 |
| 7 1503960366 4/18/2016 13019 8.59 8.59 0 |
| 8 1503960366 4/19/2016 15506 9.88 9.88 0 |
| 9 1503960366 4/20/2016 10544 6.68 6.68 0 |
| 10 1503960366 4/21/2016 9819 6.34 6.34 0 |

* Sometimes I like to work with uppercase column names, so it is easier for me to differentiate between other values. To do this, I am going to use a specific function.

rename\_with(Daily\_activity,toupper)

|  |
| --- |
| ID ACTIVITYDATE TOTALSTEPS TOTALDISTANCE TRACKERDISTANCE LOGGEDACTIVITIES… |
| *<dbl>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* |
| 1 1503960366 4/12/2016 13162 8.5 8.5 0 |
| 2 1503960366 4/13/2016 10735 6.97 6.97 0 |
| 3 1503960366 4/14/2016 10460 6.74 6.74 0 |
| 4 1503960366 4/15/2016 9762 6.28 6.28 0 |
| 5 1503960366 4/16/2016 12669 8.16 8.16 0 |
| 6 1503960366 4/17/2016 9705 6.48 6.48 0 |
| 7 1503960366 4/18/2016 13019 8.59 8.59 0 |
| 8 1503960366 4/19/2016 15506 9.88 9.88 0 |
| 9 1503960366 4/20/2016 10544 6.68 6.68 0 |
| 10 1503960366 4/21/2016 9819 6.34 6.34 |

* The next function I am going to use will make sure that all the names will contain only characters, numbers, and underscores.

clean\_names(Daily\_activity)

|  |
| --- |
| id activity\_date total\_steps total\_distance tracker\_distance logged\_activiti… |
| *<dbl>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* |
| 1 1503960366 4/12/2016 13162 8.5 8.5 0 |
| 2 1503960366 4/13/2016 10735 6.97 6.97 0 |
| 3 1503960366 4/14/2016 10460 6.74 6.74 0 |
| 4 1503960366 4/15/2016 9762 6.28 6.28 0 |
| 5 1503960366 4/16/2016 12669 8.16 8.16 0 |
| 6 1503960366 4/17/2016 9705 6.48 6.48 0 |
| 7 1503960366 4/18/2016 13019 8.59 8.59 0 |
| 8 1503960366 4/19/2016 15506 9.88 9.88 0 |
| 9 1503960366 4/20/2016 10544 6.68 6.68 0 |
| 10 1503960366 4/21/2016 9819 6.34 6.34 0 |
|  |

* I need to use the separate() function to separate the date and the time, which are included in the same column. But before I do that, I will create a data frame and use only the first six values in the column to see how it looks like as a result. At the end I will print my new data frame. The data frame will be named as, ‘Activity\_Date\_Time’.

ActivityHour <- c(“4/12/2016 12:00:00 AM”, “4/12/2016 1:00:00 AM”, 4/12/2016 2:00:00 AM”, “4/12/2016 3:00:00 AM”, 4/12/2016 4:00:00 AM”, “4/12/2016 5:00:00”)

Calories <- c(“81”, “61”, “59”, “47”, “48”, “48”)

Activity\_Date\_Time <- data.frame(ActivityHour, Calories)

print(Activity\_Date\_Time)

|  |
| --- |
| ActivityHour Calories |
| 1 4/12/2016 12:00:00 AM 81 |
| 2 4/12/2016 1:00:00 AM 61 |
| 3 4/12/2016 2:00:00 AM 59 |
| 4 4/12/2016 3:00:00 AM 47 |
| 5 4/12/2016 4:00:00 AM 48 |
| 6 4/12/2016 5:00:00 AM 48 |

* And now I am going to use the separate() function as mentioned above to have the date and time separated.

Separate(Activity\_Date\_Time, ActivityHour,into=c(‘Activity\_Date’,’Activity\_Time’), sep=’ ‘)

|  |
| --- |
| Activity\_Date Activity\_Time Calories |
| 1 4/12/2016 12:00:00 81 |
| 2 4/12/2016 1:00:00 61 |
| 3 4/12/2016 2:00:00 59 |
| 4 4/12/2016 3:00:00 47 |
| 5 4/12/2016 4:00:00 48 |
| 6 4/12/2016 5:00:00 48 |

* I would use the separate() function whenever needed or for my own convenience to keep the data separated while working on it. To reverse it and have the date and time combined again, I will use the unite() function. For this I am going to use the same data. I will unite the two new generated columns with the column ActivityHour again.

unite(Activity\_Date\_Time, ‘ActivityHour’,Activity\_Date,Activity\_Time,sep=’ ‘)

|  |
| --- |
| ActivityHour Calories |
| 1 4/12/2016 12:00:00 AM 81 |
| 2 4/12/2016 1:00:00 AM 61 |
| 3 4/12/2016 2:00:00 AM 59 |
| 4 4/12/2016 3:00:00 AM 47 |
| 5 4/12/2016 4:00:00 AM 48 |
| 6 4/12/2016 5:00:00 AM 48 |

* Doing the next step, I went back to the data where the ‘VeryActiveMinutes’ were tracked down. I decide to use hours instead of minutes. For that I will have to use the mutate() function. At the end of the function, I divide the hours with minutes by 60.

Daily\_activity %>%

mutate(VeryActiveHours=VeryActiveMinutes/60)

|  |
| --- |
| TrackerDistance LoggedActivitiesDistance VeryActiveHours |
| *<dbl>* *<dbl>* *<dbl>* |
| 8.5 0 0.417 |
| 6.97 0 0.35 |
| 6.74 0 0.5 |
| 6.28 0 0.483 |
| 8.16 0 0.6 |
| 6.48 0 0.633 |
| 8.59 0 0.7 |
| 9.88 0 0.833 |
| 6.68 0 0.467 |
| 6.34 0 0.317 |

**Analyze**

* To start the analyze phase, I start by looking at the two files I have: Daily\_activity and Sleep\_day. I use the head() function as I did before to get a better look of the data.
* Then I use the colnames() function to specifically look only at the columns.

colnames(Daily\_activity)

colnames(Sleep\_day)

|  |
| --- |
| [1] "Id" "ActivityDate" |
| [3] "TotalSteps" "TotalDistance" |
| [5] "TrackerDistance" "LoggedActivitiesDistance" |
| [7] "VeryActiveDistance" "ModeratelyActiveDistance" |
| [9] "LightActiveDistance" "SedentaryActiveDistance" |
| [11] "VeryActiveMinutes" "FairlyActiveMinutes" |
| [13] "LightlyActiveMinutes" "SedentaryMinutes" |
| [15] "Calories" |

|  |
| --- |
| [1] "Id" "SleepDay" "TotalSleepRecords" |
| [4] "TotalMinutesAsleep" "TotalTimeInBed" |
|  |

* My next step is to pull out all the unique values from these two data frames. To do this, I will use the n\_distinct() function.

n\_distinct(Daily\_activity$Id)

n\_distinct(Sleep\_day$Id)

|  |
| --- |
| [1] 33 |

|  |
| --- |
| [1] 24 |

* I also want to check how many rows the two data frames have. Therefore, I must use the nrow() function.

nrow(Daily\_activity)

nrow(Sleep\_day)

|  |
| --- |
| [1] 940 |

|  |
| --- |
| [1] 413 |

* To better absorb and understand the data, I want to use the summary() function from Daily\_activity.

Daily\_activity %>%

select(TotalSteps,TotalDistance,SedentaryMinutes) %>%

summary()

|  |
| --- |
| TotalSteps TotalDistance SedentaryMinutes |
| Min. : 0 Min. : 0.000 Min. : 0.0 |
| 1st Qu.: 3790 1st Qu.: 2.620 1st Qu.: 729.8 |
| Median : 7406 Median : 5.245 Median :1057.5 |
| Mean : 7638 Mean : 5.490 Mean : 991.2 |
| 3rd Qu.:10727 3rd Qu.: 7.713 3rd Qu.:1229.5 |
| Max. :36019 Max. :28.030 Max. :1440.0 |

* The same I will do with the Sleep\_day data.

Sleep\_day %>%

Select(TotalSleepRecords,TotalMinutesAsleep,TotalTimeInBed) %>%

summary()

|  |
| --- |
| TotalSleepRecords TotalMinutesAsleep TotalTimeInBed |
| Min. :1.000 Min. : 58.0 Min. : 61.0 |
| 1st Qu.:1.000 1st Qu.:361.0 1st Qu.:403.0 |
| Median :1.000 Median :433.0 Median :463.0 |
| Mean :1.119 Mean :419.5 Mean :458.6 |
| 3rd Qu.:1.000 3rd Qu.:490.0 3rd Qu.:526.0 |
| Max. :3.000 Max. :796.0 Max. :961.0 |

* Looking at the first sample from Daily\_activity I can see that people’s activity is much higher in the third quarter than in the first. The amount of TotalSteps and TotalDistance is significantly higher. But nevertheless, I notice that people tend to sit much more in the third quarter than in the first. Looking at the second data I can see that people sleep longer and stay longer in bed at the end of the year in contrast to the beginning of the year.
* Going forward, I want to look at the data which was recorded by day. For this I use a simple function to pull out two columns.

Daily\_activity %>%

select(TotalSteps,SedentaryMinutes)

|  |
| --- |
| TotalSteps SedentaryMinutes |
| *<dbl>* *<dbl>* |
| 1 13162 728 |
| 2 10735 776 |
| 3 10460 1218 |
| 4 9762 726 |
| 5 12669 773 |
| 6 9705 539 |
| 7 13019 1149 |
| 8 15506 775 |
| 9 10544 818 |
| 10 9819 838 |

* Having the Totals and the data per day will help me identify patterns later.
* Now let’s put this into a visualization to identify the relationship between TotalSteps taken and SedentaryMinutes.

ggplot(data=Daily\_activity,aes(x=TotalSteps,y=SedentaryMinutes))+

geom\_point()

Chart, scatter chart

Description automatically generated

* What I see at a first glance is that the less steps people take the more they use to sit. People who do not work out spent most of the time seated compared to people who work out.
* Up next I want to know if walking more can help in loosing weight.

ggplot(data=Daily\_activity,aes(x=TotalSteps,y=Calories))+

geom\_point()

Chart, scatter chart

Description automatically generated

* There is a correlation between Calories and TotalSteps. The graphic clearly shows that the more steps people make the more calories they lose.
* So, the conclusion is that when people use to walk or jog more people will lose more weight.
* My next step is to put TotalMinutesAsleep and TotalTimeInBed together and see if there are any big differences.

ggplot(data=Sleep\_day, aes(x=TotalMinutesAsleep,y=TotalTimeInBed))+

geom\_point()

Chart, scatter chart

Description automatically generated

* At first sight it seems completely linear. But there are some unexpected trends. Some dots in the top right corner show different statistics than the majority. It seems that people there spent much more time lying in bed after waking up.
* This data could be used to help people track their spending time. And for what exactly they are spending their time. Therefore, people could use this information to improve their time management.
* Going further the system could track the time spent productive and unproductive and store the information into the system. Afterwards it could use the information when people have been unproductive and come up with suggestions what they could do instead to spend their time better.
* Now I would like to combine these two data frames to see if there is a correlation between sleeping time and being active. Therefore, I will create a new data frame and name it ‘combined\_data’.

combined\_data <- merge(Sleep\_Day, Daily\_activity, by=”Id”)

* After that I will create another visualization to see if there is anything which is correlated to one another.

ggplot(data=combined\_data)+

geom\_smooth(mapping = aes(x=TotalSteps,y=TotalMinutesAsleep))

Chart, line chart

Description automatically generated

* Comparing TotalSteps and TotalMinutesAsleep you can see that people who don’t do any activities sleep the most. The more steps people make the less time they spend sleeping.
* This data could also be crucial in defining how to spend the time if the participants decide to sleep less and instead using this time to do any activity. The system which stored the data could use this information to send either feedback or suggestions to the product owner how to spend the time.
* But I want to have more reassurance about people spending time on activities and what is the outcome of it.
* For this I want to combine the same data frame I have combined before with a different data frame. I will use ‘Daily\_activity’ and combine it with ‘Weight\_log\_info’. I will name the new data frame ‘Steps\_Weight\_info’.

Steps\_Weight\_info <- merge(Daily\_activity,Weight\_log\_info, by=”Id”)

* After merging these two data frames into one I decide to combine the TotalSteps with the WeightKg into a visualization to see if people who do more activities also weight less. I also want to combine a line graph with a scatter plot to see if there are any exceptions or outliers.

ggplot(data=Steps\_Weight\_info)+

geom\_smooth(mapping = aes(x=TotalSteps,y=WeightKg))+

geom\_point(mapping = aes(x=TotalSteps,y=WeightKg))

Chart, scatter chart

Description automatically generated

* Starting at 0 you can see in bottom left people who do not do any activities and weight over 120Kg. There are some exceptions who weight under 100 and less who do not do any activities but those are just a handful. After further investigation I can say that people who do activities weight less.
* The more steps people do the less they weight.
* There are some outliers in this graph too. On the right side of it you can see people who made a tremendous number of steps and still weight 90Kg. I would say that those people are overweight and therefore they decided to exercise much more than the average.