Fitbit users insights for guided decisions

Antonio Barrera Mora

2022-07-13

Table of Contents

Fitbit users insights for guided decisions

Bellebeat main website corporative image

0. Introduction

This work is a study case part of the eighth course “Google Data Analytics Capstone” of the “Google Data Analyst” program.

Although it’s no the first time that I had perform a data analysis, both at the academic and professional level, it’s the first time following the methodology proposed in the study program, by serving the R programming language and the database query language (SQL).

Under normal circumstances, with the data we had from the start, *this work would not have been possible*, but I had priories on putting the skills learned into practice and carrying out this case study in a relatively short period of time, less than a week.

1. Ask Phase.

Bellabeat is a successful, small, high-tech company of health products for women. The heads believe that analyzing competence device data could help unlock growth chances. We should find insights in the data about the user’s behavior and make suggestions.

1.1. Business tasks

Due to find new opportunities to grow business, we will analyze competence smart device usage data by gain insights into the uses. Do apply these insights into one Bellabeat product and make recommendations.

1.2. Key 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? ### 1.3. Stakeholders

- **Urška Sršen:** Bellabeat's co-founder and Chief Creative Officer, with a marked background as an artist, aimed to develop beautifully designed technology.
- **Sando Mur:** Mathematician and Bellabeat's co-founder.
- **Bellabeat marketing analytics team.**

2. Preparation

1. The dataset from which we are recommended to start working is public. It refers to a set of data on consumption habits carried out through "Amazon Mechanical Turk" between 10/4/2016 and 12/05/2016, where the respondents (30 chosen) agreed to share the data (biometrics, minute-level output for physical activity, heart rate, and sleep monitoring) of theirs wearable devices for prospective study purposes.

2. The information is stored in long format, although some specific tables are arranged in wide format. Especially, the most important tables, those that collect the information grouped by larger time intervals (case of "dailyActivity_merged") are configured in long format in relation to the date.

3. **The data does not meet the ROCC parameters.** The information is not reliable, since they do not specify more parameters than user ID numbers, we do not know if the information contains some kind of bias. For example, we do not know the gender of the user, if this survey has been carried out only by men.

What is the point of applying the discoveries made here to a smart device designed for women? Likewise, we do not know ethnicity, nationality and most importantly, the age of the respondents.

About the data of dataset creation, we should say **the data set isn't current**, it dates from 2016, six years old. We can say that, when talking about technology, **six years is the prehistory**.

Finally and none the less, the information isn't original, the data set has been retouched to be published on the "Kaggle" platform.

For all these reasons, **we cannot consider the information reliable at all**.

4. **About Data integrity**, the datasets are in .csv format, meeting the integrity requirements with a fair level of confidence. Not for less, the datasets has been obtained from a platform whose members are passionate about data science. However, we confirmed the integrity analyzing the data set using some R programming language functions.

5. Although the data is clearly compromised, **we can still draw some conclusions that can help us meet our goals**.

6. In normal circumstances, a meeting with the stakeholders would have to be held. It would be necessary for them to agree to carry out their own survey and to provide data and primary information, that is, that is in the possession of the company.

Also, if it did not exceed the scope and requirements of this work, I would propose incorporating other open data, such as this Apple dataset:Apple Watch and Fitbit data, a much more complete and in tune with the ROCCC parameters.

2.1. Loading Datasets

```
fb_dailyAct <- read.csv("fb_data/dailyActivity_merged.csv")
fb_dailyCal <- read.csv("fb_data/dailyCalories_merged.csv")
fb_dailyInt <- read.csv("fb_data/dailyIntensities_merged.csv")
fb_dailySteps <- read.csv("fb_data/dailySteps_merged.csv")
fb_heartrate_sec <-
read.csv("fb_data/heartrate_seconds_merged.csv")
fb_hourlyCal <- read.csv("fb_data/hourlyCalories_merged.csv")
fb_hourlyInt <- read.csv("fb_data/hourlyIntensities_merged.csv")
fb_hourlySteps <- read.csv("fb_data/hourlySteps_merged.csv")
fb_minuteCaloriesNarrow <-
read.csv("fb_data/minuteCaloriesNarrow_merged.csv")
fb_minuteCaloriesWide <-
read.csv("fb_data/minuteCaloriesWide_merged.csv")
fb_minuteIntensitiesNarrow <-
read.csv("fb_data/minuteIntensitiesNarrow_merged.csv")
```

```
fb_minuteIntensitiesWide <-
read.csv("fb_data/minuteIntensitiesWide_merged.csv")
fb_minuteSleep <- read.csv("fb_data/minuteSleep_merged.csv")
fb_minuteStepsNarrow <-
read.csv("fb_data/minuteStepsNarrow_merged.csv")
fb_minuteStepsWide <-
read.csv("fb_data/minuteStepsWide_merged.csv")
fb_sleepDay <- read.csv("fb_data/sleepDay_merged.csv")
fb_weightLogInfo <- read.csv("fb_data/weightLogInfo_merged.csv")
fb_minuteMETsNarrow <-
read.csv("fb_data/weightLogInfo_merged.csv")
```

2.2. Conecting to a SQL Dataframe

Since the table frame with the heart rate is relevant to the analysis, and since its size is considerable, we decided to work with this data from Bigquery, combining the use of R and SQL language, while implementing some visualizations from Tableau:

```
library(DBI)
con <- dbConnect(odbc::odbc(), "Bellabeat", timeout = 10)
```

We will need to load an additional library:

```
library(RMySQL)
```

Loading the data set “heartrate_seconds_merged.csv in the Rstudio environment from bigQuery environment:

```
fb_heartrate_sec <- dbReadTable(con, "fb_heartrate_sec")
```

As a result, we obtain this table:

```
head(fb_heartrate_sec)

##   int64_field_0         Id Value     time       date
## 1        154299 2026352035   106 09:37:30 2016-04-25
## 2        154300 2026352035   108 09:37:35 2016-04-25
## 3        154326 2026352035   107 09:41:50 2016-04-25
## 4        154327 2026352035   108 09:41:55 2016-04-25
## 5        154328 2026352035   108 09:42:10 2016-04-25
## 6        154329 2026352035   107 09:42:25 2016-04-25
```

Finally, we had all the packages we need to be able to work with R in combination with datasets hosted in Bigquery and to use SQL.

2.3. Loading Libraries

Loading the R libraries nedeed in our Rstudio envionment:

```
library("rmarkdown")
library("tidyr")
library("tibble")
library("ggplot2")
library("skimr")
library("tibble")
library("janitor")

##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

library("kableExtra")

## Warning in !is.null(rmarkdown::metadata$output) &&
rmarkdown::metadata$output
## %in% : 'length(x) = 3 > 1' in coercion to 'logical(1)'

library("dplyr")

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library("tidyverse")

## ── Attaching packages ───────────────────────────────────────
tidyverse 1.3.1 ──

## ✔ readr   2.1.2     ✔ stringr 1.4.0
## ✔ purrr   0.3.4     ✔ forcats 0.5.1

## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::group_rows() masks kableExtra::group_rows()
## ✖ dplyr::lag()        masks stats::lag()
```

3. Process

3.1. Viewing datasets

As a summary of the visualization and complete study of all the data sets, we show the most relevant results of the tables that group the data in a wide interval (daily activity) as a sample.

```
head(fb_dailyAct)

##           Id ActivityDate TotalSteps TotalDistance
TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50
8.50
## 2 1503960366    4/13/2016      10735          6.97
6.97
## 3 1503960366    4/14/2016      10460          6.74
6.74
## 4 1503960366    4/15/2016       9762          6.28
6.28
## 5 1503960366    4/16/2016      12669          8.16
8.16
## 6 1503960366    4/17/2016       9705          6.48
6.48
##   LoggedActivitiesDistance VeryActiveDistance
ModeratelyActiveDistance
## 1                        0               1.88
0.55
## 2                        0               1.57
0.69
## 3                        0               2.44
0.40
## 4                        0               2.14
1.26
## 5                        0               2.71
0.41
## 6                        0               3.19
0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
Calories
## 1                  13                  328              728
1985
## 2                  19                  217              776
1797
## 3                  11                  181             1218
1776
## 4                  34                  209              726
1745
## 5                  10                  221              773
1863
## 6                  20                  164              539
1728

skim_without_charts("fb_dailyAct")
```

Data summary

Name

"fb_dailyAct"

Number of rows

1

Number of columns

1

Column type frequency:

character

1

Group variables

None

Variable type: character

skim_variable

n_missing

complete_rate

min

max

empty

n_unique

whitespace

data

0

1

11

11

0

1

0

```
summarise(fb_dailyAct)

## data frame with 0 columns and 1 row
```

3.2. Datasets elimination

We decided to eliminate the data sets that are structured in small time intervals (minutes) and because they have redundant data compared to datasets mad with broad time periods (Daily Grouped Data). We maintain, therefore, the “Dailyactivity, Sleepday and Weightloginfo” tables, which we will group in a single table (fb_Final_daily).

We also maintain the “Heartrate_Seconds” table, for being relevant to research, whose “Heart Rate” variable we will convert to minutes and from which we will create an additional variable with the average

3.3. Adjusting and cleaning variables in datasets

```
#Hourly Intensity
fb_hourlyInt$ActivityHour=as.POSIXct(fb_hourlyInt$ActivityHour,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())

#New time variable
```

```r
fb_hourlyInt$time <- format(fb_hourlyInt$ActivityHour, format =
"%H:%M:%S")

#New date variable
fb_hourlyInt$date <- format(fb_hourlyInt$ActivityHour, format =
"%m/%d/%Y")

#Erasing duplicates
fb_hourlyInt$ActivityHour <- NULL

#backup
write.csv(fb_hourlyInt, file=
"fb_data/hourlyIntensities_merged2.csv")

#Hourly Calories format fixing
fb_hourlyCal$ActivityHour=as.POSIXct(fb_hourlyCal$ActivityHour,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_hourlyCal$time <- format(fb_hourlyCal$ActivityHour, format =
"%H:%M:%S")
fb_hourlyCal$date <- format(fb_hourlyCal$ActivityHour, format =
"%m/%d/%Y")
fb_hourlyCal$ActivityHour <- NULL
write.csv(fb_hourlyCal, file=
"fb_data/hourlyCalories_merged2.csv")

#Fixing date format in fb_dailyAct
fb_dailyAct$ActivityDate=as.POSIXct(fb_dailyAct$ActivityDate,
format="%m/%d/%Y", tz=Sys.timezone())
fb_dailyAct$date <- format(fb_dailyAct$ActivityDate, format =
"%m/%d/%Y")
fb_dailyAct$ActivityDate <- NULL
write.csv(fb_dailyAct, file= "fb_data/dailyActivity_merged2.csv")

#Fixing date format data in fb_sleepDay
fb_sleepDay$SleepDay=as.POSIXct(fb_sleepDay$SleepDay,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_sleepDay$date <- format(fb_sleepDay$SleepDay, format =
"%m/%d/%Y")
fb_sleepDay$SleepDay <- NULL
write.csv(fb_sleepDay, file= "fb_data/sleepDay_merged2.csv")

#5a. Fixing date format fb_heartrate_sec
fb_heartrate_sec$Time=as.POSIXct(fb_heartrate_sec$Time,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_heartrate_sec$time <- format(fb_heartrate_sec$Time, format =
"%H:%M:%S")
```

```
fb_heartrate_sec$date <- format(fb_heartrate_sec$Time, format =
"%m/%d/%y")
fb_heartrate_sec$Time <- NULL
write.csv(fb_heartrate_sec, file=
"fb_data/heartrate_seconds_merged2.csv")

#Fixing date format in fb_hourlySteps
fb_hourlySteps$ActivityHour=as.POSIXct(fb_hourlySteps$ActivityHour
, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_hourlySteps$time <- format(fb_hourlySteps$ActivityHour, format
= "%H:%M:%S")
fb_hourlySteps$date <- format(fb_hourlySteps$ActivityHour, format
= "%m/%d/%Y")
fb_hourlySteps$ActivityHour <- NULL
write.csv(fb_hourlySteps, file= "fb_data/hourlySteps_merged2.csv")

#Fixing date format in fb_minuteCaloriesNarrow
fb_minuteCaloriesNarrow$ActivityMinute=as.POSIXct(fb_minuteCalorie
sNarrow$ActivityMinute, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
fb_minuteCaloriesNarrow$time <-
format(fb_minuteCaloriesNarrow$ActivityMinute, format = "%H:%M:
%S")
fb_minuteCaloriesNarrow$date <-
format(fb_minuteCaloriesNarrow$ActivityMinute, format =
"%m/%d/%y")
fb_minuteCaloriesNarrow$ActivityMinute <- NULL
write.csv(fb_minuteCaloriesNarrow, file=
"fb_data/minuteCaloriesNarrow_merged2.csv")

#Fixing date format in fb_minuteCaloriesWide
fb_minuteCaloriesWide$ActivityHour=as.POSIXct(fb_minuteCaloriesWid
e$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_minuteCaloriesWide$time <-
format(fb_minuteCaloriesWide$ActivityHour, format = "%H:%M:%S")
fb_minuteCaloriesWide$date <-
format(fb_minuteCaloriesWide$ActivityHour, format = "%m/%d/%y")
fb_minuteCaloriesWide$ActivityHour <- NULL
write.csv(fb_minuteCaloriesWide, file=
"fb_data/minuteCaloriesWide_merged2.csv")

#Fixing date format fb_minuteIntensitiesNarrow
fb_minuteIntensitiesNarrow$ActivityMinute=as.POSIXct(fb_minuteInte
nsitiesNarrow$ActivityMinute, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
fb_minuteIntensitiesNarrow$time <-
format(fb_minuteIntensitiesNarrow$ActivityMinute, format = "%H:%M:
```

```
%S")
fb_minuteIntensitiesNarrow$date <-
format(fb_minuteIntensitiesNarrow$ActivityMinute, format =
"%m/%d/%y")
fb_minuteIntensitiesNarrow$ActivityMinute <- NULL
write.csv(fb_minuteIntensitiesNarrow, file=
"fb_data/minuteIntensitiesNarrow_merged2.csv")

#Fixing date format in fb_minuteIntensitiesWide
fb_minuteIntensitiesWide$ActivityHour=as.POSIXct(fb_minuteIntensit
iesWide$ActivityHour, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
fb_minuteIntensitiesWide$time <-
format(fb_minuteIntensitiesWide$ActivityHour, format = "%H:%M:%S")
fb_minuteIntensitiesWide$date <-
format(fb_minuteIntensitiesWide$ActivityHour, format = "%m/%d/%y")
fb_minuteIntensitiesWide$ActivityHour <- NULL
write.csv(fb_minuteIntensitiesWide, file=
"fb_data/minuteIntensitiesWide_merged2.csv")


#Fixing date format in fb_minuteSleep
fb_minuteSleep$date=as.POSIXct(fb_minuteSleep$date,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_minuteSleep$time <- format(fb_minuteSleep$date, format = "%H:
%M:%S")
fb_minuteSleep$datev_2 <- format(fb_minuteSleep$date, format =
"%m/%d/%Y")
fb_minuteSleep$date <- NULL
write.csv(fb_minuteSleep, file= "fb_data/minuteSleep_merged2.csv")


#Fixing date format in fb_minuteStepsNarrow
fb_minuteStepsNarrow$ActivityMinute=as.POSIXct(fb_minuteStepsNarro
w$ActivityMinute, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
fb_minuteStepsNarrow$time <-
format(fb_minuteStepsNarrow$ActivityMinute, format = "%H:%M:%S")
fb_minuteStepsNarrow$date <-
format(fb_minuteStepsNarrow$ActivityMinute, format = "%m/%d/%y")
fb_minuteStepsNarrow$ActivityMinute <- NULL
write.csv(fb_minuteStepsNarrow, file=
"fb_data/minuteStepsNarrow_merged2.csv")


#Fixing date format in fb_minuteStepsWide
```

```
fb_minuteStepsWide$ActivityHour=as.POSIXct(fb_minuteStepsWide$Acti
vityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_minuteStepsWide$time <- format(fb_minuteStepsWide$ActivityHour,
format = "%H:%M:%S")
fb_minuteStepsWide$date <- format(fb_minuteStepsWide$ActivityHour,
format = "%m/%d/%y")
fb_minuteStepsWide$ActivityHour <- NULL
write.csv(fb_minuteStepsWide, file=
"fb_data/minuteStepsWide_merged2.csv")

#Fixing date format in fb_weightLogInfo
fb_weightLogInfo$Date=as.POSIXct(fb_weightLogInfo$Date,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_weightLogInfo$time <- format(fb_weightLogInfo$Date, format =
"%H:%M:%S")
fb_weightLogInfo$date <- format(fb_weightLogInfo$Date, format =
"%m/%d/%Y")
fb_weightLogInfo$Date <- NULL

#Factorizing 'IsManualReport' and excluding unnecessary in
'fb_weightLogInfo
fb_weightLogInfo <- fb_weightLogInfo %>%
  select(-LogId) %>%
  mutate(IsManualReport = as.factor(IsManualReport))
write.csv(fb_weightLogInfo, file=
"fb_data/WeightLogInfo_merged2.csv")

#Fixing date format in fb_minuteMETsNarrow
fb_minuteMETsNarrow$Date=as.POSIXct(fb_minuteMETsNarrow$Date,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
fb_minuteMETsNarrow$time <- format(fb_minuteMETsNarrow$Date,
format = "%H:%M:%S")
fb_minuteMETsNarrow$date <- format(fb_minuteMETsNarrow$Date,
format = "%m/%d/%y")
fb_minuteMETsNarrow$Date <- NULL
write.csv(fb_minuteMETsNarrow, file=
"fb_data/minuteMETsNarrow_merged2.csv")

#fb_dailyCal backup
write.csv(fb_dailyCal, file= "fb_data/dailyCalories_merged2.csv")

#fb_dailyInt backup
write.csv(fb_dailyInt, file=
"fb_data/dailyIntensities_merged2.csv")

#fb_dailySteps backup
```

```
write.csv(fb_dailySteps, file= "fb_data/dailySteps_merged2.csv")

#Renaming variables for uniformity
fb_dailyCal <- fb_dailyCal %>%
  mutate(date = ActivityDay) %>%
  select(-ActivityDay)

fb_dailyInt <- fb_dailyInt %>%
  mutate(date = ActivityDay) %>%
  select(-ActivityDay)

fb_dailySteps <- fb_dailySteps %>%
  mutate(date = ActivityDay) %>%
  select(-ActivityDay)
```

3.4. Merging Datsets in R

We will combine 3 Datasets ("fb_dailyAct", "fb_sleepDay", "fb_weightLogInfo"), after having cleaned and reviewed each one of them and ensured that they contain variables of the same type and name, to ensure their compatibility and that can merge without problems:

```
fb_final_daily <- merge(merge(fb_dailyAct, fb_sleepDay, by=
c('Id','date'), all = TRUE ), fb_weightLogInfo, by=
c('Id','date'), all = TRUE)
```

Thus, we have the "fb_final_daily" dataset from which we can work more comfortably and adequately, which we will create a backup of:

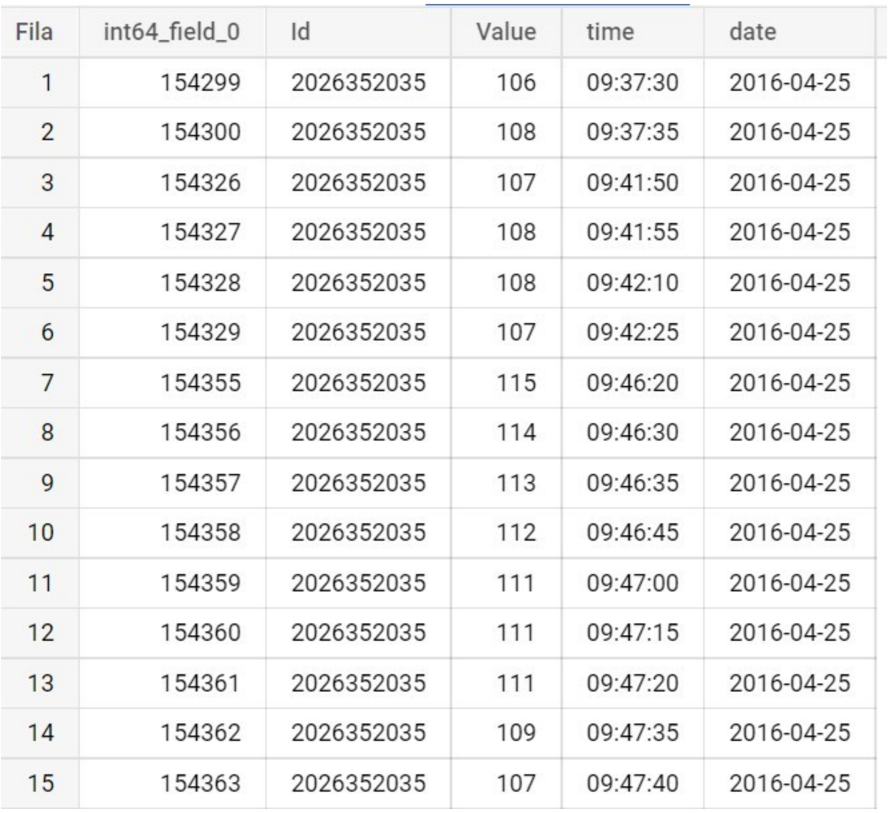
```
write.csv(fb_final_daily, file= "fb_data/fb_final_daily.csv")

fb_final_daily <- read.csv("fb_data/fb_final_daily.csv")
```

3.5. Working with Heart-Rate table in SQL

As we said before, due to size issues, we need to import the "heartrate_seconds_merged.csv" table into BigQuery and next, add it to the Rstudio workbench:

```
fb_heartrate_sec <- dbReadTable(con, "fb_heartrate_sec")
```

obtaining the next table:



Fila	int64_field_0	Id	Value	time	date
1	154299	2026352035	106	09:37:30	2016-04-25
2	154300	2026352035	108	09:37:35	2016-04-25
3	154326	2026352035	107	09:41:50	2016-04-25
4	154327	2026352035	108	09:41:55	2016-04-25
5	154328	2026352035	108	09:42:10	2016-04-25
6	154329	2026352035	107	09:42:25	2016-04-25
7	154355	2026352035	115	09:46:20	2016-04-25
8	154356	2026352035	114	09:46:30	2016-04-25
9	154357	2026352035	113	09:46:35	2016-04-25
10	154358	2026352035	112	09:46:45	2016-04-25
11	154359	2026352035	111	09:47:00	2016-04-25
12	154360	2026352035	111	09:47:15	2016-04-25
13	154361	2026352035	111	09:47:20	2016-04-25
14	154362	2026352035	109	09:47:35	2016-04-25
15	154363	2026352035	107	09:47:40	2016-04-25

Heart Rate in BigQuery

Figure 1:Heart-rate table in BigQuery

3.5.1. Cleaning the “fb_heartrate_sec”

We need to obtain the average of the heartbeats per hour and clean the variables, so we proceed through SQL to perform these tasks

```
-- !preview conn=con
SELECT
date AS ymd,
Id,
ROUND(AVG(Value),2)  AS Heartrate

FROM `bellabeat-356005.Bellabeat.fb_heartrate_sec`

GROUP BY
date, Id

ORDER BY
Id
```

And saving a new bigQuery table “c_fb_heartrateAvg”, then loading in the RStudio environment:

```
fb_heartrateAvg <- dbReadTable(con, "c_fb_heartrateAvg")
```

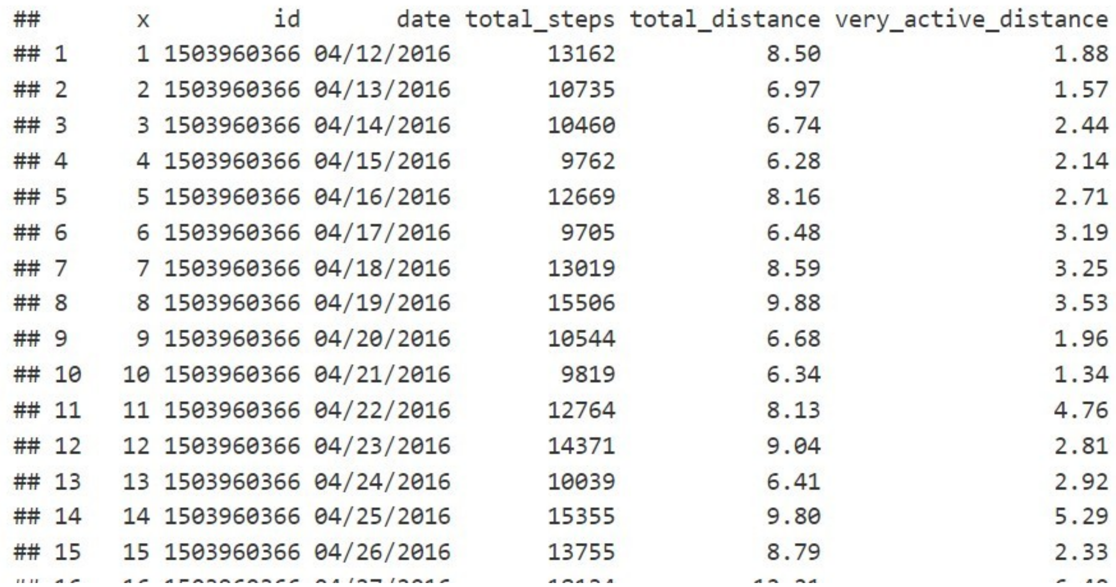
And obtaining the next table:

```
head(fb_heartrateAvg)

##          ymd         Id Heartrate
## 1 2016-04-12 2022484408     75.80
## 2 2016-04-13 2022484408     80.34
## 3 2016-04-14 2022484408     72.63
## 4 2016-04-15 2022484408     80.44
## 5 2016-04-16 2022484408     75.96
## 6 2016-04-17 2022484408     83.92
```

3.6. Unified dataset check

```
        clean_names(fb_final_daily)
```

As a snapshot of the result process:



```
##       x         id       date total_steps total_distance very_active_distance
## 1     1 1503960366 04/12/2016       13162           8.50                 1.88
## 2     2 1503960366 04/13/2016       10735           6.97                 1.57
## 3     3 1503960366 04/14/2016       10460           6.74                 2.44
## 4     4 1503960366 04/15/2016        9762           6.28                 2.14
## 5     5 1503960366 04/16/2016       12669           8.16                 2.71
## 6     6 1503960366 04/17/2016        9705           6.48                 3.19
## 7     7 1503960366 04/18/2016       13019           8.59                 3.25
## 8     8 1503960366 04/19/2016       15506           9.88                 3.53
## 9     9 1503960366 04/20/2016       10544           6.68                 1.96
## 10   10 1503960366 04/21/2016        9819           6.34                 1.34
## 11   11 1503960366 04/22/2016       12764           8.13                 4.76
## 12   12 1503960366 04/23/2016       14371           9.04                 2.81
## 13   13 1503960366 04/24/2016       10039           6.41                 2.92
## 14   14 1503960366 04/25/2016       15355           9.80                 5.29
## 15   15 1503960366 04/26/2016       13755           8.79                 2.33
```

Figure 2:“Clean_names” function summary

```
glimpse(fb_heartrateAvg)

## Rows: 334
## Columns: 3
## $ ymd       <date> 2016-04-12, 2016-04-13, 2016-04-14, 2016-04-
15, 2016-04-16,…
## $ Id        <int64> 2022484408, 2022484408, 2022484408,
2022484408, 2022484408…
```

```
## $ Heartrate <dbl> 75.80, 80.34, 72.63, 80.44, 75.96, 83.92,
82.71, 81.95, 83.4…
```

```
head(fb_final_daily)
```

```
##   X         Id       date TotalSteps TotalDistance
VeryActiveDistance
## 1 1 1503960366 04/12/2016      13162          8.50
1.88
## 2 2 1503960366 04/13/2016      10735          6.97
1.57
## 3 3 1503960366 04/14/2016      10460          6.74
2.44
## 4 4 1503960366 04/15/2016       9762          6.28
2.14
## 5 5 1503960366 04/16/2016      12669          8.16
2.71
## 6 6 1503960366 04/17/2016       9705          6.48
3.19
##   ModeratelyActiveDistance LightActiveDistance
SedentaryActiveDistance
## 1                     0.55                6.06
0
## 2                     0.69                4.71
0
## 3                     0.40                3.91
0
## 4                     1.26                2.83
0
## 5                     0.41                5.04
0
## 6                     0.78                2.51
0
##   VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
SedentaryMinutes
## 1                25                  13                  328
728
## 2                21                  19                  217
776
## 3                30                  11                  181
1218
## 4                29                  34                  209
726
## 5                36                  10                  221
773
## 6                38                  20                  164
539
```

```
##   Calories TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
WeightKg Fat BMI
## 1     1985                 1                327            346
NA  NA  NA
## 2     1797                 2                384            407
NA  NA  NA
## 3     1776                NA                 NA             NA
NA  NA  NA
## 4     1745                 1                412            442
NA  NA  NA
## 5     1863                 2                340            367
NA  NA  NA
## 6     1728                 1                700            712
NA  NA  NA
##   time
## 1 <NA>
## 2 <NA>
## 3 <NA>
## 4 <NA>
## 5 <NA>
## 6 <NA>

skim_without_charts(fb_final_daily)
```

Data summary

Name

fb_final_daily

Number of rows

943

Number of columns

21

Column type frequency:

character

2

numeric

19

Group variables

None

Variable type: character

skim_variable

n_missing

complete_rate

min

max

empty

n_unique

whitespace

date

0

1.00

10

10

0

31

0

time

876

0.07

8

8

0

26

0

Variable type: numeric

skim_variable

n_missing

complete_rate

mean

sd

p0

p25

p50

p75

p100

X

0

1.00

4.720000e+02

2.723600e+02

1.00000e+00

2.365000e+02

4.720000e+02

7.075000e+02

9.430000e+02

Id

0

1.00

```
4.858486e+09
2.423712e+09
1.50396e+09
2.320127e+09
4.445115e+09
6.962181e+09
8.877689e+09
TotalSteps
0
1.00
7.652190e+03
5.086530e+03
0.00000e+00
3.795000e+03
7.439000e+03
1.073400e+04
3.601900e+04
TotalDistance
0
1.00
5.500000e+00
3.930000e+00
0.00000e+00
2.620000e+00
5.260000e+00
7.720000e+00
```

2.803000e+01

VeryActiveDistance

0

1.00

1.500000e+00

2.660000e+00

0.00000e+00

0.000000e+00

2.200000e-01

2.060000e+00

2.192000e+01

ModeratelyActiveDistance

0

1.00

5.700000e-01

8.800000e-01

0.00000e+00

0.000000e+00

2.400000e-01

8.100000e-01

6.480000e+00

LightActiveDistance

0

1.00

3.350000e+00

2.050000e+00

0.00000e+00

1.950000e+00

3.380000e+00

4.790000e+00

1.071000e+01

SedentaryActiveDistance

0

1.00

0.000000e+00

1.000000e-02

0.00000e+00

0.000000e+00

0.000000e+00

0.000000e+00

1.100000e-01

VeryActiveMinutes

0

1.00

2.124000e+01

3.295000e+01

0.00000e+00

0.000000e+00

4.000000e+00

3.200000e+01

2.100000e+02

FairlyActiveMinutes

```
0
1.00
1.363000e+01
2.000000e+01
0.00000e+00
0.000000e+00
7.000000e+00
1.900000e+01
1.430000e+02
LightlyActiveMinutes
0
1.00
1.930300e+02
1.093100e+02
0.00000e+00
1.270000e+02
1.990000e+02
2.640000e+02
5.180000e+02
SedentaryMinutes
0
1.00
9.903500e+02
3.012600e+02
0.00000e+00
7.290000e+02
```

```
1.057000e+03
1.229000e+03
1.440000e+03
Calories
0
1.00
2.307510e+03
7.208200e+02
0.00000e+00
1.829500e+03
2.140000e+03
2.796500e+03
4.900000e+03
TotalSleepRecords
530
0.44
1.120000e+00
3.500000e-01
1.00000e+00
1.000000e+00
1.000000e+00
1.000000e+00
3.000000e+00
TotalMinutesAsleep
530
0.44
```

4.194700e+02

1.183400e+02

5.80000e+01

3.610000e+02

4.330000e+02

4.900000e+02

7.960000e+02

TotalTimeInBed

530

0.44

4.586400e+02

1.271000e+02

6.10000e+01

4.030000e+02

4.630000e+02

5.260000e+02

9.610000e+02

WeightKg

876

0.07

7.204000e+01

1.392000e+01

5.26000e+01

6.140000e+01

6.250000e+01

8.505000e+01

1.335000e+02

Fat

941

0.00

2.350000e+01

2.120000e+00

2.20000e+01

2.275000e+01

2.350000e+01

2.425000e+01

2.500000e+01

BMI

876

0.07

2.519000e+01

3.070000e+00

2.14500e+01

2.396000e+01

2.439000e+01

2.556000e+01

4.754000e+01

3.6.1 Variable cleaning in the final tables

```
fb_final_daily$X <- NULL %>%
  fb_final_daily$LoggedActivitiesDistance <- NULL %>%
  fb_final_daily$TrackerDistance <- NULL %>%
  fb_final_daily$IsManualReport <- NULL %>%
  fb_final_daily$Date <- NULL %>%
  fb_final_daily$WeightPounds <- NULL %>%
```

4. Analyze

We could start this section, summarizing the state of affairs, which would happen by saying that we have obtained as a product, two tables with which we are going to proceed with the analysis: -“fb_final_daily” -“fb_heartrateAvg”

Before starting the analysis is needed to mention that this section contains insights and ideas from the MIGUEL FZZZdesign.

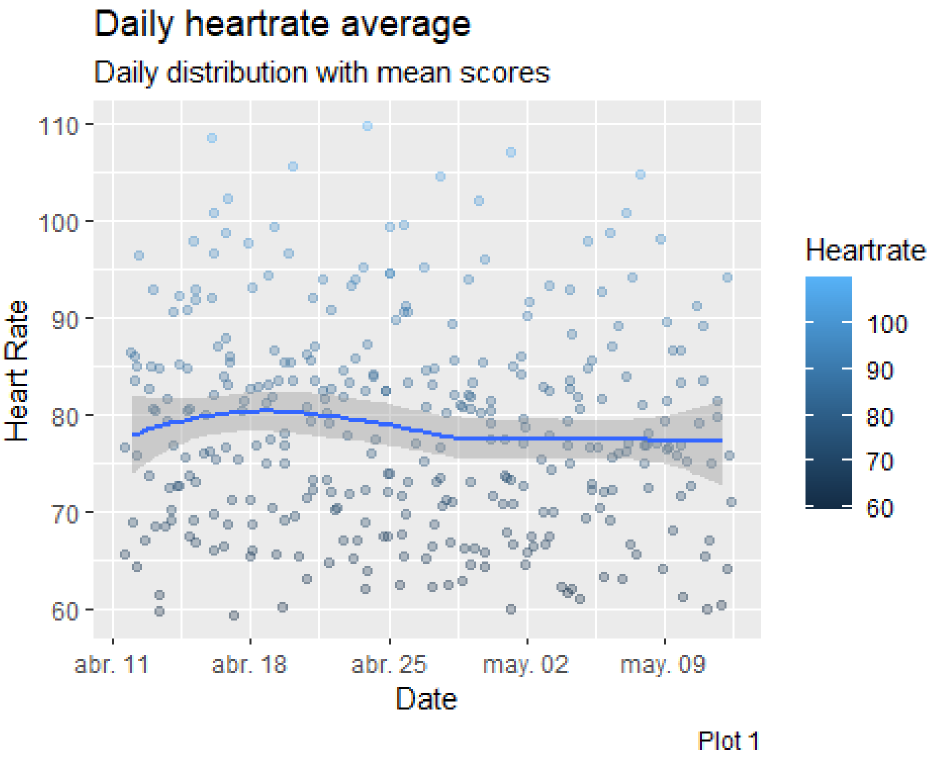
First, we need to set a theme for the plots:

```
custom_theme_original <- function() {
  theme(
    panel.border = element_rect(colour = "black",
                                fill = NA,
                                linetype = 1),
    panel.background = element_rect(fill = "white",
                                    color = 'grey50'),
    panel.grid.minor.y = element_blank(),
    axis.text = element_text(colour = "blue",
                             face = "italic",
                             family = "Arial"),
    axis.title = element_text(colour = "gray",
                              family = "Arial"),
    axis.ticks = element_line(colour = "blue"),
    plot.title = element_text(size=20,
                              hjust = 0.5,
                              family = "Arial"),
    plot.subtitle=element_text(size=13,
                               hjust = 0.5),
    plot.caption = element_text(colour = "brown",
                                face = "Arial",
                                family = "Arial")
  )
}
```

4.1 Physiological activity:Heart-rate as a predictor of health problems

```
fb_heartrateAvg %>%
  group_by(Id) %>%
  ggplot(aes(x=ymd, y=Heartrate,  color=Heartrate)) +
  geom_point(alpha=0.3, position = position_jitter())+
  geom_smooth()+
  labs(title = "Daily heartrate average", subtitle= "Daily
distribution with mean scores", x= "Date", y="Heart Rate", caption
= "Plot 1")
```

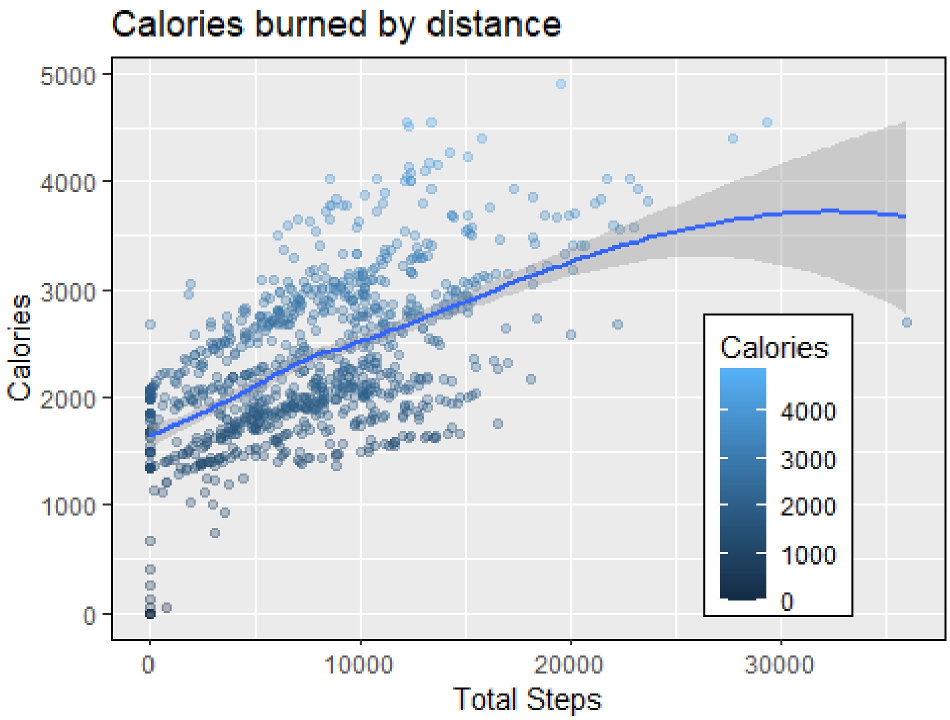
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



You can display an interactive plot by clicking **[here]** (https://public.tableau.com/app/profile/anbamo/viz/BellabeatInsightsfromFitbitHeartRate-Date/Physiological)

4.2 Physical activity 1: Calories by activity (total distance)

```
fb_final_daily %>%
 group_by(TotalDistance, Calories) %>%
  ggplot(aes(x = TotalSteps, y = Calories, color = Calories)) +
  geom_point(alpha=0.3, position = position_jitter()) +
  geom_smooth() +
  theme(legend.position = c(.8, .3),
        legend.spacing.y = unit(1, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black")) +
  labs(title = 'Calories burned by distance',
       y = 'Calories',
       x = 'Total Steps',
       caption = 'Plot 2')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



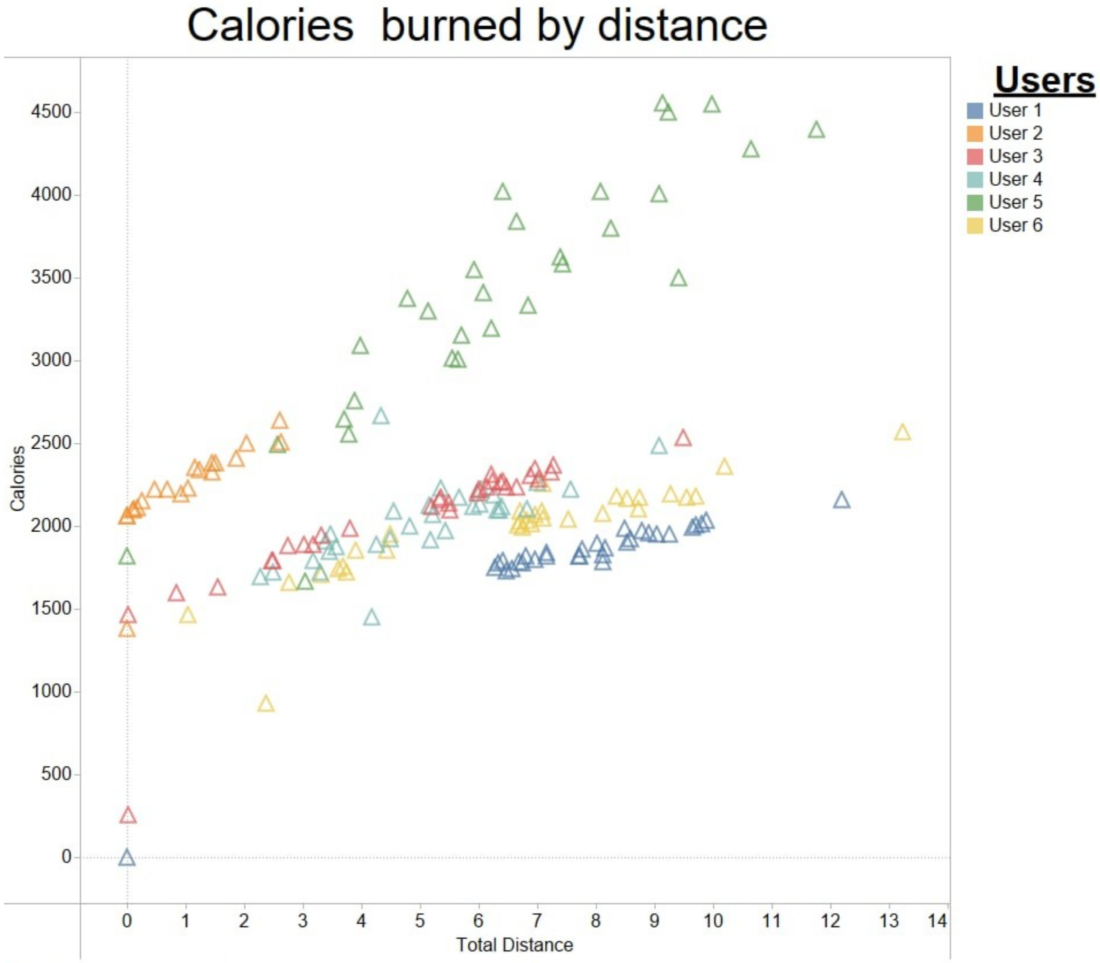
Plot 2 **Pearson correlation index**

```
cor.test(fb_final_daily$TotalDistance, fb_final_daily$Calories,
method = 'pearson', conf.level = 0.95)
```

```
##
##  Pearson's product-moment correlation
##
## data:  fb_final_daily$TotalDistance and fb_final_daily$Calories
## t = 26.002, df = 941, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6078539 0.6822785
## sample estimates:
##       cor
## 0.6466023
```

You can display an interactive plot by clicking here

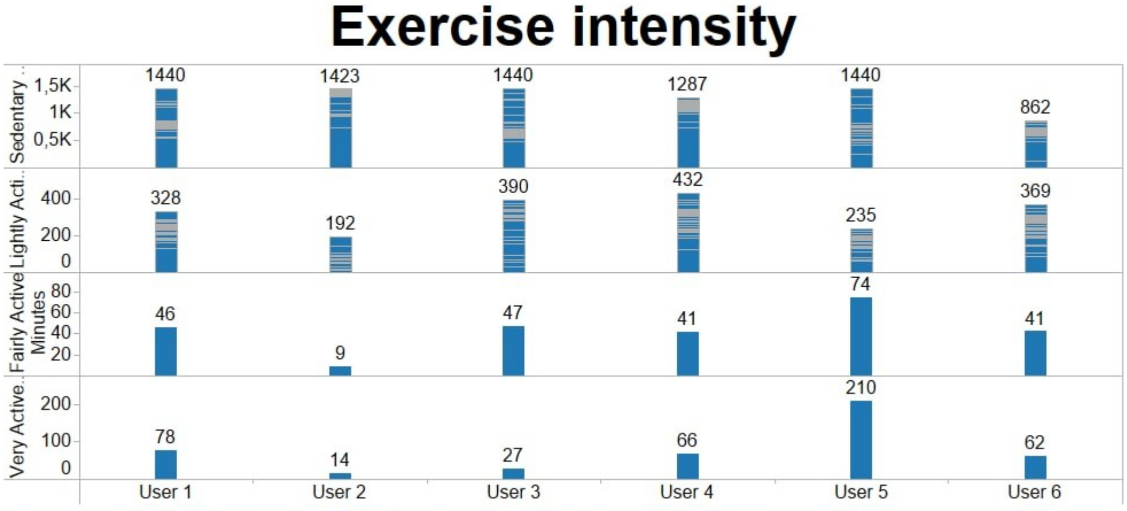
4.3 Physical Activity 2: Calories by activity (total distance)



Total Distance vs. Calories. Colour shows details about Id (a fb finalDaily).

Plot 3: Daily Activity

4.4 Intensity of exercise activity



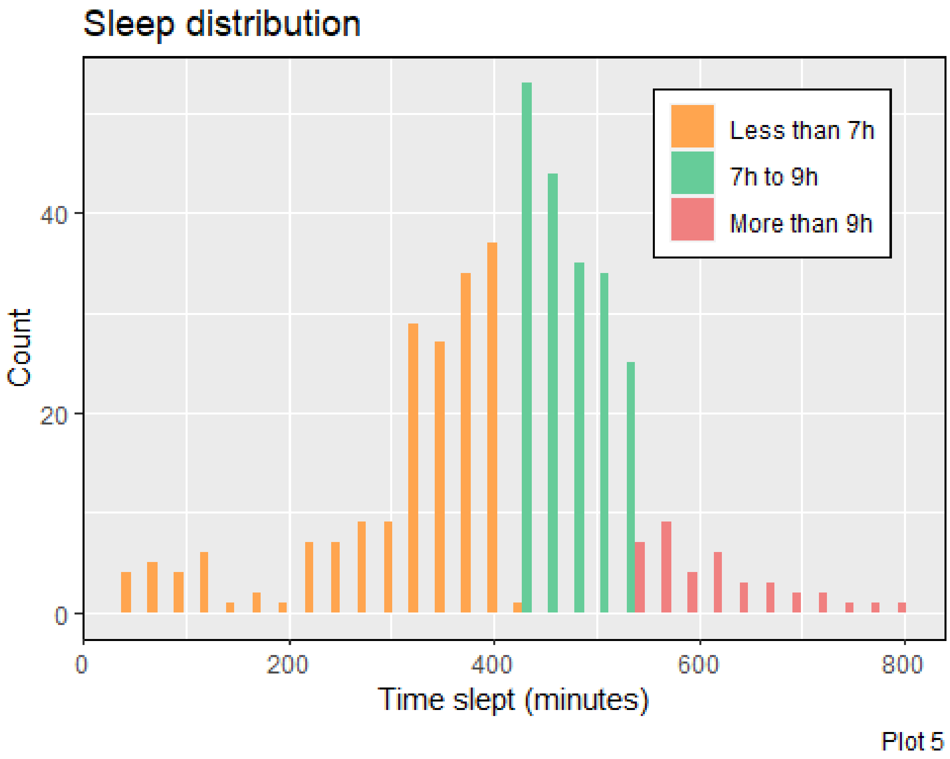
Sedentary Minutes,Lightly Active Minutes,Fairly Active MinutesandVery Active Minutes for each Id (a fb finalDaily). For pane Sedentary Minutes: Details are shown for Id (a fb finalDaily). For pane Lightly Active Minutes: Details are shown for Id (a fb finalDaily).

Plot 4: Excercise Intensity

You can display an interactive plot by clicking here

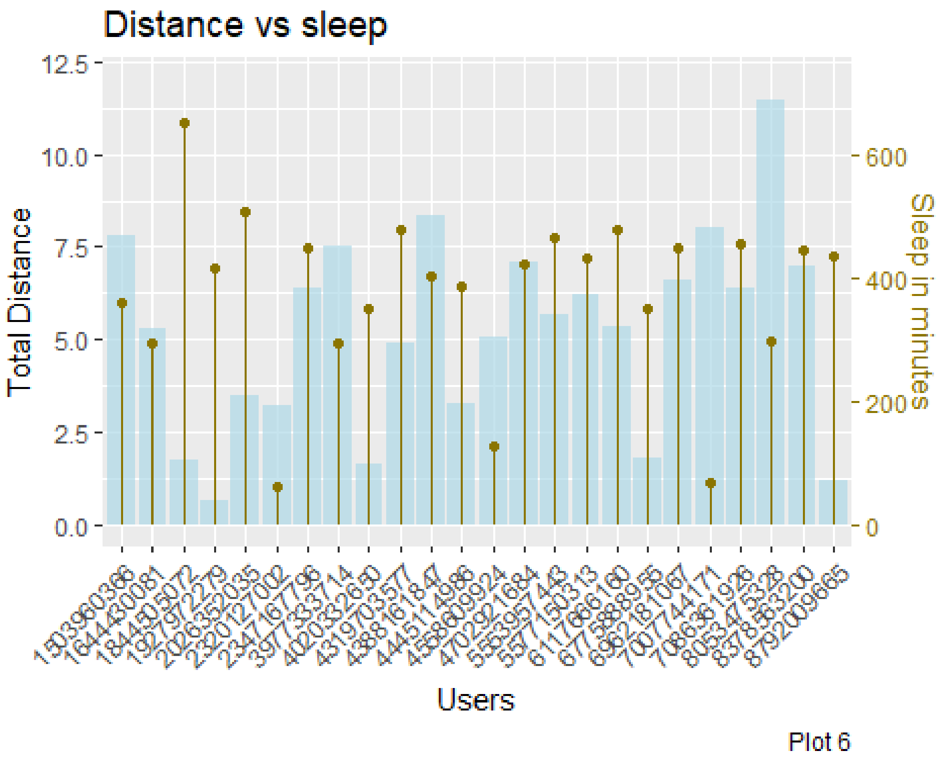
4.5 Sleep distribution

```
fb_final_daily %>%
  select(TotalMinutesAsleep) %>%
  drop_na() %>%
  mutate(sleep_quality = ifelse(TotalMinutesAsleep <= 420, 'Less
than 7h',
                         ifelse(TotalMinutesAsleep <= 540, '7h to
9h',
                         'More than 9h'))) %>%
  mutate(sleep_quality = factor(sleep_quality,
                          levels = c('Less than 7h','7h to 9h',
                                     'More than 9h'))) %>%
  ggplot(aes(x = TotalMinutesAsleep, fill = sleep_quality)) +
  geom_histogram(position = 'dodge', bins = 30) +
  scale_fill_manual(values=c("tan1", "#66CC99", "lightcoral")) +
  theme(legend.position = c(.80, .80),
        legend.title = element_blank(),
        legend.spacing.y = unit(0, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black")) +
  labs(
    title = "Sleep distribution",
    x = "Time slept (minutes)",
    y = "Count",
    caption = 'Plot 5'
  )
```



4.6 Sleep vs distance covered

```
fb_final_daily%>%
  select(Id, TotalDistance, TotalMinutesAsleep) %>%
  group_by(Id) %>%
  summarise_all(list(~mean(., na.rm=TRUE))) %>%
  drop_na() %>%
  mutate(Id = factor(Id)) %>%
  ggplot() +
  geom_bar(aes(x = Id, y = TotalDistance), stat = "identity", fill
= 'lightblue', alpha = 0.7) +
  geom_point(aes(x = Id, y = TotalMinutesAsleep/60), color =
'gold4') +
  geom_segment(aes(x = Id, xend = Id, y = 0, yend =
TotalMinutesAsleep/60), color = 'gold4' ,group = 1) +
  scale_y_continuous(limits=c(0, 12), name = "Total Distance",
    sec.axis = sec_axis(~.*60, name = "Sleep in minutes")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(axis.title.y.right = element_text(color = "gold4"),
        axis.ticks.y.right = element_line(color = "gold4"),
        axis.text.y.right = element_text(color = "gold4")) +
  labs(
    title = "Distance vs sleep",
    x = "Users",
    caption = 'Plot 6'
  )
```



5. Share

5.1. Weight key takeaways

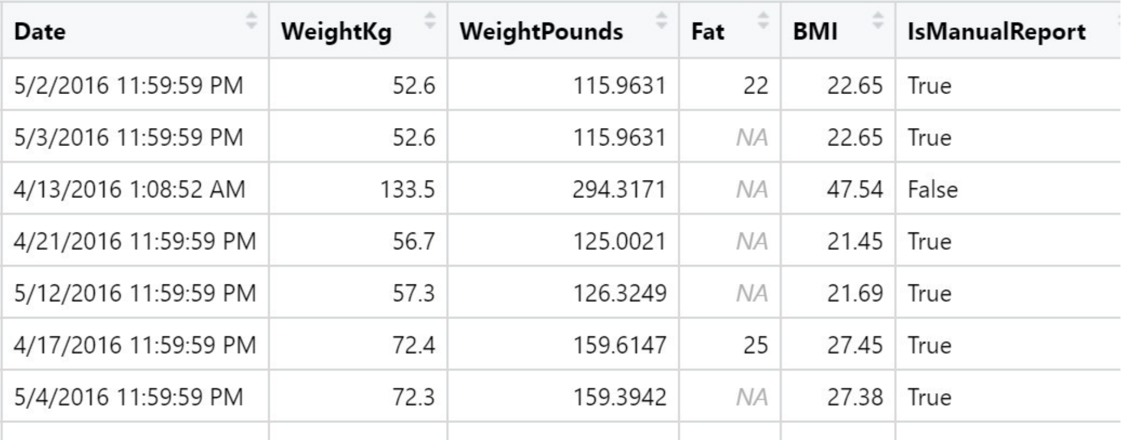
5.1.1 Weight as a key valor

1. Weight is one of the most important biometric measurements.

- As can be seen in this interactive graph, weight is a good predictor of physical activity, **in this case, the correlation between higher weight and lower intensity of physical activity and exercises**.

- In this interactive graph, we can see easily that a higher weight is synonymous with shorter distances traveled.

- That the few users who entered their real weight did so manually, as we can see in this snapshot:



Date	WeightKg	WeightPounds	Fat	BMI	IsManualReport
5/2/2016 11:59:59 PM	52.6	115.9631	22	22.65	True
5/3/2016 11:59:59 PM	52.6	115.9631	NA	22.65	True
4/13/2016 1:08:52 AM	133.5	294.3171	NA	47.54	False
4/21/2016 11:59:59 PM	56.7	125.0021	NA	21.45	True
5/12/2016 11:59:59 PM	57.3	126.3249	NA	21.69	True
4/17/2016 11:59:59 PM	72.4	159.6147	25	27.45	True
5/4/2016 11:59:59 PM	72.3	159.3942	NA	27.38	True

Plot 5: Manual weight data introduction

5.1.2 Weight issues recomendations for "Bellabeat membership"

Weight is a recognized **medical risk factor for health**, but as we can deduce from the information analysed, we observe that it's a great predictor of physical activity. The subscription service should encourage the user to provide biometric data, but especially the weight, as it's vital for this subscription (pay) program to be really useful for our customers. In the same way, the technology behind scenes in the **app will be improves** for collect automatically the weight values. Finally, a rewards program should also be implemented to encourage physical activity.

5.2. Heart rate key takeways

5.2.1. Heart rate monitorization

Abnormal cardiological activity and other health problems can be clearly reflected in the pulse with heart rate monitoring.

The data records, in the format that was presented did not lend themselves to understanding whether too high a pulse rate corresponded to high physical activity. In many cases, we can see with a simple glance at the tables, which people with high weight had a much higher average heart rate.

5.2.2. Heart rate recomendations for "Bellabeat membership" and app

Both the subscription service and the app should implement artificial intelligence to understand when a heart rate is normal based on the physical activity that is taking place. Also manage to keep a record of the anomalies and the times that a high pulse has been had without correspondence of a physical activity that justifies it.

5.3. Ending with other considerations

In the case of other variables such as sleep, the analysis reflects an apparently normal distribution of sleep (in terms of quantity) and also when correlating in terms of distances traveled, that is, the higher the level of rest, the more willingness to accumulate steps, or what is the same, more physical activity. This is not surprising, since it’s something that falls within common sense. But it would be important for our company to study the sleep patterns based on age and moment, like **women ovulation** as a variable that can affect -among others-, the sleep quality.