Bellabeat: Case study

Chaymae Boussetta

2023-07-08

---

1. Introduction

1.1 Business Task

The goal of this project is to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices and how to apply these insights into Bellabeat's marketing strategy using these three 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?

---

2. Prepare the Data and Libraries in RStudio

Collect the data required for analysis but since the data is available on Kaggle publicly, FitBit Fitness Tracker Data (CC0: Public Domain) and download the dataset.

2.1 Data Limitation

- **Demographically-limited:** Bellebeat is a health tracker made specifically for women, it is important to know the **gender** of the data.
- **Time frame:** 31 days is limited to make any solid recommendation since there are seasons involved in a given month to consider someone's health well being.

Next, once the dataset's been downloaded, I prepare RStudio, an Integrated Development Environment (IDE) for R, a programming language for statistical computing and graphics. R itself can clean and make visualizations so it's my go-to cloud software.

2.2 Install and load the packages

Install the RStudio libraries for analysis and visualizations, then load the libraries

```
install.packages("tidyverse")  # core package for cleaning and analysis
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")  # date library mdy()
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")  # clean_names() to consists only _, character,
numbers, and letters.
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("ggpubr")  # for the donut chart ggdonutchart()
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
```

```
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(janitor)
```

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

```
library(lubridate)
library(ggpubr)
```

2.3 Import and Prepare the Dataset

Upload the archived dataset to RStudio by clicking the Upload button.

```
d_activity <- read_csv("dailyActivity_merged.csv")

## Rows: 940 Columns: 15
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance,
LoggedActivitiesDi...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

d_sleep <- read_csv("sleepDay_merged.csv")

## Rows: 413 Columns: 5
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

h_calories <- read_csv("hourlyCalories_merged.csv")

## Rows: 22099 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, Calories
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

h_intensities <- read_csv("hourlyIntensities_merged.csv")

## Rows: 22099 Columns: 4
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
```

```
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

h_steps <- read_csv("hourlySteps_merged.csv")

## Rows: 22099 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, StepTotal
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

2.4 Preview Dataset

```
head(d_activity)

## # A tibble: 6 × 15
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
##        <dbl> <chr>             <dbl>         <dbl>           <dbl>
## 1 1503960366 04/12/2016        13162          8.5             8.5
## 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
## # ℹ 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>

head(d_sleep)

## # A tibble: 6 × 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep
TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>
<dbl>
## 1 1503960366 04/12/2016                      1                327
346
## 2 1503960366 4/13/2016 12:0…                 2                384
407
## 3 1503960366 4/15/2016 12:0…                 1                412
442
## 4 1503960366 4/16/2016 12:0…                 2                340
```

```
367
## 5 1503960366 4/17/2016 12:0…                 1                700
712
## 6 1503960366 4/19/2016 12:0…                 1                304
320

head(h_calories)

## # A tibble: 6 × 3
##           Id ActivityHour          Calories
##        <dbl> <chr>                    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       81
## 2 1503960366 4/12/2016 1:00:00 AM        61
## 3 1503960366 4/12/2016 2:00:00 AM        59
## 4 1503960366 4/12/2016 3:00:00 AM        47
## 5 1503960366 4/12/2016 4:00:00 AM        48
## 6 1503960366 4/12/2016 5:00:00 AM        48

head(h_intensities)

## # A tibble: 6 × 4
##           Id ActivityHour          TotalIntensity AverageIntensity
##        <dbl> <chr>                          <dbl>            <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM             20            0.333
## 2 1503960366 4/12/2016 1:00:00 AM               8            0.133
## 3 1503960366 4/12/2016 2:00:00 AM               7            0.117
## 4 1503960366 4/12/2016 3:00:00 AM               0            0
## 5 1503960366 4/12/2016 4:00:00 AM               0            0
## 6 1503960366 4/12/2016 5:00:00 AM               0            0

head(h_steps)

## # A tibble: 6 × 3
##           Id ActivityHour          StepTotal
##        <dbl> <chr>                     <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366 4/12/2016 1:00:00 AM        160
## 3 1503960366 4/12/2016 2:00:00 AM        151
## 4 1503960366 4/12/2016 3:00:00 AM          0
## 5 1503960366 4/12/2016 4:00:00 AM          0
## 6 1503960366 4/12/2016 5:00:00 AM          0

colnames(d_activity)

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

```
colnames(d_sleep)

## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"

colnames(h_calories)

## [1] "Id"           "ActivityHour" "Calories"

colnames(h_intensities)

## [1] "Id"               "ActivityHour"     "TotalIntensity"
"AverageIntensity"

colnames(h_steps)

## [1] "Id"           "ActivityHour" "StepTotal"
```

3. Data Cleaning

With the data assigned to their own values and recognizing the data structures, I can start the cleaning process. The goal of cleaning is to find: * **Data type**: Values must be of a certain type. * **Data range**: Values must fall between predefined maximum and minimum values. * **Mandatory values**: Ensure the values can't be left blank or empty. * **Unique**: No duplications. * **Regular expression (regex) patterns**: Values must match a prescribed pattern. * **Cross-field validation**: Certain conditions for multiple fields must be satisfied. Eg. Percentages must add up to 100%. * **Accuracy**: The data conforms to the actual entity being measured or described. Eg. zip codes are validated by street location. * **Completeness**: Data contains all desired components or described. * **Consistency**: Data is repeatable from different points of entry or collection.

```
glimpse(d_activity)

## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366,
150396036…
## $ ActivityDate             <chr> "04/12/2016", "4/13/2016", "4/14/2016",
"4/15…
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705,
13019…
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59,
9.8…
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59,
9.8…
## $ LoggedActivitiesDistance <dbl> 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,
3.5…
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64,
```

```
1.3…
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71,
5.0…
## $ SedentaryActiveDistance  <dbl> 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, 4…
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8,
27, 21…
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264,
205, …
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775,
818…
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921,
203…

glimpse(d_sleep)

## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366,
150…
## $ SleepDay           <chr> "04/12/2016", "4/13/2016 12:00:00 AM",
"4/15/2016 1…
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361,
430, 2…
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384,
449, 3…

glimpse(h_calories)

## Rows: 22,099
## Columns: 3
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366,
150396036…
## $ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM",
"4/12/20…
## $ Calories     <dbl> 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73,
66, …

glimpse(h_intensities)

## Rows: 22,099
## Columns: 4
## $ Id               <dbl> 1503960366, 1503960366, 1503960366, 1503960366,
15039…
## $ ActivityHour     <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM",
"4/1…
## $ TotalIntensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6,
```

```
36, 5…
## $ AverageIntensity <dbl> 0.333333, 0.133333, 0.116667, 0.000000, 0.000000,
0.0…

glimpse(h_steps)

## Rows: 22,099
## Columns: 3
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366,
150396036…
## $ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM",
"4/12/20…
## $ StepTotal    <dbl> 373, 160, 151, 0, 0, 0, 0, 0, 250, 1864, 676, 360,
253, 2…
```

Checked and verified data type is accurate and consistent.

```
clean_names(d_activity)

## # A tibble: 940 × 15
##            id activity_date total_steps total_distance tracker_distance
##         <dbl> <chr>               <dbl>          <dbl>            <dbl>
##  1 1503960366 04/12/2016          13162           8.5              8.5
##  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
##  7 1503960366 4/18/2016           13019           8.59             8.59
##  8 1503960366 4/19/2016           15506           9.88             9.88
##  9 1503960366 4/20/2016           10544           6.68             6.68
## 10 1503960366 4/21/2016            9819           6.34             6.34
## # ℹ 930 more rows
## # ℹ 10 more variables: logged_activities_distance <dbl>,
## #   very_active_distance <dbl>, moderately_active_distance <dbl>,
## #   light_active_distance <dbl>, sedentary_active_distance <dbl>,
## #   very_active_minutes <dbl>, fairly_active_minutes <dbl>,
## #   lightly_active_minutes <dbl>, sedentary_minutes <dbl>, calories <dbl>

clean_names(d_sleep)

## # A tibble: 413 × 5
##          id sleep_day total_sleep_records total_minutes_asleep
total_time_in_bed
##       <dbl> <chr>                  <dbl>                <dbl>
<dbl>
##  1   1.50e9 04/12/20…                  1                  327
346
##  2   1.50e9 4/13/201…                  2                  384
407
##  3   1.50e9 4/15/201…                  1                  412
```

```
442
##  4   1.50e9 4/16/201…                    2                  340
367
##  5   1.50e9 4/17/201…                    1                  700
712
##  6   1.50e9 4/19/201…                    1                  304
320
##  7   1.50e9 4/20/201…                    1                  360
377
##  8   1.50e9 4/21/201…                    1                  325
364
##  9   1.50e9 4/23/201…                    1                  361
384
## 10   1.50e9 4/24/201…                    1                  430
449
## # ℹ 403 more rows

clean_names(h_calories)

## # A tibble: 22,099 × 3
##            id activity_hour         calories
##         <dbl> <chr>                   <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM       81
##  2 1503960366 4/12/2016 1:00:00 AM        61
##  3 1503960366 4/12/2016 2:00:00 AM        59
##  4 1503960366 4/12/2016 3:00:00 AM        47
##  5 1503960366 4/12/2016 4:00:00 AM        48
##  6 1503960366 4/12/2016 5:00:00 AM        48
##  7 1503960366 4/12/2016 6:00:00 AM        48
##  8 1503960366 4/12/2016 7:00:00 AM        47
##  9 1503960366 4/12/2016 8:00:00 AM        68
## 10 1503960366 4/12/2016 9:00:00 AM       141
## # ℹ 22,089 more rows

clean_names(h_intensities)

## # A tibble: 22,099 × 4
##            id activity_hour         total_intensity average_intensity
##         <dbl> <chr>                          <dbl>             <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM             20             0.333
##  2 1503960366 4/12/2016 1:00:00 AM               8             0.133
##  3 1503960366 4/12/2016 2:00:00 AM               7             0.117
##  4 1503960366 4/12/2016 3:00:00 AM               0             0
##  5 1503960366 4/12/2016 4:00:00 AM               0             0
##  6 1503960366 4/12/2016 5:00:00 AM               0             0
##  7 1503960366 4/12/2016 6:00:00 AM               0             0
##  8 1503960366 4/12/2016 7:00:00 AM               0             0
##  9 1503960366 4/12/2016 8:00:00 AM              13             0.217
## 10 1503960366 4/12/2016 9:00:00 AM              30             0.5
## # ℹ 22,089 more rows
```

```
clean_names(h_steps)
```

```
## # A tibble: 22,099 × 3
##            id activity_hour         step_total
##         <dbl> <chr>                      <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM        373
##  2 1503960366 4/12/2016 1:00:00 AM         160
##  3 1503960366 4/12/2016 2:00:00 AM         151
##  4 1503960366 4/12/2016 3:00:00 AM           0
##  5 1503960366 4/12/2016 4:00:00 AM           0
##  6 1503960366 4/12/2016 5:00:00 AM           0
##  7 1503960366 4/12/2016 6:00:00 AM           0
##  8 1503960366 4/12/2016 7:00:00 AM           0
##  9 1503960366 4/12/2016 8:00:00 AM         250
## 10 1503960366 4/12/2016 9:00:00 AM        1864
## # ℹ 22,089 more rows
```

To make sure the names are consistent and doesn't contain any special characters.

```
sum(duplicated(d_activity))
```

```
## [1] 0
```

```
sum(duplicated(d_sleep))
```

```
## [1] 3
```

```
sum(duplicated(h_calories))
```

```
## [1] 0
```

```
sum(duplicated(h_intensities))
```

```
## [1] 0
```

```
sum(duplicated(h_steps))
```

```
## [1] 0
```

```
# Duplicates found and removed.
d_sleep <- d_sleep[!duplicated(d_sleep), ]
```

Duplicates found 3 at the **d_sleep** data and removed.

```
sum(is.na(d_activity))
```

```
## [1] 0
```

```
sum(is.na(d_sleep))
```

```
## [1] 0
```

```
sum(is.na(h_calories))
```

```
## [1] 0

sum(is.na(h_intensities))

## [1] 0

sum(is.na(h_steps))

## [1] 0
```

Mandatory values: Check and remove any NA values. Since the column in **weight_info** has too many empty values, the column "Fat" is removed.

3.1 Data Formatting

```
# d_activity table
d_activity <- d_activity %>%
  rename(date = ActivityDate) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

# d_sleep table
d_sleep <- d_sleep %>%
  rename(date = SleepDay) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `date = as_date(date, format = "%m/%d/%Y")`.
## Caused by warning:
## !  251 failed to parse.
```

I will be joining the data frame of **d_sleep** into **d_activity** data frame. So I need to make the date format consistent and formatted between them.

```
# h_calories table
h_calories<- h_calories %>%
  rename(date_time = ActivityHour) %>%
  mutate(date_time = as.POSIXct(date_time,format ="%m/%d/%Y %I:%M:%S %p" ,
tz=Sys.timezone()))

# h_intensities
h_intensities<- h_intensities %>%
  rename(date_time = ActivityHour) %>%
  mutate(date_time = as.POSIXct(date_time,format ="%m/%d/%Y %I:%M:%S %p" ,
tz=Sys.timezone()))

# h_steps
h_steps<- h_steps %>%
  rename(date_time = ActivityHour) %>%
  mutate(date_time = as.POSIXct(date_time,format ="%m/%d/%Y %I:%M:%S %p" ,
tz=Sys.timezone()))
```

For the hourly tables, I will format the date time into a the **24-hour clock** type since it is currently using the **12-hour clock**.

3.2 Data Merging

```
# Merge of d_activity + d_sleep
d_merged <- merge(d_activity, d_sleep, by = c("Id", "date"))

# Merge of h_calories + h_intensities + h_steps
h_calories_intensities <- merge(h_calories, h_intensities, by = c("Id",
"date_time"))

# Merge the h_calories_intensities with h_steps to have the full data
h_merged <- merge(h_calories_intensities, h_steps, by = c("Id", "date_time"))

# Check the new table with head()
head(d_merged)
```

```
##           Id       date TotalSteps TotalDistance TrackerDistance
## 1 1503960366 2016-04-12      13162          8.50            8.50
## 2 1503960366 2016-05-01      10602          6.81            6.81
## 3 1503960366 2016-05-02      14727          9.71            9.71
## 4 1503960366 2016-05-03      15103          9.66            9.66
## 5 1503960366 2016-05-05      14070          8.90            8.90
## 6 1503960366 2016-05-06      12159          8.03            8.03
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               2.29                     1.60
## 3                        0               3.21                     0.57
## 4                        0               3.73                     1.05
## 5                        0               2.92                     1.08
## 6                        0               1.97                     0.25
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                2.92                       0                33
## 3                5.92                       0                41
## 4                4.88                       0                50
## 5                4.88                       0                45
## 6                5.81                       0                24
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  35                  246              730     1820
## 3                  15                  277              798     2004
## 4                  24                  254              816     1990
## 5                  24                  250              857     1959
## 6                   6                  289              754     1896
##   TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1                 1                327            346
## 2                 1                369            396
## 3                 1                277            309
```

```
## 4                 1                273            296
## 5                 1                247            264
## 6                 1                334            367
```

```
head(h_merged)
```

```
##           Id           date_time Calories TotalIntensity AverageIntensity
## 1 1503960366 2016-04-12 00:00:00       81             20         0.333333
## 2 1503960366 2016-04-12 01:00:00       61              8         0.133333
## 3 1503960366 2016-04-12 02:00:00       59              7         0.116667
## 4 1503960366 2016-04-12 03:00:00       47              0         0.000000
## 5 1503960366 2016-04-12 04:00:00       48              0         0.000000
## 6 1503960366 2016-04-12 05:00:00       48              0         0.000000
##   StepTotal
## 1       373
## 2       160
## 3       151
## 4         0
## 5         0
## 6         0
```

Merge all the tables into two major tables for the final process of analysis and visualization.

4. Data Analysis

This is the part of analyzing the data by formatting and adjusting, identifying relationships and patterns between the data, and making calculations.

I will first the mean (average) steps for each user to find the amount of activity and put them into a new category.

```
d_avg_steps <- d_merged %>%
  group_by(Id) %>%
  summarise(avg_d_steps = mean(TotalSteps), avg_d_calories = mean(Calories),
avg_d_sleep = mean(TotalMinutesAsleep))

# Check the new table with head()
head(d_avg_steps)
```

```
## # A tibble: 6 × 4
##           Id avg_d_steps avg_d_calories avg_d_sleep
##        <dbl>       <dbl>          <dbl>       <dbl>
## 1 1503960366      12625.          1880.        342.
## 2 1644430081       5241           2784.        466.
## 3 1844505072       2573           1541         590
## 4 1927972279        678           2220         750
## 5 2026352035       6675.          1586.        499.
## 6 3977333714      12588           1609.        284.
```

Parameter:

- **Inactive:** less than **5,000** steps a day
- **Average (somewhat active):** ranges from **5,000** to **10,000** steps
- **Active:** above **10,000** steps

```
# Add new column to categorize user steps and sleep quality
active_users <- d_avg_steps %>%
mutate(active_users = case_when(
avg_d_steps < 5000 ~ "Inactive",
avg_d_steps >= 5000 & avg_d_steps < 9999 ~ "Average",
avg_d_steps >= 10000 & avg_d_steps < 12499 ~ "Active",
avg_d_steps > 12500 ~ "Very Active")) %>%
mutate(sleep_quality = case_when(
avg_d_sleep < 420 ~ "Insufficient Sleep",
avg_d_sleep >= 420 & avg_d_sleep < 540 ~ "Good Sleep",
avg_d_sleep > 540 ~ "Excessive Sleep" ))
# Check the new table with head()
head(active_users)
```

```
## # A tibble: 6 × 6
##           Id avg_d_steps avg_d_calories avg_d_sleep active_users
sleep_quality
##        <dbl>       <dbl>          <dbl>       <dbl> <chr>        <chr>
## 1 1503960366      12625.          1880.        342. Very Active
Insufficient S…
## 2 1644430081       5241           2784.        466. Average      Good
Sleep
## 3 1844505072       2573           1541         590  Inactive     Excessive
Sleep
## 4 1927972279        678           2220         750  Inactive     Excessive
Sleep
## 5 2026352035       6675.          1586.        499. Average      Good
Sleep
## 6 3977333714      12588           1609.        284. Very Active
Insufficient S…
```

Created a new table for further analysis and visualizations.

```
# Create a new percentage table from active_users
active_users_perc <- active_users %>%
group_by(active_users) %>%
summarise(total = n()) %>%
mutate(totals = sum(total)) %>%
group_by(active_users) %>%
summarise(total_percent = total / totals) %>%
mutate(labels = scales::percent(total_percent))

# And create percentage table for sleep_quality
sleep_users_perc <- active_users %>%
group_by(sleep_quality) %>%
summarise(total = n()) %>%
mutate(totals = sum(total)) %>%
```

```
group_by(sleep_quality) %>%
summarise(total_percent = total / totals) %>%
mutate(labels = scales::percent(total_percent))

# Check the new table with head()
 head(active_users_perc)

## # A tibble: 4 × 3
##   active_users total_percent labels
##   <chr>                <dbl> <chr>
## 1 Active              0.0476 5%
## 2 Average             0.571  57%
## 3 Inactive            0.190  19%
## 4 Very Active         0.190  19%

head(sleep_users_perc)

## # A tibble: 3 × 3
##   sleep_quality      total_percent labels
##   <chr>                      <dbl> <chr>
## 1 Excessive Sleep           0.0952 9.5%
## 2 Good Sleep                0.476  47.6%
## 3 Insufficient Sleep        0.429  42.9%
```

Clean up the unused tables to keep the Data Environment clean of aliases in the RStudio.

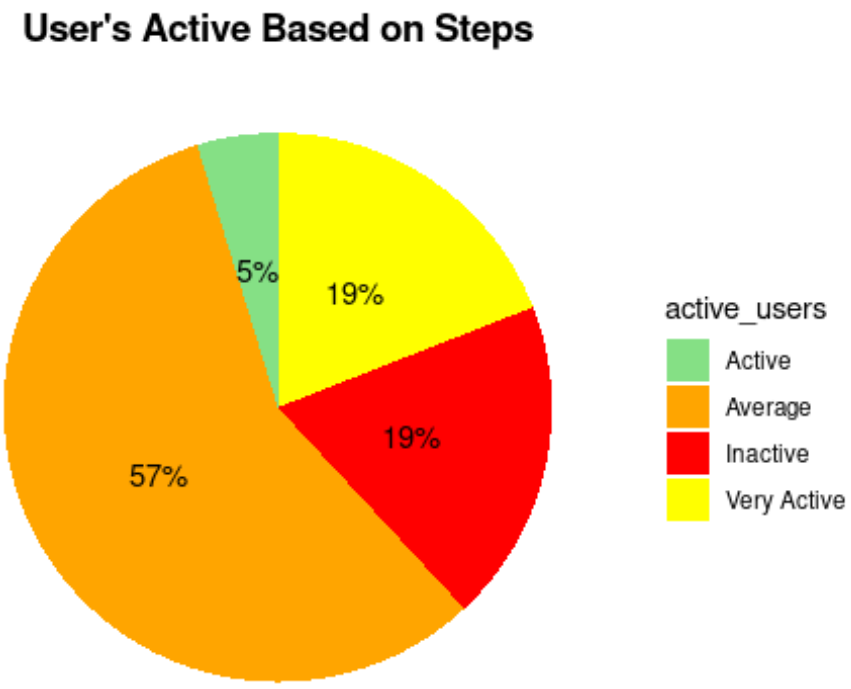
```
rm(d_avg_steps)
rm(d_sleep)
rm(h_calories)
rm(h_calories_intensities)
rm(h_intensities)
rm(h_steps)
```

5. Visualizations & Key Findings

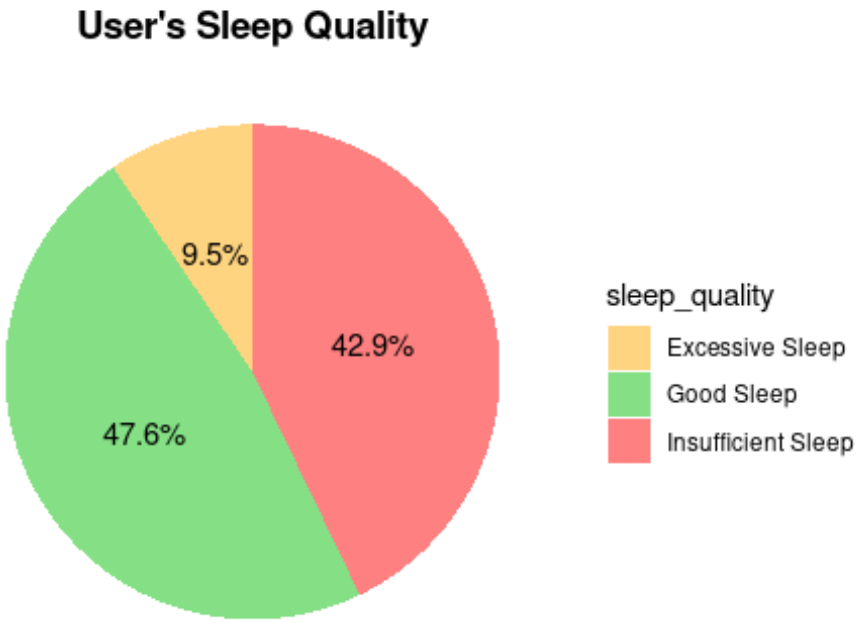
Create 2 pie charts to show the proportions of each active users and their sleep qualities.

```
# Pie Chart for Active Users
active_users_perc %>%
ggplot(aes(x="",y=total_percent, fill=active_users)) +
geom_bar(stat = "identity", width = 1)+
coord_polar("y", start=0)+
theme_minimal()+
theme(axis.title.x= element_blank(),
axis.title.y = element_blank(),
panel.border = element_blank(),
panel.grid = element_blank(),
axis.ticks = element_blank(),
axis.text.x = element_blank(),
plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
```

```
scale_fill_manual(values = c("#85e085","#FFA500", "#FF0000", "#FFFF00")) +
geom_text(aes(label = labels),
position = position_stack(vjust = 0.5))+
labs(title="User's Active Based on Steps")
```




```
# Pie Chart for User's Sleep Quality
sleep_users_perc %>%
ggplot(aes(x="",y=total_percent, fill=sleep_quality)) +
geom_bar(stat = "identity", width = 1)+
coord_polar("y", start=0)+
theme_minimal()+
theme(axis.title.x= element_blank(),
axis.title.y = element_blank(),
panel.border = element_blank(),
panel.grid = element_blank(),
axis.ticks = element_blank(),
axis.text.x = element_blank(),
plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
scale_fill_manual(values = c("#ffd480", "#85e085", "#ff8080")) +
geom_text(aes(label = labels),
position = position_stack(vjust = 0.5))+
labs(title="User's Sleep Quality")
```



5.1 Pie Chart's Observation

User's Active Based on Steps As shown on the first chart, the majority of people are averagely active (with 5,000 to 10,000 steps per day). The runner up is the inactive people and only a minority of 4.2% of the people are highly active (minimum of 12,500 steps per day).

User's Sleep Quality There is a near equal split between people who doesn't get enough sleep (less than 7 hours per day) in comparison to those who get the right amount of sleep (7 to 9 hours per day).

5.2 Correlation Coefficient: Calories vs. Steps

Now we need to see if there is a correlation between steps and calories. Does more steps equal to higher calorie counts or not? How strong is their relationship?

Correlation Coefficient 1. Very Weak 0.00 to 0.19 2. Weak 0.20 to 0.39 3. Moderate 0.40 to 0.59 4. Strong 0.60 to 0.79 5. Very Strong 0.80 to 1.0

```
d_merged %>%
group_by(TotalSteps, Calories) %>%
ggplot(aes(x = TotalSteps, y = Calories, color = Calories)) +
geom_point() +
geom_smooth(color = "blue") +
theme(legend.position = c(.8, .3),
legend.spacing.y = unit(2, "mm"),
panel.border = element_rect(colour = "black", fill=NA),
```

```
legend.background = element_blank(),
legend.box.background = element_rect(colour = "black")) +
labs(title = 'Calories vs. Total Steps',
y = 'Calories',
x = 'Total Steps',
caption = 'Data Source: FitBit Fitness Tracker Data')

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

After looking at the result, it's time to count the Correlation Coefficient between Calories vs. Steps.

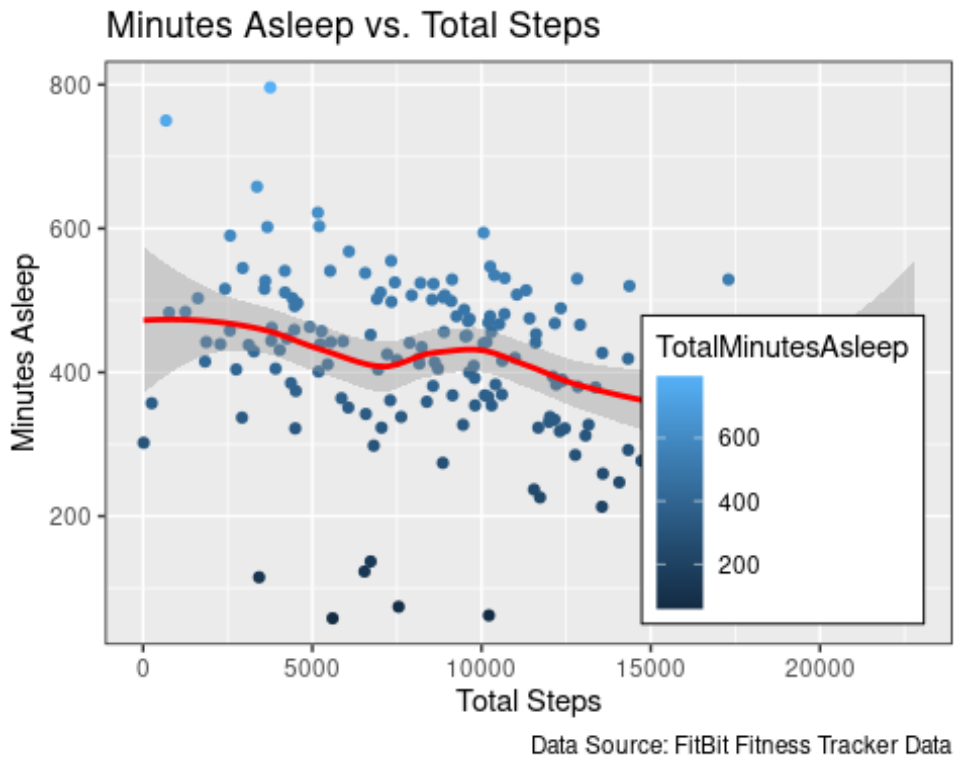
```
cor(d_merged$TotalSteps, d_merged$Calories)

## [1] 0.3437457

#It outputs as 0.4063007
```

After calculating the Correlation Coefficient, the result shows as **0.4063007.**

As the table above, it shows that there its relationship strength is **Moderate.**

```
d_merged %>%
group_by(TotalSteps, TotalMinutesAsleep) %>%
ggplot(aes(x = TotalSteps, y = TotalMinutesAsleep, color =
TotalMinutesAsleep)) +
geom_point() +
geom_smooth(color = "red") +
```

```
theme(legend.position = c(.8, .3),
legend.spacing.y = unit(2, "mm"),
panel.border = element_rect(colour = "black", fill=NA),
legend.background = element_blank(),
legend.box.background = element_rect(colour = "black")) +
labs(title = 'Minutes Asleep vs. Total Steps',
y = 'Minutes Asleep',
x = 'Total Steps',
caption = 'Data Source: FitBit Fitness Tracker Data')

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




```
cor(d_merged$TotalSteps, d_merged$TotalMinutesAsleep)

## [1] -0.2791272

#It outputs as -0.1903439
```

The result of **-0.1903439** shows that there is **no correlation** between the amount a user steps per day and the amount of sleep they have at that night.

After this discovered correlation, next, I need to find the average steps taken to discover which days tend to have more steps, more activities.

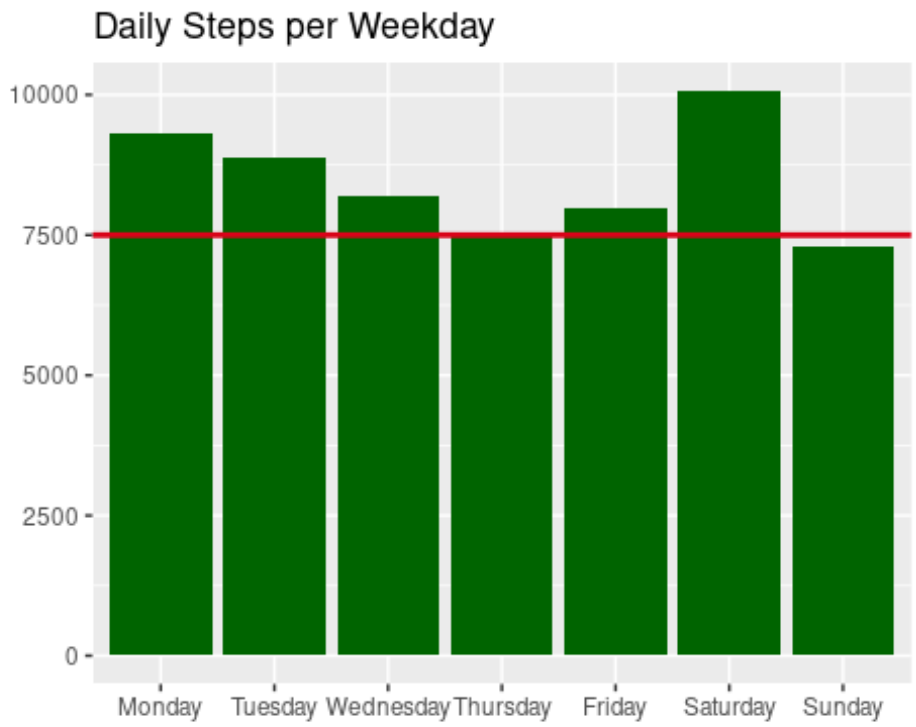
5.3 Bar Chart: Daily Average Sleeps & Steps

Prepare a new table for the bar chart to visualize which day, on average, has the most and least activities in a week

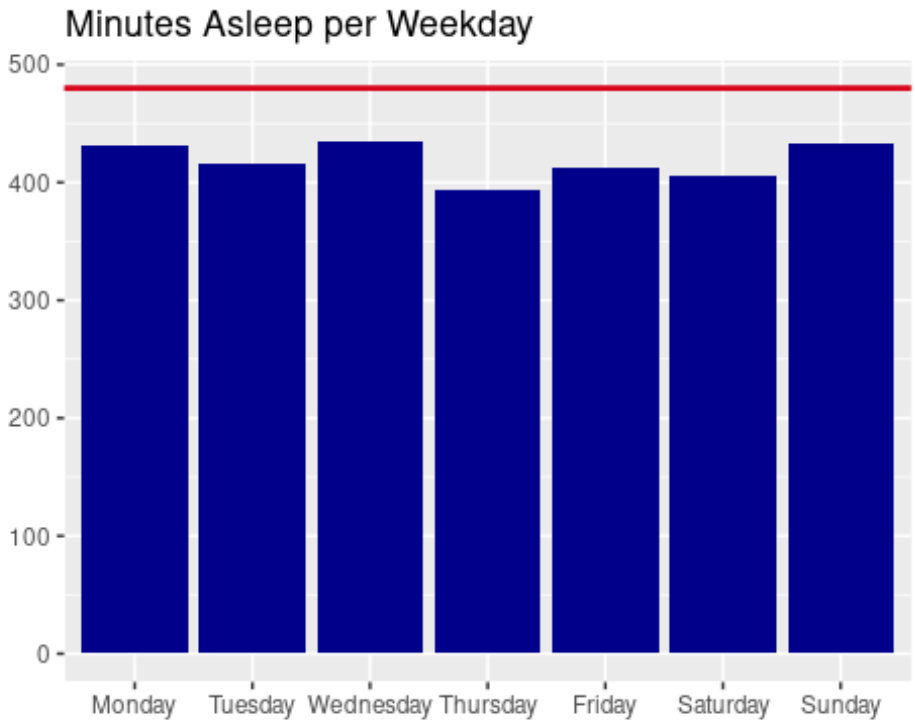
```r
# Table for bar charts
weekday_d <- d_merged %>%
mutate(weekday = weekdays(date), TotalSteps, TotalMinutesAsleep)

weekday_d$weekday <- ordered(weekday_d$weekday, levels = c("Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

weekday_d <- weekday_d %>%
group_by(weekday) %>%
summarize (daily_steps = mean(TotalSteps), daily_sleep =
mean(TotalMinutesAsleep))

# Check the new table with head()
head(weekday_d)
```

```
## # A tibble: 6 × 3
##   weekday   daily_steps daily_sleep
##   <ord>           <dbl>       <dbl>
## 1 Monday          9314.        431.
## 2 Tuesday         8870.        416.
## 3 Wednesday       8199.        434.
## 4 Thursday        7528.        394.
## 5 Friday          7983.        412.
## 6 Saturday       10083.        406.
```

```r
# Bar chart for Steps
ggplot(weekday_d, aes(weekday, daily_steps)) +
geom_col(fill = "#006400") +
geom_hline(yintercept = 7500, linewidth=1, color = "#D90319") +
labs(title = "Daily Steps per Weekday", x= "", y = "")
```




```
# Bar chart for Sleeps
ggplot(weekday_d, aes(weekday, daily_sleep)) +
geom_col(fill = "#00008B") +
geom_hline(yintercept = 480, linewidth=1, color = "#D90319") +
labs(title = "Minutes Asleep per Weekday", x = "", y = "")
```



Bar Chart for Steps

- Users are able to maintain a healthy average steps of around or above 7,500 steps per day except Sundays.
- Saturday has the highest amount of steps per day, knowing that Saturday is in the weekend.

Bar Chart for Sleeps

- Users did not meet the recommended amount of sleeps in minutes per day (8 hours) in any given day.
- The most amount of sleep users can get on average is on Sundays.

###5.4 Key Findings With the data cleaned and analyzed, it has provided valuable insights for Bellabeat's marketing strategy team.

- The majority of the users are moderately active and the next majority of the users are inactive (under 5,000 steps per day).
- 54% of the users didn't has insufficient amount of sleep and 42% has good sleep. It's fairly equal.
- There is a fair amount of positive correlation between steps and calories burnt per day.
- There is no correlation between the amount of steps and amount of sleep.
- Users are mostly active on Saturday, least active during Sundays, and follows up to a near-equal amount of activities on Monday and Tuesday.

- Users, on average did not get enough sleep every day. The most amount of sleep they get is only on Sunday.

6. Recommendations

Keeping in mind, Bellabeat is a high-tech manufacturer of health-focused products for women and the main goal of this project is to gain insight into how consumers use non-Bellabeat smart devices and provide high-level recommendations for how these trends can inform Bellabeat's marketing strategy.

6.1 Marketing Strategy Recommendations for Bellabeat are:

- With health-focused product in mind, focusing on the long-term health of the users is a priority. Since this is a women-based product, it is possible to collect more data by adding an additional feature to add their menstrual cycle the product so Bellabeat can have more in-depth analysis in the future, complimenting features that may have positive impact during their time of the month.
- Focus on the majority of the users, which are the average active people but still keep other users in mind. Provide solutions to the problem they are currently facing, with or without their knowledge that their lifestyle may impact their long term health. A small friendly reminder that it is important to maintain a good amount of sleep and remind them that it is only "x amount of steps left" to keep a active lifestyle.
- Adding a new feature, knowing their calories intake will help maintain a balanced calories amount of intake and outtake can be important. With the new data, it is possible to help users give a friendly reminder to not skip out meals, or remind if they had enough meal to consume or not.
- Further possibilities with this new data, it is possible to help users keep track of their goal, if they wanted to gain or lose weight with the in and out of calories per day.
- Gamification. WIth features that may have "Levels", compare to other users in a positive manner can motivate users to stay active and healthy. Making daily and weekly goals can help add "Experience" to the user's profile and have sharing feature after each successful goal. Referral system with rewards to other people can help them build communities focusing more with using Bellabeat as their main product.

6.2 Further Recommendations, Product Related:

With better products, users will have a better time using it. Collecting and processing health related data, it is ideal to have users to always use their products, even during sleep. With less down time, it is important to make further improvements on:

- Make the product lightweight and skin-friendly material.
- Longer battery life for less down time due to charging the product.
- Make the product gives an elegant sensation and universal design so it can fit most outfits. (Further data collection necessary).

Sometimes a short battery life can cause missing data due to users forgetting to wear their product again and leaving it at home.

High quality materials in a product can make users proud of wearing it, and with more users always wearing it, it advertises to people that Bellabeat is out there, making them remember that there is a health product that people can wear everyday.