Bellabeat Case Study

David Gonzalez

2025-03-02

Practical Case Report – Google Data Analytics Course.

Bellabeat, a high-tech company that manufactures health-focused smart products wants to analyze the usage of one of their products in order to gain insight into how people are already using their smart devices. Then, using this information, they would like personalized strategies with high-level recommendations for how these trends can inform Bellabeat marketing strategy.

Asking

- **Business Task**: Analyze consumption patterns in smart devices and explore how to use these trends to drive demand for Bellabeat products through personalized strategies aligned with consumer preferences.
- **Key stakeholders**: In this case the principal interested of the results of this analysis are going to be Urška Sršen, Bellabeat's co-founder and Chief Creative Officer; Sando Mur, Mathematician and Bellabeat's co founder; And the rest of the Bellabeat marketing analytics team.

Preparing the data

- **Source of the data and credibility**: The data set that it is going to be used in this analysis was generated by respondents to a distributed survey via Amazon Mechanical Turk. It was done by Thirty eligible Fit bit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring.
- **Tools for the project**: For this analysis we are going to be using Rstudio Cloud for storing, cleaning, processing and finally analyzing. In addition, the presented document is being elaborated in R Markdown.
- **Data structure and organization**: The data set consists of 18 csv files in total, some of them are marked as narrow format and others are marked as wide but not all of them are marked.
  - **Important things to mention on data**: All of the files have "merged" on the name which can be redundant and changed for easier naming.
- **Selecting files for analysis**: In this analysis we are going to give a general focus on the data to obtain a better focus on the overall performance of the devices. I will use the daily activity data which includes the most amount of data, then weight, sleep and heart rate data.

Cleaning data

```r
#Installing packages and loading libraries

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(dplyr)
library(ggplot2)
```

```r
#Importing the csv files as R data frames

daily_activity <- read.csv("~/9775499/mturkfitbit_export_4.12.16-5.12.16/Fitabase Data 4.12.16-5.12.16/
weight <- read.csv("~/9775499/mturkfitbit_export_4.12.16-5.12.16/Fitabase Data 4.12.16-5.12.16/weightLog
sleep <- read.csv("~/9775499/mturkfitbit_export_4.12.16-5.12.16/Fitabase Data 4.12.16-5.12.16/sleepDay_m
heartrate <- read.csv("~/9775499/mturkfitbit_export_4.12.16-5.12.16/Fitabase Data 4.12.16-5.12.16/heartr
```

```r
#Preview the data frames:

head(daily_activity)
```

```
##           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
```

```
head(weight)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

```
head(sleep)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```
head(heartrate)
```

```
##           Id                  Time Value
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM    95
```

With the preview I notice that the date in all 4 of the data frames has a different name which makes it more confusing and also it is in a different format since it is in "character" and it should be in "date".

```
#Converting data type:

daily_activity$ActivityDate <- as.Date(daily_activity$ActivityDate, "%m/%d/%y")
weight$Date <- as.Date(weight$Date, "%m/%d/%y")
sleep$SleepDay <- as.Date(sleep$SleepDay, "%m/%d/%y")
heartrate$Time <- as.Date(heartrate$Time, "%m/%d/%y")
```

```
#Renaming the columns of date:

daily_activity <- daily_activity %>%
  rename(Date = ActivityDate)
sleep <- sleep %>%
  rename(Date = SleepDay)
heartrate <- heartrate %>%
  rename(Date = Time)
```

```
#Checking for NA values on our data frames:

print("NA values in daily_activity dataframe:")
```

```
## [1] "NA values in daily_activity dataframe:"
```

```
sum(is.na(daily_activity))
```

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

```
print("NA values in weight dataframe:")
```

```
## [1] "NA values in weight dataframe:"
```

```
sum(is.na(weight))
```

```
## [1] 65
```

```
print("NA values in sleep dataframe:")
```

```
## [1] "NA values in sleep dataframe:"
```

```
sum(is.na(sleep))
```

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

```
print("NA values in heartrate dataframe:")
```

```
## [1] "NA values in heartrate dataframe:"
```

```r
sum(is.na(heartrate))
```

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

```r
#Checking which values are empty in weight data frame:

(is.na(weight))
```

```
##          Id  Date WeightKg WeightPounds   Fat   BMI IsManualReport LogId
##  [1,] FALSE FALSE    FALSE        FALSE FALSE FALSE          FALSE FALSE
##  [2,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [3,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [4,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [5,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [6,] FALSE FALSE    FALSE        FALSE FALSE FALSE          FALSE FALSE
##  [7,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [8,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [9,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [10,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [11,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [12,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [13,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [14,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [15,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [16,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [17,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [18,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [19,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [20,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [21,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [22,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [23,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [24,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [25,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [26,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [27,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [28,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [29,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [30,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [31,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [32,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [33,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [34,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [35,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [36,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [37,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [38,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [39,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [40,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [41,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [42,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [43,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [44,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
```

```
## [45,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [46,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [47,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [48,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [49,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [50,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [51,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [52,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [53,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [54,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [55,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [56,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [57,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [58,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [59,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [60,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [61,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [62,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [63,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [64,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [65,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [66,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [67,] FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
```

We can see that the data frame weight has 65 NA values in Fat column, while total amount of values in this column is 67.

With 97% of empty values, Fat column can not be used for analysis.

```
# Checking for the results of cleaning:

head(daily_activity)
```

```
##           Id       Date TotalSteps TotalDistance TrackerDistance
## 1 1503960366 2020-04-12      13162          8.50            8.50
## 2 1503960366 2020-04-13      10735          6.97            6.97
## 3 1503960366 2020-04-14      10460          6.74            6.74
## 4 1503960366 2020-04-15       9762          6.28            6.28
## 5 1503960366 2020-04-16      12669          8.16            8.16
## 6 1503960366 2020-04-17       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
```

```
head(weight)
```

```
##           Id       Date WeightKg WeightPounds Fat   BMI IsManualReport
## 1 1503960366 2020-05-02     52.6     115.9631  22 22.65           True
## 2 1503960366 2020-05-03     52.6     115.9631  NA 22.65           True
## 3 1927972279 2020-04-13    133.5     294.3171  NA 47.54          False
## 4 2873212765 2020-04-21     56.7     125.0021  NA 21.45           True
## 5 2873212765 2020-05-12     57.3     126.3249  NA 21.69           True
## 6 4319703577 2020-04-17     72.4     159.6147  25 27.45           True
##          LogId
## 1 1.462234e+12
## 2 1.462320e+12
## 3 1.460510e+12
## 4 1.461283e+12
## 5 1.463098e+12
## 6 1.460938e+12
```

```
head(sleep)
```

```
##           Id       Date TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1 1503960366 2020-04-12                 1                327            346
## 2 1503960366 2020-04-13                 2                384            407
## 3 1503960366 2020-04-15                 1                412            442
## 4 1503960366 2020-04-16                 2                340            367
## 5 1503960366 2020-04-17                 1                700            712
## 6 1503960366 2020-04-19                 1                304            320
```

```
head(heartrate)
```

```
##           Id       Date Value
## 1 2022484408 2020-04-12    97
## 2 2022484408 2020-04-12   102
## 3 2022484408 2020-04-12   105
## 4 2022484408 2020-04-12   103
## 5 2022484408 2020-04-12   101
## 6 2022484408 2020-04-12    95
```

Analysing the data

Checking for individual users that provided their data for every data set.

```
#Checking for individual users on every data set:

n_distinct(daily_activity$Id)
```

```
## [1] 33
```

```
n_distinct(weight$Id)
```

```
## [1] 8
```

```
n_distinct(sleep$Id)
```

```
## [1] 24
```

```
n_distinct(heartrate$Id)
```

```
## [1] 14
```

Unfortunately only 8 users have provided their weight data, which is not enough for getting value from analysis.

Next, I will get the summary statistics from the data frames.

```
print("Summary statistics for daily_activity data frame:")
```

```
## [1] "Summary statistics for daily_activity data frame:"
```

```
summary(daily_activity)
```

```
##        Id                 Date              TotalSteps    TotalDistance
##  Min.   :1.504e+09   Min.   :2020-04-12   Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   1st Qu.:2020-04-19   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Median :2020-04-26   Median : 7406   Median : 5.245
##  Mean   :4.855e+09   Mean   :2020-04-26   Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09   3rd Qu.:2020-05-04   3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09   Max.   :2020-05-12   Max.   :36019   Max.   :28.030
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.245   Median :0.0000           Median : 0.210
##  Mean   : 5.475   Mean   :0.1082           Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421           Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000
##  Median :0.2400           Median : 3.365      Median :0.000000
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000
```

```
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

```
print("Summary statistics for weight data frame:")
```

```
## [1] "Summary statistics for weight data frame:"
```

```
summary(weight)
```

```
##        Id                 Date               WeightKg       WeightPounds
##  Min.   :1.504e+09   Min.   :2020-04-12   Min.   : 52.60   Min.   :116.0
##  1st Qu.:6.962e+09   1st Qu.:2020-04-19   1st Qu.: 61.40   1st Qu.:135.4
##  Median :6.962e+09   Median :2020-04-27   Median : 62.50   Median :137.8
##  Mean   :7.009e+09   Mean   :2020-04-26   Mean   : 72.04   Mean   :158.8
##  3rd Qu.:8.878e+09   3rd Qu.:2020-05-04   3rd Qu.: 85.05   3rd Qu.:187.5
##  Max.   :8.878e+09   Max.   :2020-05-12   Max.   :133.50   Max.   :294.3
##
##       Fat              BMI        IsManualReport         LogId
##  Min.   :22.00   Min.   :21.45   Length:67          Min.   :1.460e+12
##  1st Qu.:22.75   1st Qu.:23.96   Class :character   1st Qu.:1.461e+12
##  Median :23.50   Median :24.39   Mode  :character   Median :1.462e+12
##  Mean   :23.50   Mean   :25.19                      Mean   :1.462e+12
##  3rd Qu.:24.25   3rd Qu.:25.56                      3rd Qu.:1.462e+12
##  Max.   :25.00   Max.   :47.54                      Max.   :1.463e+12
##  NA's   :65
```

```
print("Summary statistics for sleep data frame:")
```

```
## [1] "Summary statistics for sleep data frame:"
```

```
summary(sleep)
```

```
##        Id                 Date            TotalSleepRecords TotalMinutesAsleep
##  Min.   :1.504e+09   Min.   :2020-04-12   Min.   :1.000     Min.   : 58.0
##  1st Qu.:3.977e+09   1st Qu.:2020-04-19   1st Qu.:1.000     1st Qu.:361.0
##  Median :4.703e+09   Median :2020-04-27   Median :1.000     Median :433.0
##  Mean   :5.001e+09   Mean   :2020-04-26   Mean   :1.119     Mean   :419.5
##  3rd Qu.:6.962e+09   3rd Qu.:2020-05-04   3rd Qu.:1.000     3rd Qu.:490.0
##  Max.   :8.792e+09   Max.   :2020-05-12   Max.   :3.000     Max.   :796.0
```

```
##  TotalTimeInBed
##  Min.   : 61.0
##  1st Qu.:403.0
##  Median :463.0
##  Mean   :458.6
##  3rd Qu.:526.0
##  Max.   :961.0
```

```
print("Summary statistics for heartrate data frame")
```

```
## [1] "Summary statistics for heartrate data frame"
```

```
summary(heartrate)
```

```
##        Id                 Date                Value
##  Min.   :2.022e+09   Min.   :2020-04-12   Min.   : 36.00
##  1st Qu.:4.388e+09   1st Qu.:2020-04-19   1st Qu.: 63.00
##  Median :5.554e+09   Median :2020-04-26   Median : 73.00
##  Mean   :5.514e+09   Mean   :2020-04-26   Mean   : 77.33
##  3rd Qu.:6.962e+09   3rd Qu.:2020-05-04   3rd Qu.: 88.00
##  Max.   :8.878e+09   Max.   :2020-05-12   Max.   :203.00
```

Discoveries of the summarize:

- Fit bit users have a regular heart rate(**75**) according to the British heart foundation(see references)
- Fit bit users take long walks with a slow pace since most of the distance covered(5.3km) is during light activities.
- Fit bit users are considered as normal weight since the average is 25kg/m2 and that fits with the description of the national institute of statistics.
- fit bit users sleep in average 7 hours a day which fits in the normal conditions of an adult.

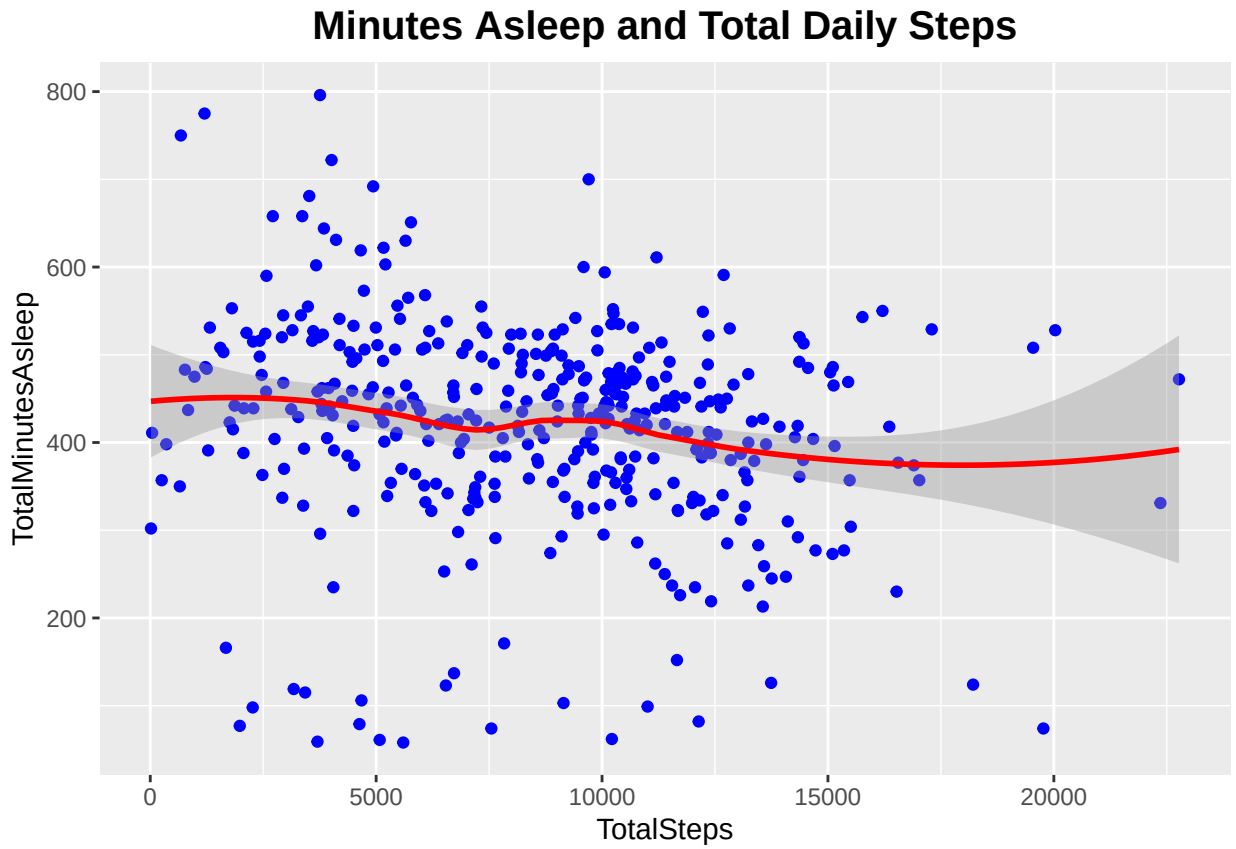
Visualizations

It is time to show the correletation of data in a visual way.

```
daily_activity_and_sleep <- merge(x= daily_activity, y= sleep, by = c("Date","Id"))
```

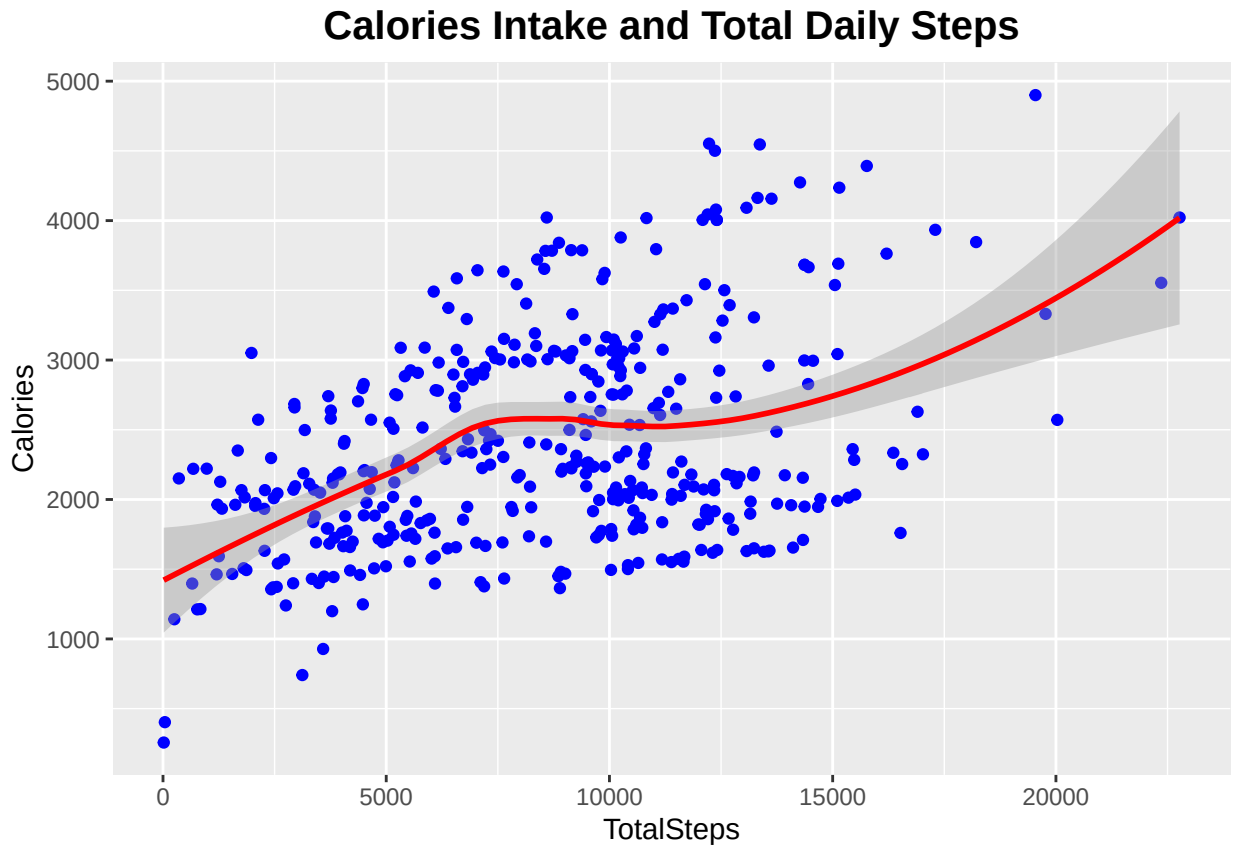
```
ggplot(data = daily_activity_and_sleep) +
     geom_point(mapping = aes(x = TotalSteps, y = TotalMinutesAsleep), color = 'blue') +
     geom_smooth(mapping = aes(x = TotalSteps, y = TotalMinutesAsleep), color = 'red') +
     labs(title="Minutes Asleep and Total Daily Steps") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

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




```
ggplot(data = daily_activity_and_sleep) +
     geom_point(mapping = aes(x = TotalSteps, y = Calories), color = 'blue') +
     geom_smooth(mapping = aes(x = TotalSteps, y = Calories), color = 'red') +
     labs(title="Calories Intake and Total Daily Steps") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

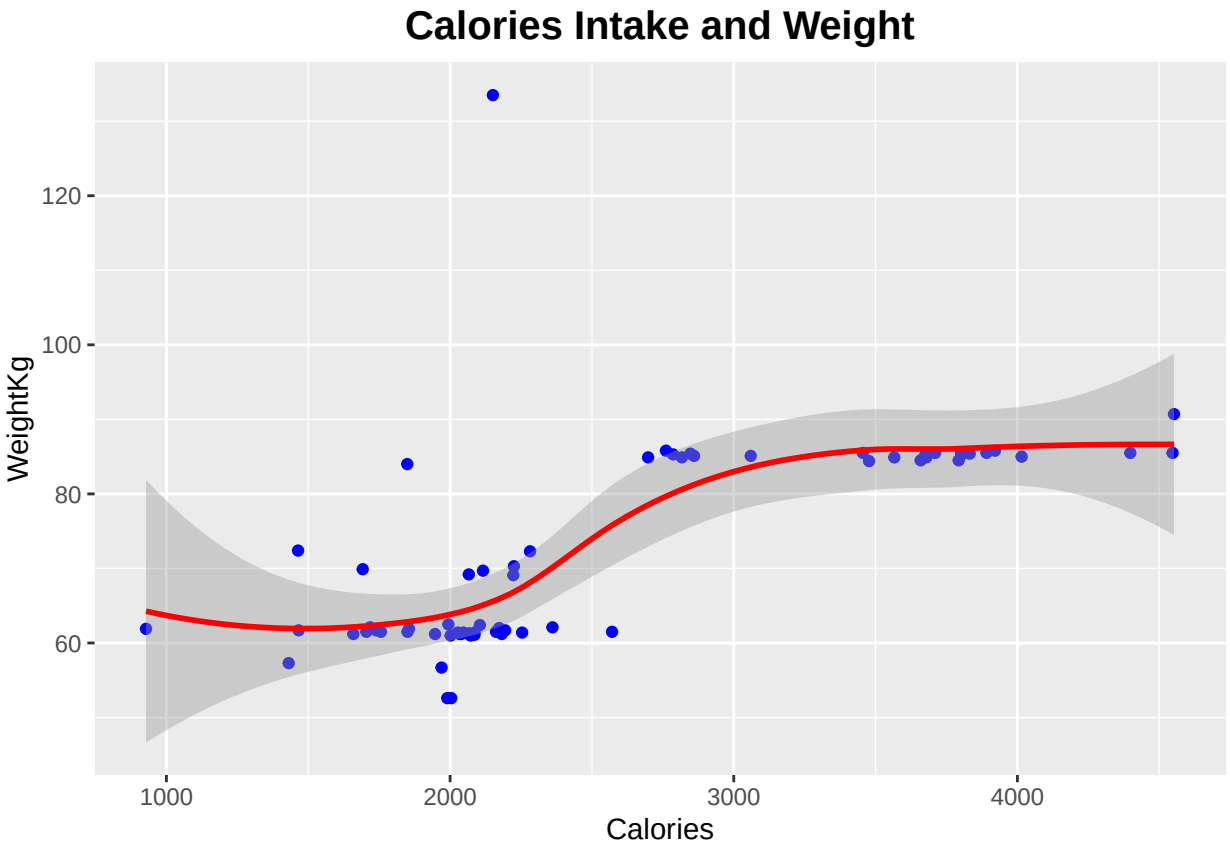
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
daily_activity_and_weight <- merge(x = daily_activity, y = weight, by = c("Date", "Id"))
```

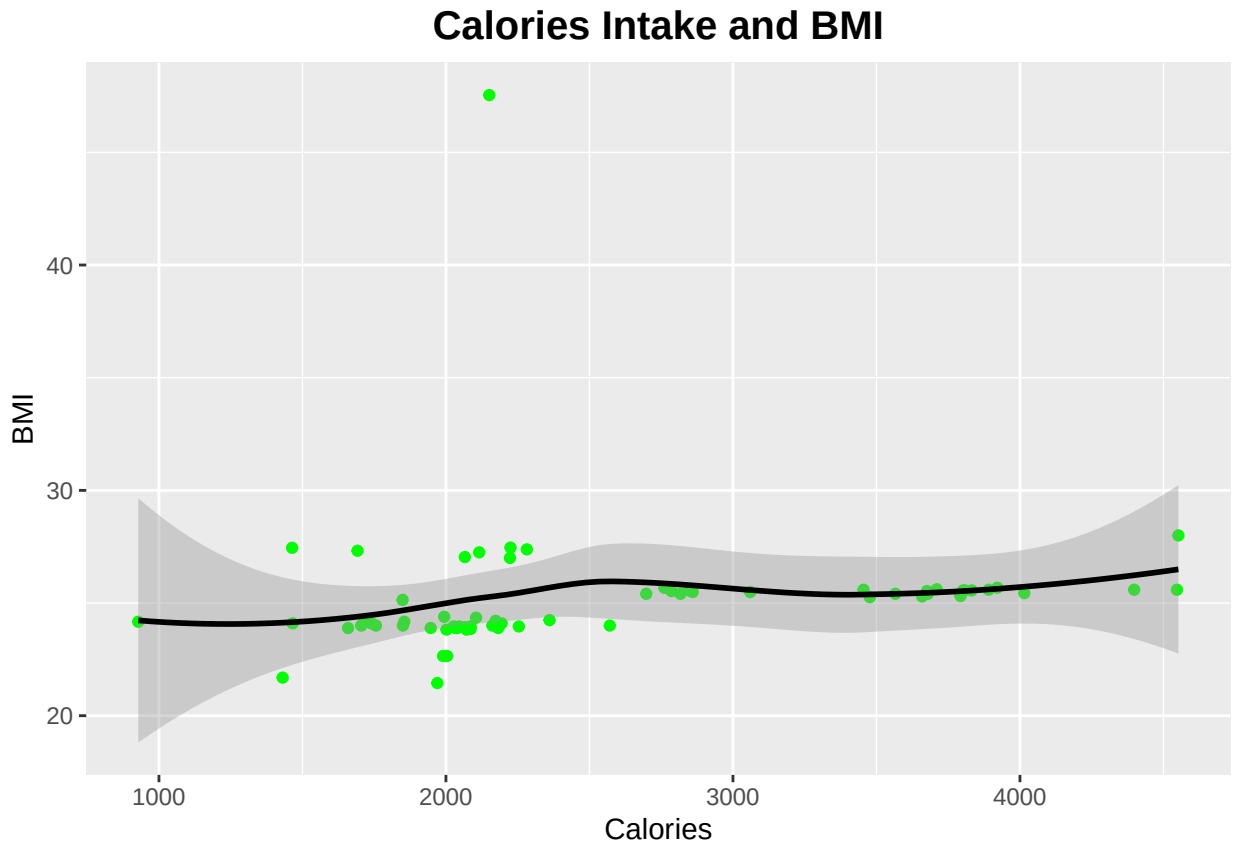
```
ggplot(data = daily_activity_and_weight) +
     geom_point(mapping = aes(x = Calories, y = WeightKg), color = 'blue') +
     geom_smooth(mapping = aes(x = Calories, y = WeightKg), color = 'red') +
     labs(title="Calories Intake and Weight") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

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



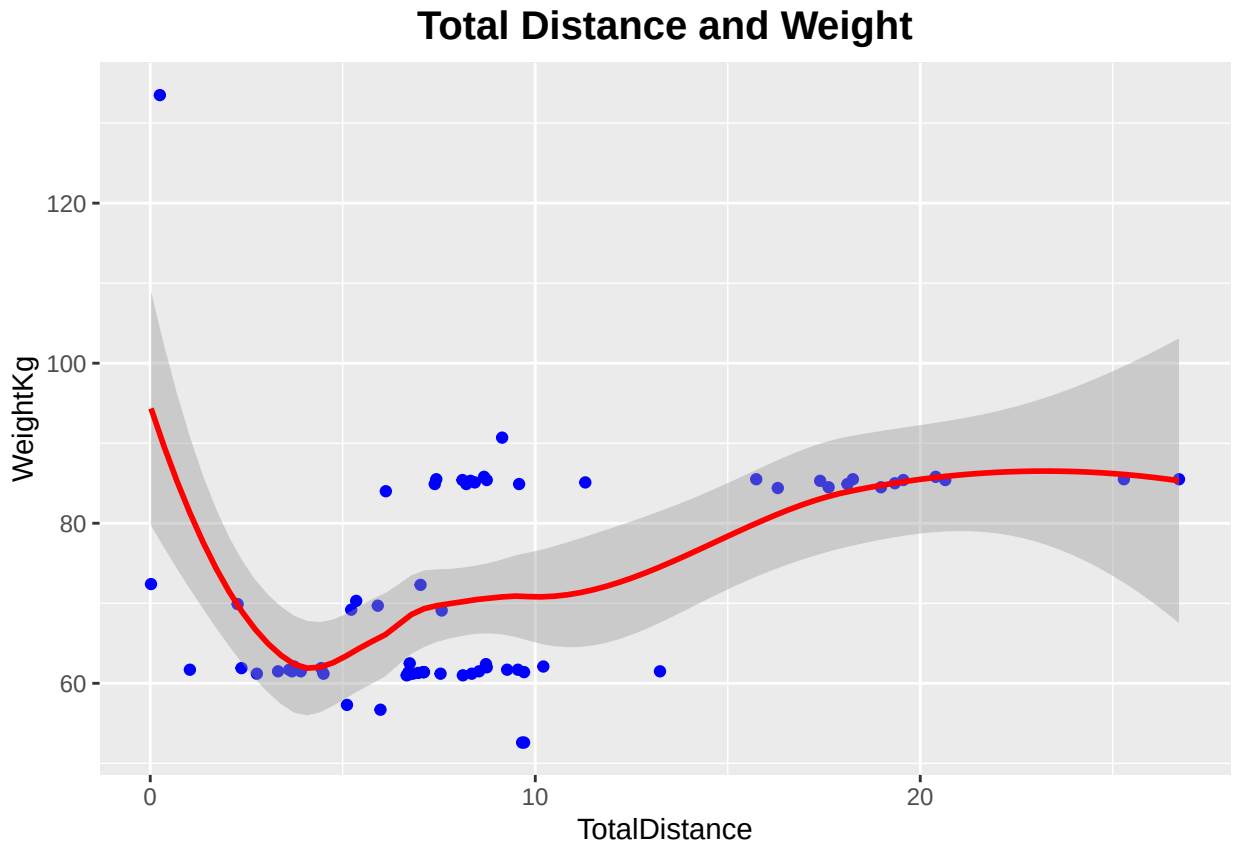

```
ggplot(data = daily_activity_and_weight) +
     geom_point(mapping = aes(x = Calories, y = BMI), color = 'green') +
     geom_smooth(mapping = aes(x = Calories, y = BMI), color = 'black') +
     labs(title="Calories Intake and BMI") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

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



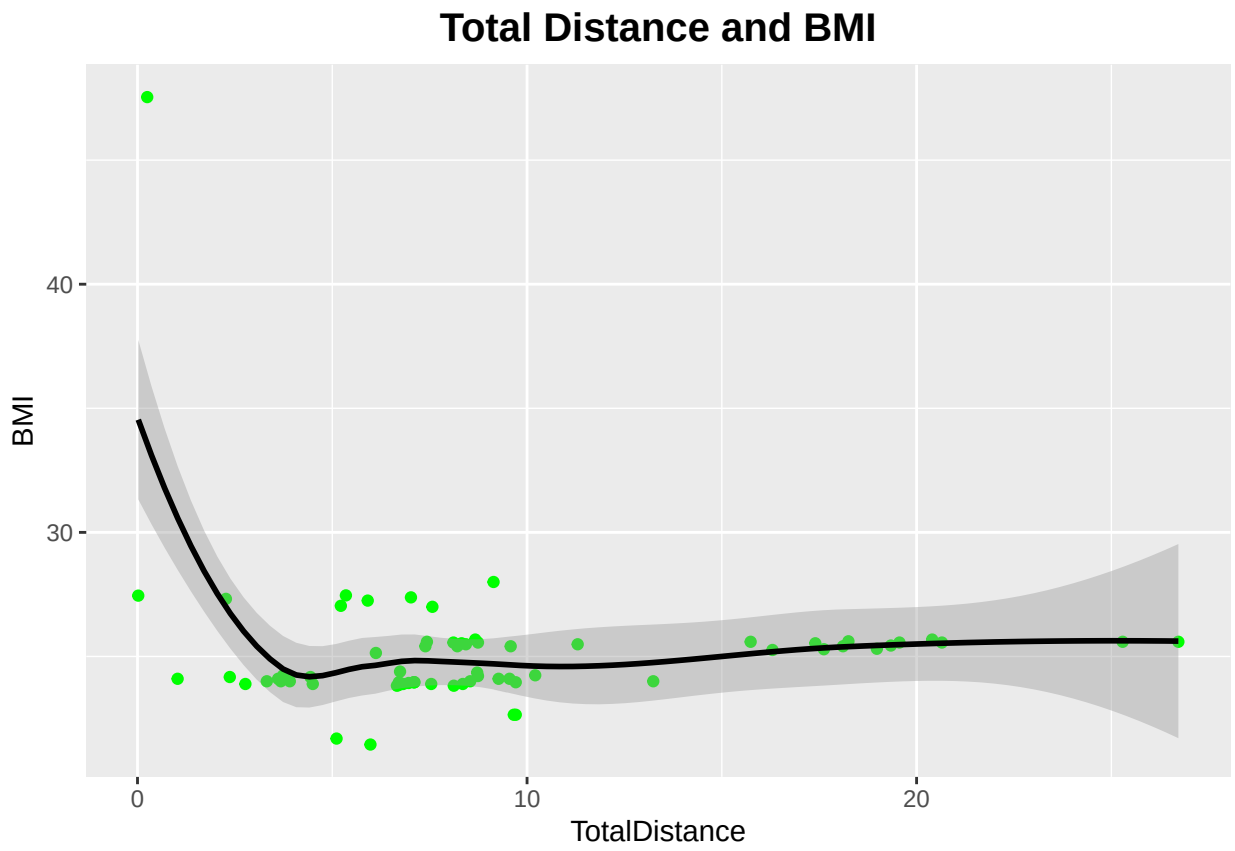

```
ggplot(data = daily_activity_and_weight) +
     geom_point(mapping = aes(x = TotalDistance, y = WeightKg), color = 'blue') +
     geom_smooth(mapping = aes(x = TotalDistance, y = WeightKg), color = 'red') +
     labs(title="Total Distance and Weight") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

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



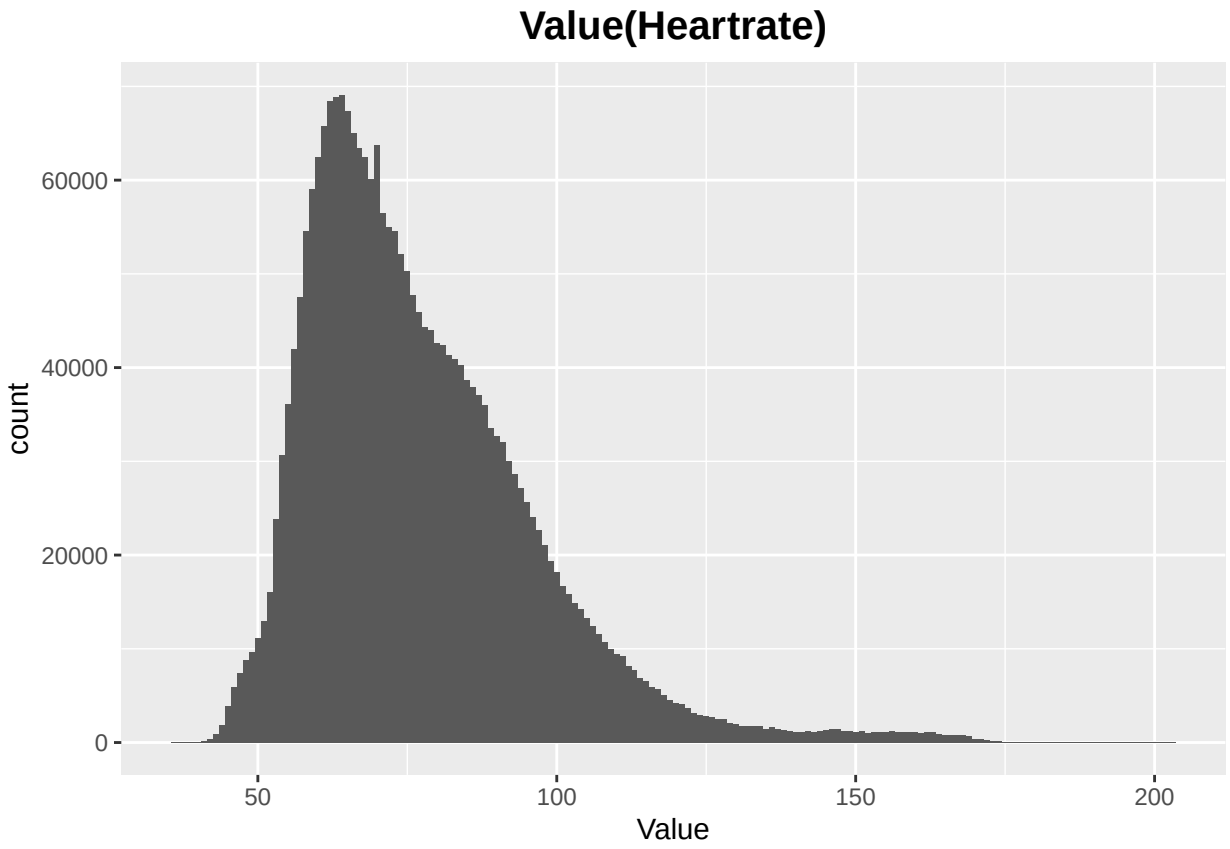

```
ggplot(data = daily_activity_and_weight) +
     geom_point(mapping = aes(x = TotalDistance, y = BMI), color = 'green') +
     geom_smooth(mapping = aes(x = TotalDistance, y = BMI), color = 'black') +
     labs(title="Total Distance and BMI") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

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



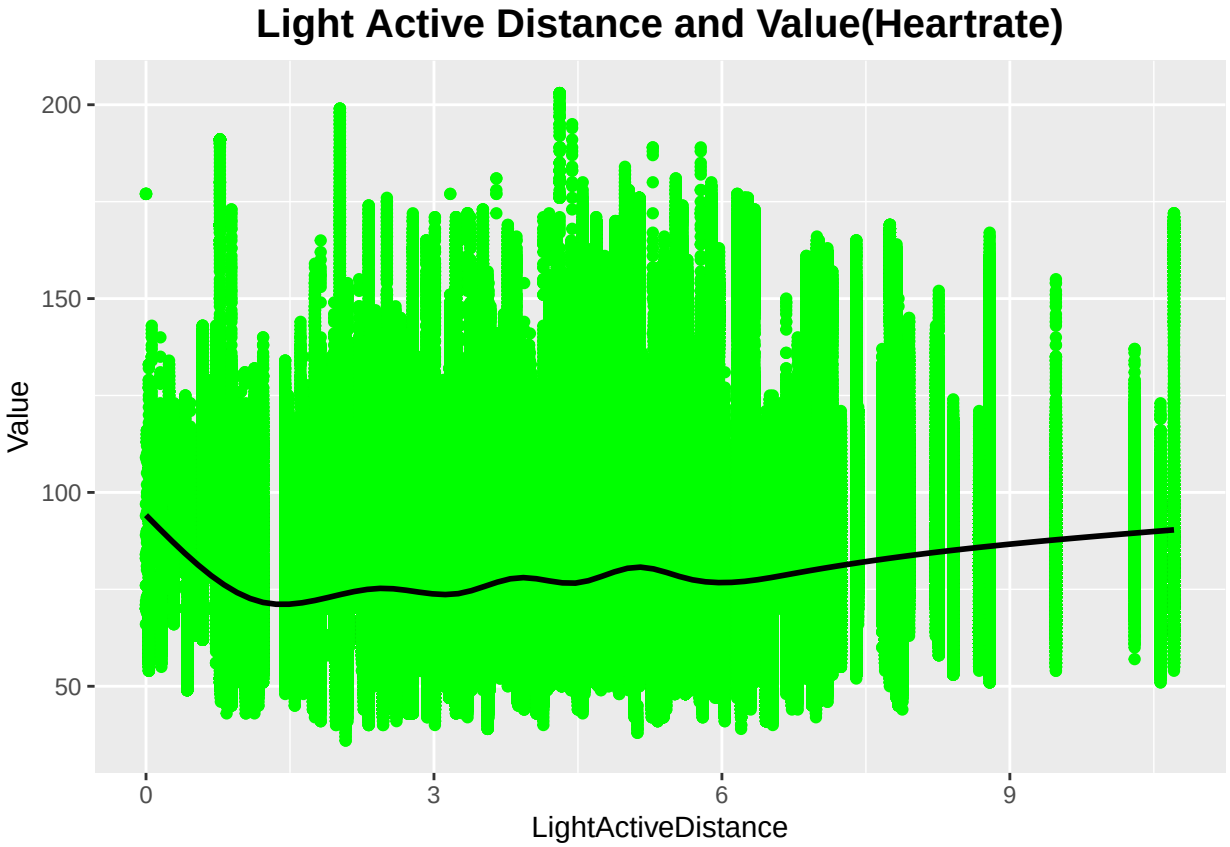

```
daily_activity_and_heartrate <- merge(x = daily_activity, y = heartrate, by = c("Date","Id"))
```

```
ggplot(data = daily_activity_and_heartrate) +
     geom_bar(mapping = aes(x = Value)) +
    labs(title="Value(Heartrate)") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```




```
ggplot(data = daily_activity_and_heartrate) +
     geom_point(mapping = aes(x = LightActiveDistance, y = Value), color = 'green') +
     geom_smooth(mapping = aes(x = LightActiveDistance, y = Value), color = 'black') +
     labs(title="Light Active Distance and Value(Heartrate)") +
     theme(plot.title = element_text(hjust = 0.5, size = 15, face = "bold"))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Conclusions and Recommendations

After doing an in depth research of the data and crossing different types of data between each other we came to some conclusions.

- The most amount of users of have a light type of exercise with an average heart rate and based on that Bellabeat could offer this type of customer personalized exercises that are not very demanding since it is what the maintain as a regular basis.
- Users that consume more calories are more active which makes them a good target for suggesting healthy eating options in their programs or based on location.
- There is a notable relation between sleep and active time which can relate to less sleep time so a suggestion that Bellabeat could do to their customers is related to keeping healthy sleeping habits.
- Analysis shows that users with higher weight tend to move more, and there is no dependency between BMI and activity. This insight is an important basis for promoting physical activity among everyone; again, women tend to think that they can not go to the gym due to being overweight, or that they will look not beutiful enough in sports clothes. Bellabeat could show models of every body constitution in marketing campaigns to give a nice example for women and especially young girls, who are very sensitive to such things.

References and links

1.- https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/ask-the-experts/pulse-rate#:~:text=A%20normal%20resting%20heart%20rate,vary%20from%20person%20to%20person.