Bellabeat Capstone Project

Felipe Cortés

2022-03-29

HOW CAN BELLABEAT PLAY IT SMART

ASK

1. About Bellabeat.

Bellabeat is the go-to wellness brand for women with an ecosystem of products and services focused on women's health. Through its products, Bellabeat collects data on activity, sleep, stress, and reproductive health of users, which empowers women with knowledge about their own health and habits.They develop wearables and accompanying products that monitor biometric and lifestyle data to help women better understand how their bodies work and make healthier choices. Its products include the Bellabeat app, Leaf (a wellness tracker worn as a bracelet, necklace or clip), Time (a wellness watch) and spring (a water bottle).

2. What is the problem you are trying to solve?

As a data analyst I am trying to determine the best possible course to follow regarding Bellabeat's product marketing strategy based on how users are currently using smart devices and provide high level recommendations for how these trends can unlock potential new growth opportunities.

3. How can your insights drive business decisions?

Identifying how smart devices are being used will allow the Marketing and Design team to strengthen specific Bellabeat's product features.

4. Key Stakeholders

Urška Sršen: Bellabeat's co-founder and Chief Creative Officer. Sandro Mur: Bellabeat's co-founder. Bellabeat's marketing team.

PREPARE

1. About the Dataset

Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016

These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Individual reports can be parsed by export session ID (column A) or timestamp (column B). Variation between output represents use of different types of Fitbit trackers and individual tracking behaviours / preferences.

Furberg, R., Brinton, J., Keating, M., & Ortiz, A. (2016). Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016 [Data set]. Zenodo. https://doi.org/10.5281/zenodo.53894

The data is long as each row is one time point per subject, with each subject having multiple rows.

```
head(daily_activity, 10)
```

Previewing Datasets

```
## # A tibble: 10 x 15
##        Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie~
##     <dbl> <chr>             <dbl>         <dbl>           <dbl>             <dbl>
##  1 1.50e9 4/12/2016         13162          8.5             8.5                 0
##  2 1.50e9 4/13/2016         10735          6.97            6.97                0
##  3 1.50e9 4/14/2016         10460          6.74            6.74                0
##  4 1.50e9 4/15/2016          9762          6.28            6.28                0
##  5 1.50e9 4/16/2016         12669          8.16            8.16                0
##  6 1.50e9 4/17/2016          9705          6.48            6.48                0
##  7 1.50e9 4/18/2016         13019          8.59            8.59                0
##  8 1.50e9 4/19/2016         15506          9.88            9.88                0
##  9 1.50e9 4/20/2016         10544          6.68            6.68                0
## 10 1.50e9 4/21/2016          9819          6.34            6.34                0
## # ... with 9 more variables: VeryActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>
```

```
head(daily_calories, 10)
```

```
## # A tibble: 10 x 3
##            Id ActivityDay Calories
##         <dbl> <chr>          <dbl>
##  1 1503960366 4/12/2016       1985
##  2 1503960366 4/13/2016       1797
##  3 1503960366 4/14/2016       1776
##  4 1503960366 4/15/2016       1745
##  5 1503960366 4/16/2016       1863
##  6 1503960366 4/17/2016       1728
##  7 1503960366 4/18/2016       1921
##  8 1503960366 4/19/2016       2035
##  9 1503960366 4/20/2016       1786
## 10 1503960366 4/21/2016       1775
```

```
head(daily_steps, 10)
```

```
## # A tibble: 10 x 3
##            Id ActivityDay StepTotal
##         <dbl> <chr>           <dbl>
##  1 1503960366 4/12/2016       13162
##  2 1503960366 4/13/2016       10735
##  3 1503960366 4/14/2016       10460
##  4 1503960366 4/15/2016        9762
##  5 1503960366 4/16/2016       12669
##  6 1503960366 4/17/2016        9705
##  7 1503960366 4/18/2016       13019
##  8 1503960366 4/19/2016       15506
##  9 1503960366 4/20/2016       10544
## 10 1503960366 4/21/2016        9819
```

```
head(heart_rate_seconds, 10)
```

```
## # A tibble: 10 x 3
##            Id Time                  Value
##         <dbl> <chr>                 <dbl>
##  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
##  7 2022484408 4/12/2016 7:22:10 AM    91
##  8 2022484408 4/12/2016 7:22:15 AM    93
##  9 2022484408 4/12/2016 7:22:20 AM    94
## 10 2022484408 4/12/2016 7:22:25 AM    93
```

```
head(daily_sleep, 10)
```

```
## # A tibble: 10 x 5
##            Id SleepDay          TotalSleepRecor~ TotalMinutesAsl~ TotalTimeInBed
##         <dbl> <chr>                        <dbl>            <dbl>          <dbl>
##  1 1503960366 4/12/2016 12:00:~                1              327            346
##  2 1503960366 4/13/2016 12:00:~                2              384            407
##  3 1503960366 4/15/2016 12:00:~                1              412            442
##  4 1503960366 4/16/2016 12:00:~                2              340            367
##  5 1503960366 4/17/2016 12:00:~                1              700            712
##  6 1503960366 4/19/2016 12:00:~                1              304            320
##  7 1503960366 4/20/2016 12:00:~                1              360            377
##  8 1503960366 4/21/2016 12:00:~                1              325            364
##  9 1503960366 4/23/2016 12:00:~                1              361            384
## 10 1503960366 4/24/2016 12:00:~                1              430            449
```

```
head(weight_log_info, 10)
```

```
## # A tibble: 10 x 8
##            Id Date      WeightKg WeightPounds   Fat   BMI IsManualReport   LogId
##         <dbl> <chr>        <dbl>        <dbl> <dbl> <dbl> <lgl>            <dbl>
##  1 1503960366 5/2/2016~     52.6         116.    22  22.6 TRUE           1.46e12
##  2 1503960366 5/3/2016~     52.6         116.    NA  22.6 TRUE           1.46e12
##  3 1927972279 4/13/201~    134.          294.    NA  47.5 FALSE          1.46e12
##  4 2873212765 4/21/201~     56.7         125.    NA  21.5 TRUE           1.46e12
##  5 2873212765 5/12/201~     57.3         126.    NA  21.7 TRUE           1.46e12
##  6 4319703577 4/17/201~     72.4         160.    25  27.5 TRUE           1.46e12
##  7 4319703577 5/4/2016~     72.3         159.    NA  27.4 TRUE           1.46e12
##  8 4558609924 4/18/201~     69.7         154.    NA  27.2 TRUE           1.46e12
##  9 4558609924 4/25/201~     70.3         155.    NA  27.5 TRUE           1.46e12
## 10 4558609924 5/1/2016~     69.9         154.    NA  27.3 TRUE           1.46e12
```

PROCESS

1. First we take a look at the sample size.

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

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

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

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

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

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

We notice that all numbers differ from the description of the dataset, being 30 participants, so we must doubt of the integrity and credibility of the dataset.

Removing Duplicates

2. Then we check for duplicates.

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(daily_calories))
```

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

```
sum(duplicated(daily_steps))
```

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

```
sum(duplicated(heart_rate_seconds))
```

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

```
sum(duplicated(daily_sleep))
```

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

```
sum(duplicated(weight_log_info))
```

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

3. Now we remove the duplicates in the sleep_day dataset.

```
daily_sleep <- daily_sleep %>%
  distinct() %>%
  drop_na()
```

4. We check again for duplicates.

```
sum(duplicated(daily_sleep))
```

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

Correcting Date Format

5. Now we start by splitting up date and time and naming all date columns as 'Date' to work better with our data sets.

```
daily_sleep <- daily_sleep %>%
  separate(SleepDay, c("Date", "Time"), " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 410 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
heart_rate_seconds <- heart_rate_seconds %>%
  separate(Time, c("Date", "Time"), " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 2483658 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
weight_log_info <- weight_log_info %>%
    separate(Date, c("Date", "Time"), " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 67 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

6. And we change all remaining date column names to 'Date'.

```
# Changing all date columns to 'Date'
daily_activity <- rename(daily_activity, Date = ActivityDate)

daily_calories <- rename(daily_calories, Date = ActivityDay)

daily_steps <- rename(daily_steps, Date = ActivityDay)
```

7. Next we noticed that the date is formatted as characters and not as dates so we changed the format.

```
daily_activity$Date <- mdy(daily_activity$Date)
daily_calories$Date <- mdy(daily_calories$Date)
daily_sleep$Date <- mdy(daily_sleep$Date)
daily_steps$Date <- mdy(daily_steps$Date)
heart_rate_seconds$Date <- mdy(heart_rate_seconds$Date)
weight_log_info$Date <- mdy(weight_log_info$Date)
```

Checking Credibility of the Data

8. Now we check if there are inconsistencies in the number of rows.

```
nrow(daily_activity)
```

```
## [1] 940
```

```
nrow(daily_calories)
```

```
## [1] 940
```

```
nrow(daily_sleep)
```

```
## [1] 410
```

```
nrow(daily_steps)
```

```
## [1] 940
```

```
nrow(heart_rate_seconds)
```

```
## [1] 2483658
```

```
nrow(weight_log_info)
```

```
## [1] 67
```

9. We proceed to check for missing values in all of the tables.

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

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

```
sum(is.na(daily_calories))
```

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

```
sum(is.na(daily_sleep))
```

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

```
sum(is.na(daily_steps))
```

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

```
sum(is.na(heart_rate_seconds))
```

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

```
sum(is.na(weight_log_info))
```

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

We get 65 missing values in the weight_log_info table. We check the missing values and discover that the missing values are in the 'Fat' column.

```
is.na(weight_log_info)
```

Transforming Data

```
##          Id  Date  Time WeightKg WeightPounds   Fat   BMI IsManualReport LogId
##  [1,] FALSE FALSE FALSE    FALSE        FALSE FALSE FALSE          FALSE FALSE
##  [2,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [3,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [4,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [5,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [6,] FALSE FALSE FALSE    FALSE        FALSE FALSE FALSE          FALSE FALSE
##  [7,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [8,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
##  [9,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [10,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [11,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [12,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [13,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [14,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [15,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [16,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [17,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [18,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [19,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [20,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [21,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [22,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [23,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [24,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [25,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [26,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [27,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [28,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [29,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [30,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [31,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
```

```
## [32,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [33,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [34,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [35,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [36,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [37,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [38,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [39,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [40,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [41,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [42,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [43,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [44,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [45,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [46,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [47,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [48,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [49,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [50,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [51,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [52,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [53,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [54,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [55,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [56,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [57,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [58,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [59,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [60,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [61,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [62,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [63,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [64,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [65,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [66,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
## [67,] FALSE FALSE FALSE    FALSE        FALSE  TRUE FALSE          FALSE FALSE
```

10. We decide to exclude the 'Fat' column from the analysis.

```
weight_log_info <- subset(weight_log_info, select = -Fat)
colnames(weight_log_info)
```

```
## [1] "Id"             "Date"           "Time"           "WeightKg"
## [5] "WeightPounds"   "BMI"            "IsManualReport" "LogId"
```

11. Now we join the tables with the same amount of observations (daily_activity, daily_calories, daily_steps) in order to have a single table to make future analysis easier. To merge daily_steps correctly we renamed StepTotal column to TotalSteps.

```
daily_steps <- rename(daily_steps, TotalSteps = StepTotal)
daily_overall <- merge(daily_activity, daily_calories,by = c("Id", "Date", "Calories"))
daily_overall <- merge(daily_overall, daily_steps,by = c("Id", "Date", "TotalSteps"))
head(daily_overall)
```

```
##           Id       Date TotalSteps Calories TotalDistance TrackerDistance
## 1 1503960366 2016-04-12      13162     1985          8.50            8.50
```

```
## 2 1503960366 2016-04-13      10735     1797          6.97            6.97
## 3 1503960366 2016-04-14      10460     1776          6.74            6.74
## 4 1503960366 2016-04-15       9762     1745          6.28            6.28
## 5 1503960366 2016-04-16      12669     1863          8.16            8.16
## 6 1503960366 2016-04-17       9705     1728          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
## 1                  13                 328              728
## 2                  19                 217              776
## 3                  11                 181             1218
## 4                  34                 209              726
## 5                  10                 221              773
## 6                  20                 164              539
```

ANALYSE

1. First of all, we evaluated some general information about our newly created and cleaned dataset in order to identify some highlights and starting points.

```
summary(daily_overall)
```

```
##        Id                 Date              TotalSteps       Calories
##  Min.   :1.504e+09   Min.   :2016-04-12   Min.   :    0   Min.   :   0
##  1st Qu.:2.320e+09   1st Qu.:2016-04-19   1st Qu.: 3790   1st Qu.:1828
##  Median :4.445e+09   Median :2016-04-26   Median : 7406   Median :2134
##  Mean   :4.855e+09   Mean   :2016-04-26   Mean   : 7638   Mean   :2304
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04   3rd Qu.:10727   3rd Qu.:2793
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :36019   Max.   :4900
##  TotalDistance    TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.245   Median : 5.245   Median :0.0000           Median : 0.210
##  Mean   : 5.490   Mean   : 5.475   Mean   :0.1082           Mean   : 1.503
##  3rd Qu.: 7.713   3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053
##  Max.   :28.030   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
```

2. The general glimpse of the data led to a series of initial hypotheses which would open our view to deeper analysis.

We then wondered: + How does the correlation look between burnt calories and daily steps? + Is there a correlation between daily steps and sleeping time? + Is there a correlation between sleeping longer and burning more calories? + Is there a correlation between a sedentary lifestyle and sleeping time? + How does the correlation look between weight and burnt calories?

In the following graphics we will evaluate the different hypotheses.

```
ggplot(data= daily_overall, aes(x=TotalSteps, y=Calories)) +
  geom_point(alpha= .5, color="blue") +
  geom_smooth(fill= "green") +
  labs(title= "Correlation Betweeen Steps and Calories", subtitle = "Walking more burns more calories",
```

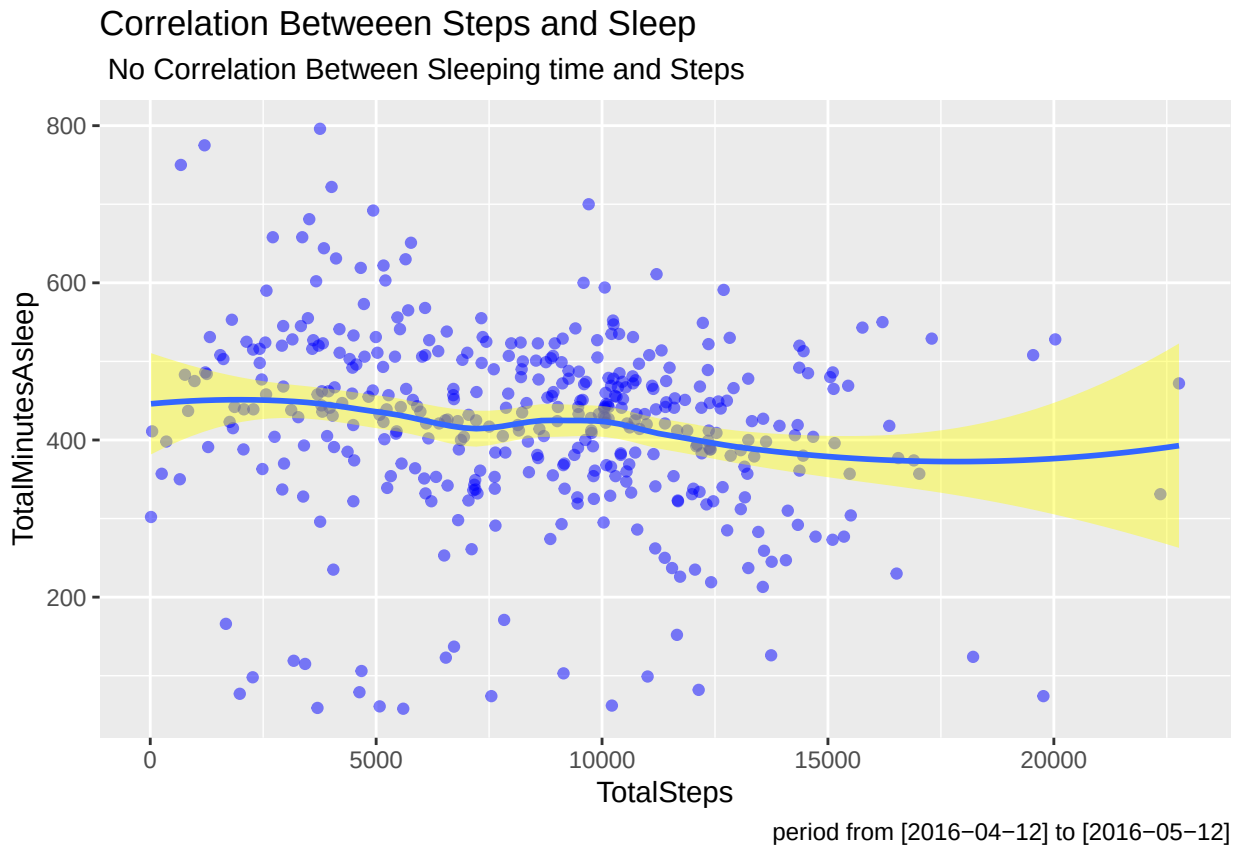
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


```
ggplot(data= activity_vs_sleep, aes(x=TotalSteps, y=TotalMinutesAsleep)) +
  geom_point(alpha= .5, color="blue") +
```

```
  geom_smooth(fill= "yellow") +
  labs(title= "Correlation Betweeen Steps and Sleep", subtitle = " No Correlation Between Sleeping time
```

Correlation Betweeen Steps and Sleep

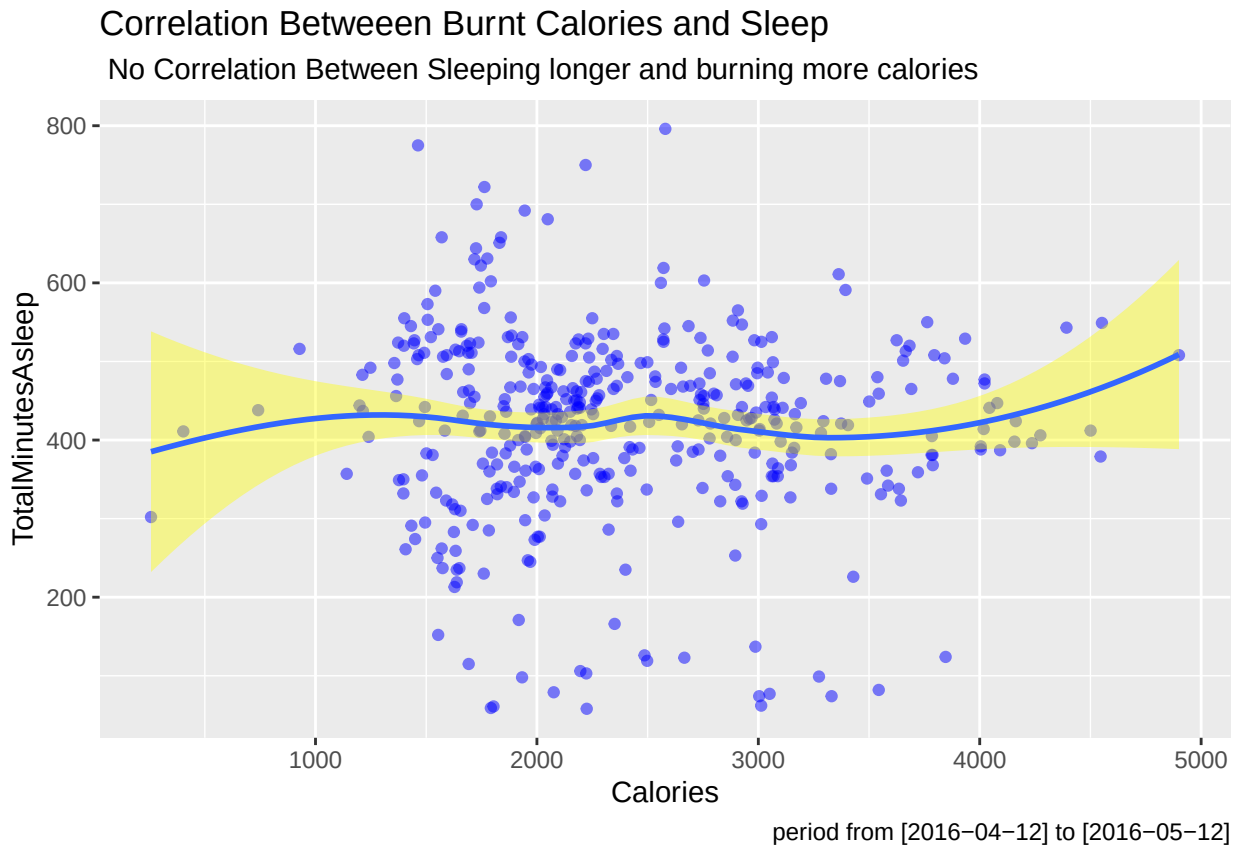
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data= activity_vs_sleep, aes(x=Calories, y=TotalMinutesAsleep)) +
  geom_point(alpha= .5, color="blue") +
  geom_smooth(fill= "yellow") +
  labs(title= "Correlation Betweeen Burnt Calories and Sleep", subtitle = " No Correlation Between Slee
```

Correlation Betweeen Burnt Calories and Sleep

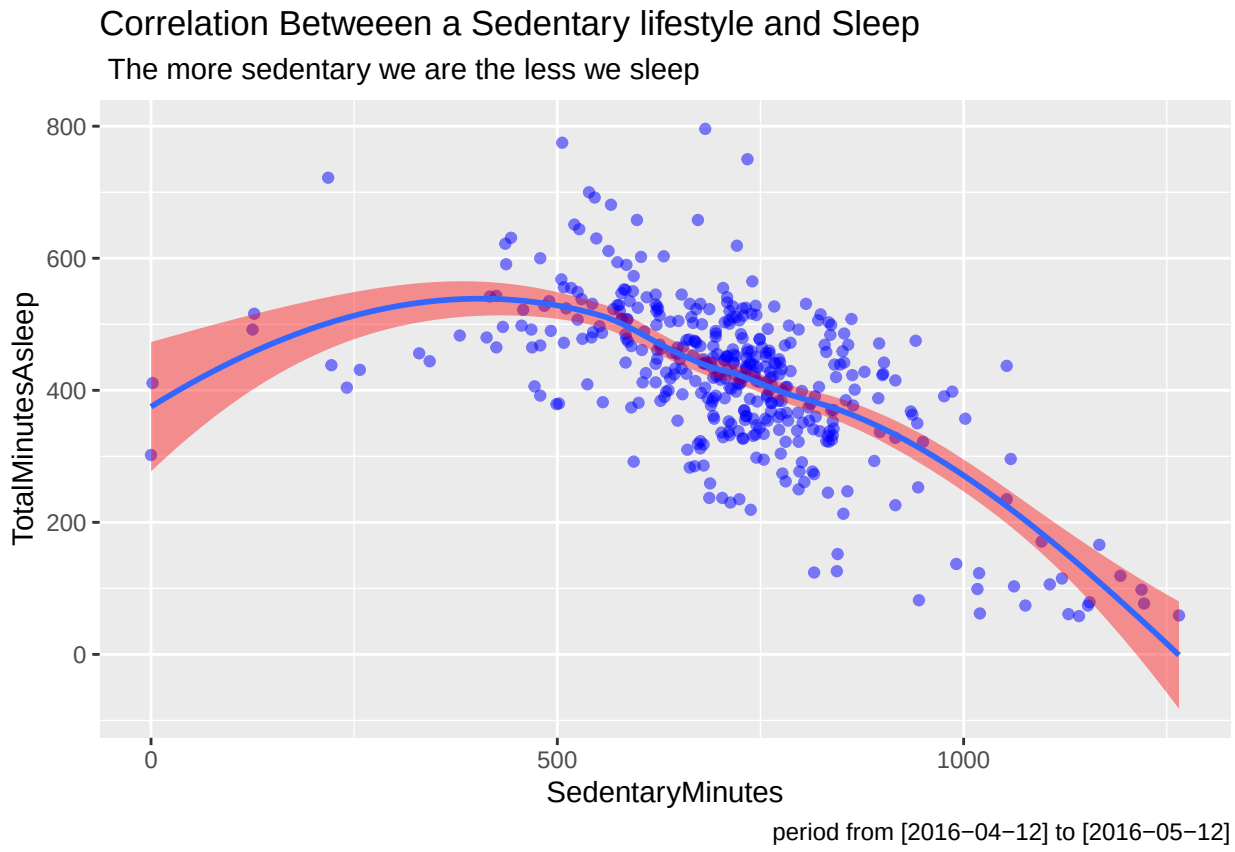
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data= overall_vs_sleep, aes(x=SedentaryMinutes, y=TotalMinutesAsleep)) +
  geom_point(alpha= .5, color="blue") +
  geom_smooth(fill= "red") +
  labs(title= "Correlation Betweeen a Sedentary lifestyle and Sleep", subtitle = " The more sedentary w
```

Correlation Betweeen a Sedentary lifestyle and Sleep

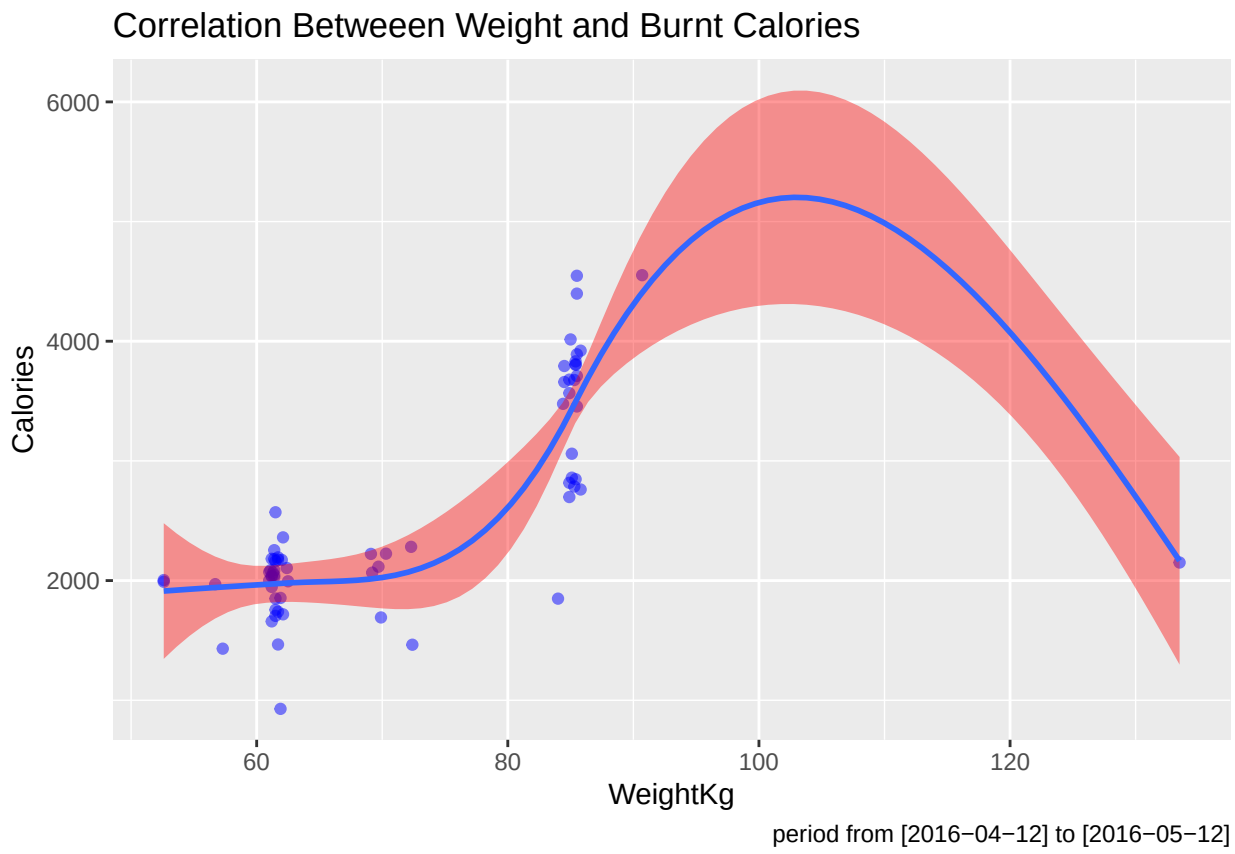
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data= weight_vs_activity, aes(x=WeightKg, y=Calories)) +
  geom_point(alpha= .5, color="blue") +
  geom_smooth(fill= "red") +
  labs(title= "Correlation Betweeen Weight and Burnt Calories", caption = "period from [2016-04-12] to
```

Correlation Betweeen Weight and Burnt Calories

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

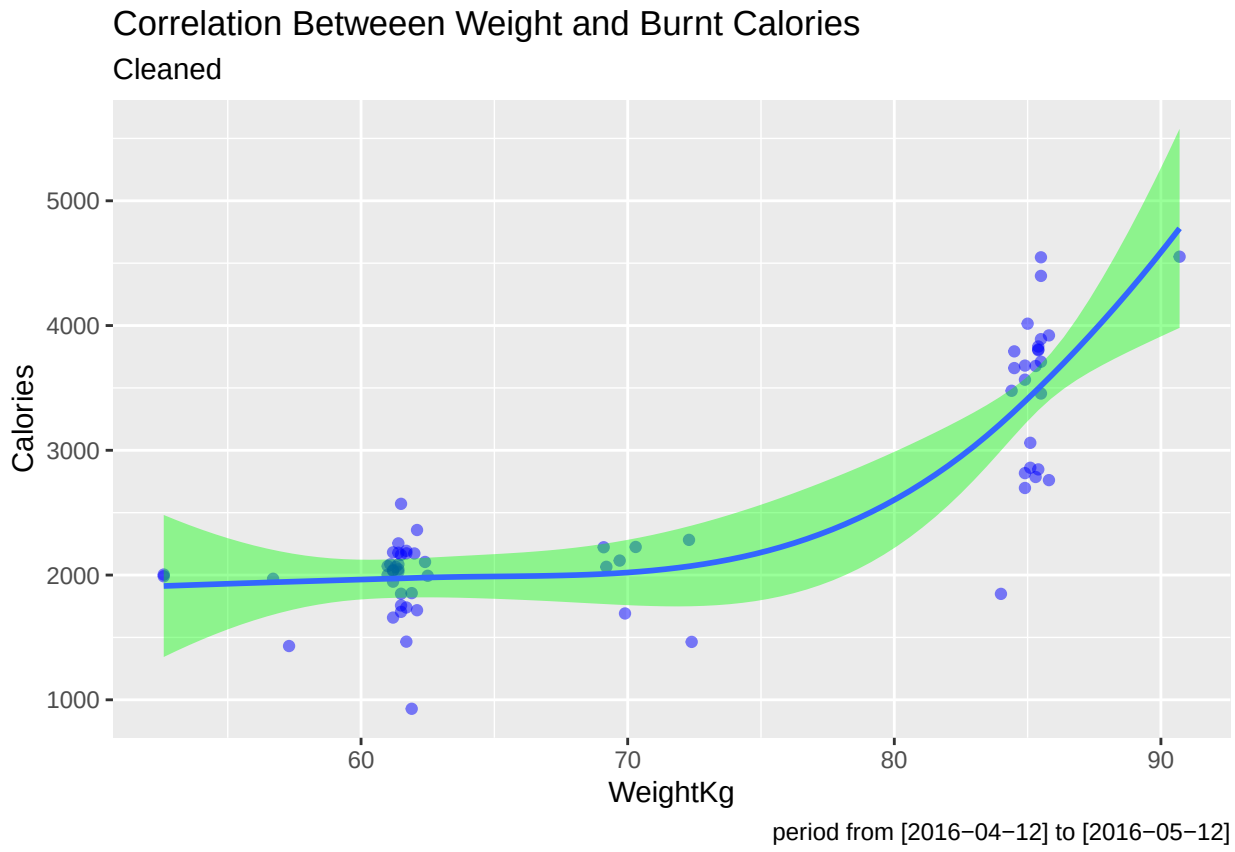


We noticed in this last plot a value that altered completely the visualization. Therefore we decided to eliminate the value and run the plot again. This is the result:

```
weight_vs_activity_cleaned <- weight_vs_activity[!(weight_vs_activity$Id=="1927972279"),]

ggplot(data= weight_vs_activity_cleaned, aes(x=WeightKg, y=Calories)) +
  geom_point(alpha= .5, color="blue") +
  geom_smooth(fill= "green") +
  labs(title= "Correlation Betweeen Weight and Burnt Calories", subtitle = "Cleaned", caption = "period
```

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



The plot shows an alleged correlation between weight and burnt calories. But to draw any conclusions from this data would require more information.

3. One of the main objectives of our analysis is to determine the best possible course to follow regarding Bellabeat's product marketing strategy based on how users are currently using smart devices. Therefore, we decided to analyse the frequency of use of the device.

```
daily_use <- activity_vs_sleep %>%
  group_by(Id) %>%
  summarize(days_used=sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ "low use",
    days_used >= 11 & days_used <= 20 ~ "moderate use",
    days_used >= 21 & days_used <= 31 ~ "high use",
  ))

head(daily_use)
```

```
## # A tibble: 6 x 3
##           Id days_used usage
##        <dbl>     <int> <chr>
## 1 1503960366        25 high use
## 2 1644430081         4 low use
## 3 1844505072         3 low use
## 4 1927972279         5 low use
## 5 2026352035        28 high use
## 6 2320127002         1 low use
```

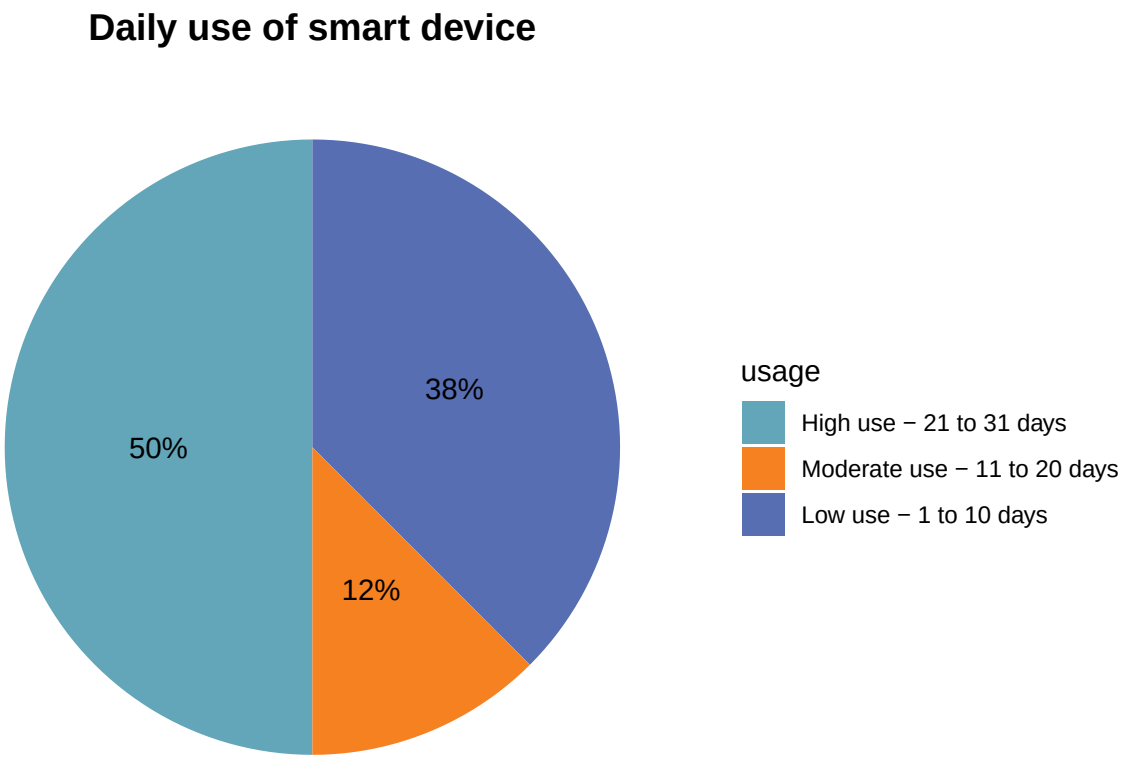
```r
daily_use_percent <- daily_use %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

daily_use_percent$usage <- factor(daily_use_percent$usage, levels = c("high use", "moderate use", "low u

head(daily_use_percent)
```

```
## # A tibble: 3 x 3
##   usage        total_percent labels
##   <fct>                <dbl> <chr>
## 1 high use             0.5   50%
## 2 low use              0.375 38%
## 3 moderate use         0.125 12%
```

```r
daily_use_percent %>%
  ggplot(aes(x="",y=total_percent, fill=usage)) +
  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")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#63a6b9","#f68121", "#576eb2"),
                    labels = c("High use - 21 to 31 days",
                               "Moderate use - 11 to 20 days",
                               "Low use - 1 to 10 days"))+
  labs(title="Daily use of smart device")
```



Conclusions: Daily Use of Smart Devices The analysis demonstrated that a majority of the evaluated sample uses their device more than 20 days. But it also shows. Bellabeat might need to address specifically why is people not using their device everyday or even for sleeping. This allows us to make a couple of recommendations.

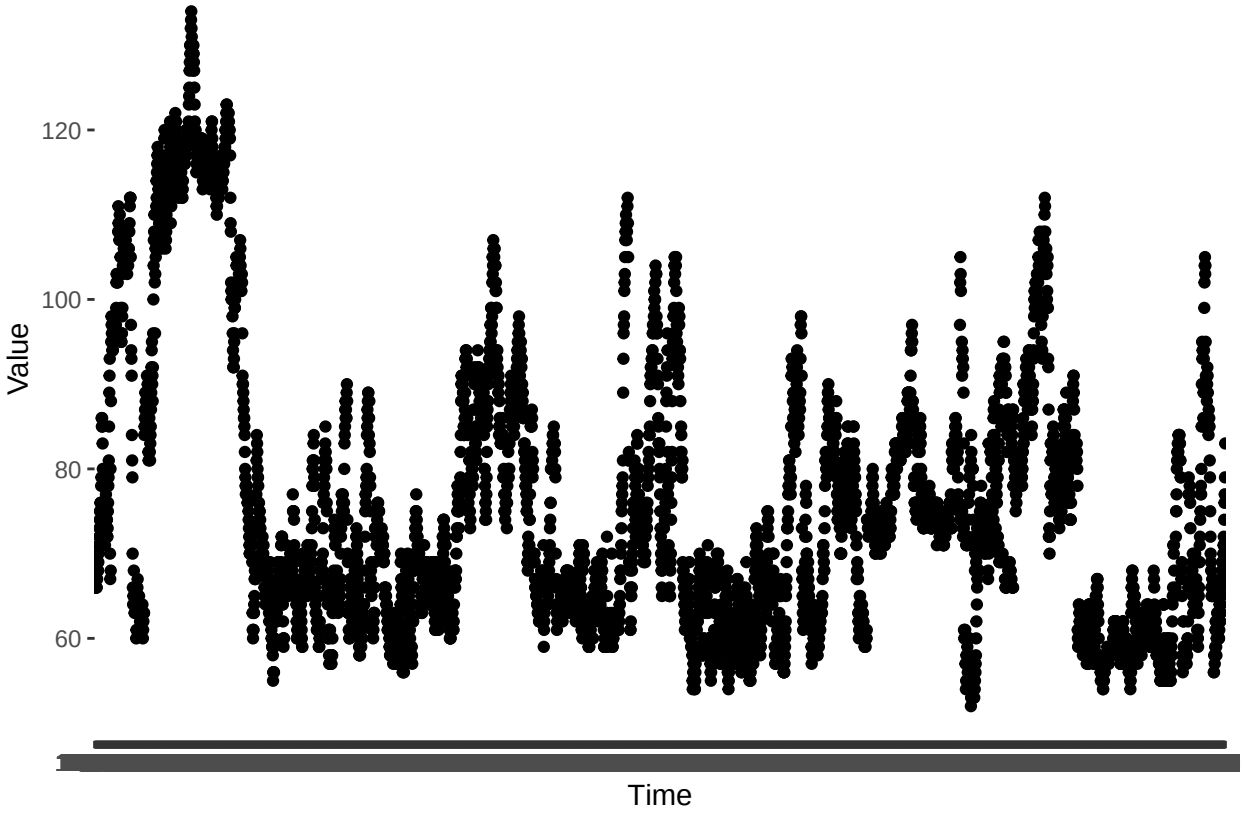
- Run more tests with the UX design department to determine if the cause of the low use may lie on design or discomfort that it might bring.
- Does the device have enough battery to allow the user to keep it with them for a longer time?
- Is the device fashionable enough? Can it be used at any occasion?

Analysing Hear Rate Our next analysis brought us to the heart rate of the users. Due to the sample size, We first chose a random user and display on a graph how heart rate looks throughout the day.

```
sample_heartrate <- heart_rate_seconds %>% filter(Id == 2022484408) %>% select(Id, Date, Time, Value)

sample_heartrate <- sample_heartrate %>% filter(Date == "2016-04-12")

ggplot(data = sample_heartrate, mapping = aes(x = Time, y = Value)) + geom_point()
```



When we saw the irregular variation of the heart rate during the day made us wonder if this hear rate was characteristic of a healthy person. Upon further investigation, we learned that, in fact, healthy people tend to have higher heart rate variability. This is at the same time translated into:

- Greater willpower
- Positive mood
- High resilience
- Overall wellness and healthfulness
- High productivity

On the other hand, having lower heart rate variability scores could lead the body to experience:

- Diminished willpower
- Negative mood
- Low resilience
- Overall poor health
- Low productivity

Source: https://www.kosmotime.com/heart-rate-variability/#How_to_Use_Heart_Rate_Variability_for_Greater_Productivity

Conclusions: Heart Rate Bellabeat could use their devices to track heart rate and alert the users of unusual variations. Whenever a person is experiencing lower heart rate variability scores, the app should trigger an alert and deliver some suggestions based on the historical data of the user. This information could also be helpful to guide the user to calmer states of mind.

As seen in the plot, there is a spike in hear rate. The device could also follow and identify abnormal, recurrent spikes on the heart rate and inform the user of possible risks to allow the person to find the source of the spikes.