# Case study from the google analytics course by Sören Nonnengart

Sören Nonnengart

2022-05-19

## Ask Phase

### About the company

Bellabeat is a high-tech company that manufactures health-focused smart products. One of the founders, Urška Sršen, used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on:

- activity
- sleep
- stress
- reproductive health.

has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women

### Questions for the analysis

#### 1. What are some trends in smart device usage?

According to Statista, the current number of smartphone users worldwide today is 6.648 billion. This means that 83.72 % of the world's population owns a smartphone.The trend thus shows that in the near future almost all people worldwide could own a smartphone. It is estimated that 7.33 billion people could already own a smartphone by 2025. link.

Another interesting study by Seifert & Vandelanotte (2021) shows that 75.0% of older adults used at least one mobile device; 22.9 % of them used health-related apps. Younger individuals and those with a strong interest in new technology had a higher likelihood of using health apps. link According to Statista, it can also be shown that these apps are esspecially used for fitness-tracking link.

#### 2. How could these trends apply to Bellabeat customers?

A: Bellabeat customers are perfectly fit into this trend because the probability is really high that more and more people will use a smartphone in the future. It is also a fact that health has become a really important part in the people's daily life. Therefore, the probability is high that this trend will continue

### 3. How could these trends help influence Bellabeat marketing strategy

A: This trend shows that it could be more worthwhile to target a younger customer base with Bellabeat's products. But there should also be an interest in targeting older people since it can be assumed that demand here will also increase in the future. link.

#### Business task

Identify potential opportunities for growth and recommendations for the Bellabeat marketing strategy improvement based on trends in smart device usage.

## Prepare Phase

*Which dataset will be used for the analysis?*

- The data source used for this case study is called "FitBit Fitness Tracker Data".
- This dataset is stored in Kaggle and was made available through Mobius.
- As it is suggested by google analytics and is free to download in Kaggle it is guaranteed that the data is open-source and can be used without hesitation for statistical analyses.
- 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. Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors/preferences.

*Installing packages and libraries that are necessary for the analysis*

packages

- foreign
- idyverse
- lubridate
- dplyr
- ggplot2
- tidyr
- janitor
- ggpubr

```
library(foreign)
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
library(here)
library(janitor)
library(ggpubr)
library(tidyverse)
```

Read the csv_files for the analysis and rename them for an easier usage

Here I want to use the dailyActivity_merged-dataset and the sleepDay_merged-dataset

```
activity <- read.csv("/Users/sorennonnengart/Desktop/google analytics/Case st
udy/Bellabeat/data_orig/dailyActivity_merged.csv", na="NA", sep=",")


sleep <- read.csv("/Users/sorennonnengart/Desktop/google analytics/Case study
/Bellabeat/data_orig/sleepDay_merged.csv", na="NA", sep=",")
```

Now I will preview the variables of the dataframe activity as an example

```
head(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(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
```

## process phase

There are no NA-values in both datasets which can be shown by the message “integer(0)”

```
which(is.na(activity))

## integer(0)

which(is.na(sleep))

## integer(0)
```

Count the number of NA values —> (There are 0 NA-Values in the datasets)

```
sum(is.na(activity))

## [1] 0

sum(is.na(sleep))

## [1] 0
```

Remove duplicates but first summarize duplicates

```
sum(duplicated(activity))

## [1] 0

sum(duplicated(sleep))

## [1] 3
```

➔ There are 3 duplicates in the sleep-dataset

remove the duplicates with the unique-function

```
sleep <- unique(sleep)
```

➔ 3 duplicates were deleted for the “activity” dataset

rename columns for avoiding problems with case-sensitivity in R to lower case

```
activity <- rename_with(activity, tolower)
sleep <- rename_with(sleep, tolower)
```

Now I’ll use the clean names function in the Janitor package. This will automatically make sure that the

column names are unique and consistent.

```
clean_names(activity)
clean_names(sleep)
```

Time formatting with the **as.POSIXct-function** that converts an object to one of the two classes used to represent date/times (calendar dates plus time to the nearest second). They can convert objects of the other class and of class

"Date" to these classes.

***Dataset: activity***

```
activity$activitydate=as.POSIXct(activity$activitydate, format="%m/%d/%Y", tz
=Sys.timezone())
activity$dt <- format(activity$activitydate, format = "%m/%d/%y")
```

***Dataset: sleep***

```
sleep$sleepday=as.POSIXct(sleep$sleepday, format="%m/%d/%Y %I:%M:%S %p", tz=S
ys.timezone())
sleep$dt <- format(sleep$sleepday, format = "%m/%d/%y")



head(sleep$dt)

"04/12/16" "04/13/16" "04/15/16" "04/16/16" "04/17/16" "04/19/16"

head(activity$dt)

"04/12/16" "04/13/16" "04/14/16" "04/15/16" "04/16/16" "04/17/16"
```

Describe the datasets for getting an overview (seperate results of the single values)

Variable = totalsteps from activity-dataset

Showing the mean, median, range and IQr of the activity-dataset

```
mean(activity$totalsteps)

## [1] 7637.911

median(activity$totalsteps)

## [1] 7405.5

range(activity$totalsteps)

## [1]     0 36019

IQR(activity$totalsteps)

## [1] 6937.25
```

For the sleep-dataset I'll first generate a variable "totalhoursasleep"

```
sleep %>%
  mutate(totalhoursasleep=totalminutesasleep/60) %>%
  summarise(mean(totalhoursasleep))

##   mean(totalhoursasleep)
## 1               6.98622
```

- **The average sleeptime in hours ist 6.98 hours**

*Showing the mean, median, range and IQR of the sleep-dataset*

```
mean(sleep$totalminutesasleep)

419.1732

median(sleep$totalminutesasleep)

432.5

range(sleep$totalminutesasleep)

58 796

IQR(sleep$totalminutesasleep)

129
```

*Summarize the two datasets showing the Min, Max, Median, Mean, 1st and 3rd Quantile*

```
summary(sleep)

##        id                 sleepday                    totalsleeprecords
##  Min.   :1.504e+09   Min.   :2016-04-12 00:00:00   Min.   :1.00
##  1st Qu.:3.977e+09   1st Qu.:2016-04-19 00:00:00   1st Qu.:1.00
##  Median :4.703e+09   Median :2016-04-27 00:00:00   Median :1.00
##  Mean   :4.995e+09   Mean   :2016-04-26 11:38:55   Mean   :1.12
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04 00:00:00   3rd Qu.:1.00
##  Max.   :8.792e+09   Max.   :2016-05-12 00:00:00   Max.   :3.00
##  totalminutesasleep totaltimeinbed       dt
##  Min.   : 58.0      Min.   : 61.0   Length:410
##  1st Qu.:361.0      1st Qu.:403.8   Class :character
##  Median :432.5      Median :463.0   Mode  :character
##  Mean   :419.2      Mean   :458.5
##  3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :796.0      Max.   :961.0
```

```
summary(activity)

##        id             activitydate                   totalsteps
##  Min.   :1.504e+09   Min.   :2016-04-12 00:00:00   Min.   :    0
##  1st Qu.:2.320e+09   1st Qu.:2016-04-19 00:00:00   1st Qu.: 3790
##  Median :4.445e+09   Median :2016-04-26 00:00:00   Median : 7406
##  Mean   :4.855e+09   Mean   :2016-04-26 06:53:37   Mean   : 7638
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04 00:00:00   3rd Qu.:10727
##  Max.   :8.878e+09   Max.   :2016-05-12 00:00:00   Max.   :36019
##  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
##     calories         dt
##  Min.   :   0   Length:940
##  1st Qu.:1828   Class :character
##  Median :2134   Mode  :character
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

Merging the datasets now

```
activity_sleep_merged <- merge(activity, sleep, by=c("id", "dt"))
```

Create a subset of the dataset called “df_as” for analyzing the variables I am interested in

```
df_as <- subset(activity_sleep_merged, select=c("id","dt","totalsteps","total
distance",
         "veryactivedistance", "calories", "totalminutesasleep"))
nrow(df_as) # only 410 rows left

410
```

## Analyze phase

First: I generate the mean values for every user for the variables shown in bracktes and save them in a new dataset called “average_dist”

```
average_dist <- df_as %>%
  group_by(id) %>%
  summarise (mean_steps = mean(totalsteps), mean_calories = mean(calories), m
ean_sleep = mean(totalminutesasleep),
             mean_activedist = mean(veryactivedistance), mean_dist = mean(tot
aldistance))

head(average_dist)
```

```
## # A tibble: 6 × 6
##           id mean_steps mean_calories mean_sleep mean_activedist mean_dist
##        <dbl>      <dbl>         <dbl>      <dbl>           <dbl>     <dbl>
## 1 1503960366     12406.         1872.       360.         2.77         7.97
## 2 1644430081      7968.         2978.       294          0.175        5.79
## 3 1844505072      3477          1676.       652          0            2.30
## 4 1927972279      1490          2316.       417          0            1.03
## 5 2026352035      5619.         1541.       506.         0.00679      3.49
## 6 2320127002      5079          1804         61          0            3.42
```

I now create different active groups in relation to the steps they made and label them as

group 1: <5000 (fewe steps) to group 4: >=1000 (many) steps

```
average_dist$steps_group[which(average_dist$mean_steps <5000)] <- 1
average_dist$steps_group[which(average_dist$mean_steps >=5000 & average_dist$
mean_steps<7500)] <- 2
average_dist$steps_group[which(average_dist$mean_steps >=7500 & average_dist$
mean_steps<10000)] <- 3
average_dist$steps_group[which(average_dist$mean_steps >=10000)] <- 4
## label values
average_dist$steps_group <- ordered(average_dist$steps_group, levels=c(1,2,3,
4),
                                    labels=c("G1: <5000", "G2: >=5000 & <7500",
                                             "G3: >=7500 & <1000", "G4: >=10000"))
```

And there you can see the number of groups in a tabel

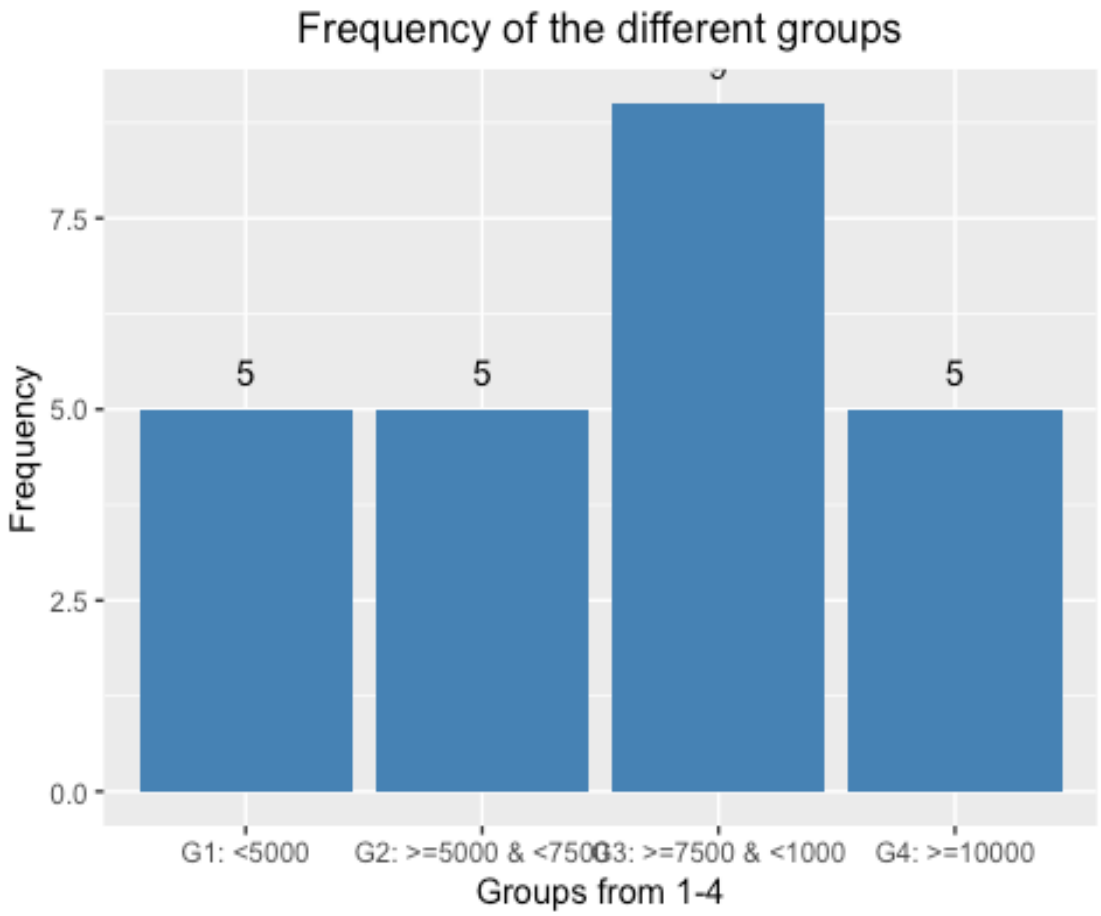
```
average_dist %>%
  group_by(steps_group) %>%
  summarise(n = n())
```

```
## # A tibble: 4 × 2
##   steps_group            n
##   <ord>              <int>
## 1 G1: <5000              5
## 2 G2: >=5000 & <7500     5
## 3 G3: >=7500 & <1000     9
## 4 G4: >=10000            5
```

**Now I will analyse the data with different plots like bargraphs, linegraphs, scatterplots and so forth…**

Get the same output above shown by a bar graph

```
ggplot(data=average_dist, aes(x=steps_group)) +
  geom_bar(fill="steelblue") +
  labs(y="Frequency", x="Groups from 1-4") +
  ggtitle("Frequency of the different groups") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label=stat(count)), stat="count", vjust=-1)
```



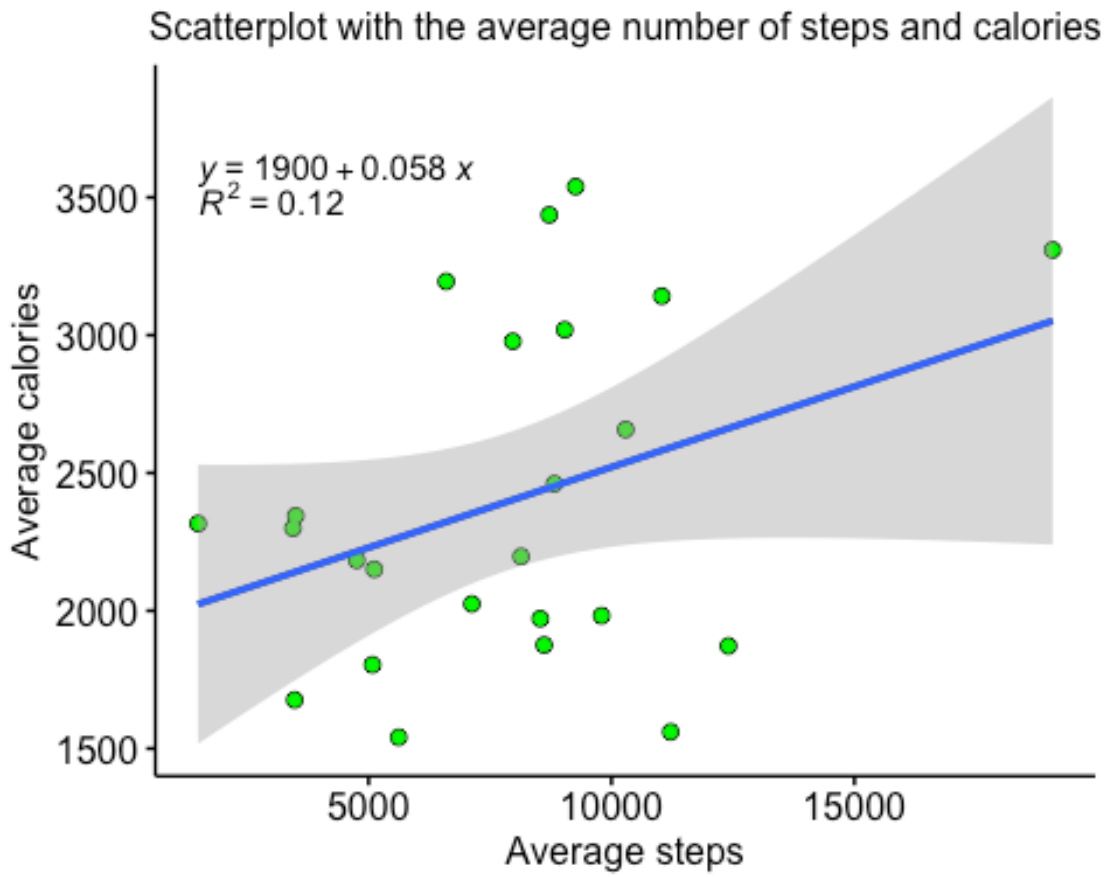
The scatterplot below shows the relationship between “average steps taken by the users” and the average amount of calories that was burned

The b-coefficient of 0.058 in the upper left of the regression equation indicates that for 1000 additional steps, on average 50.8 calories

more are consumed. The $R^2$-value means, that the x-value “Average steps” explains 12 % of the varianceof the y value “Average calories” which is quite good for just one variable

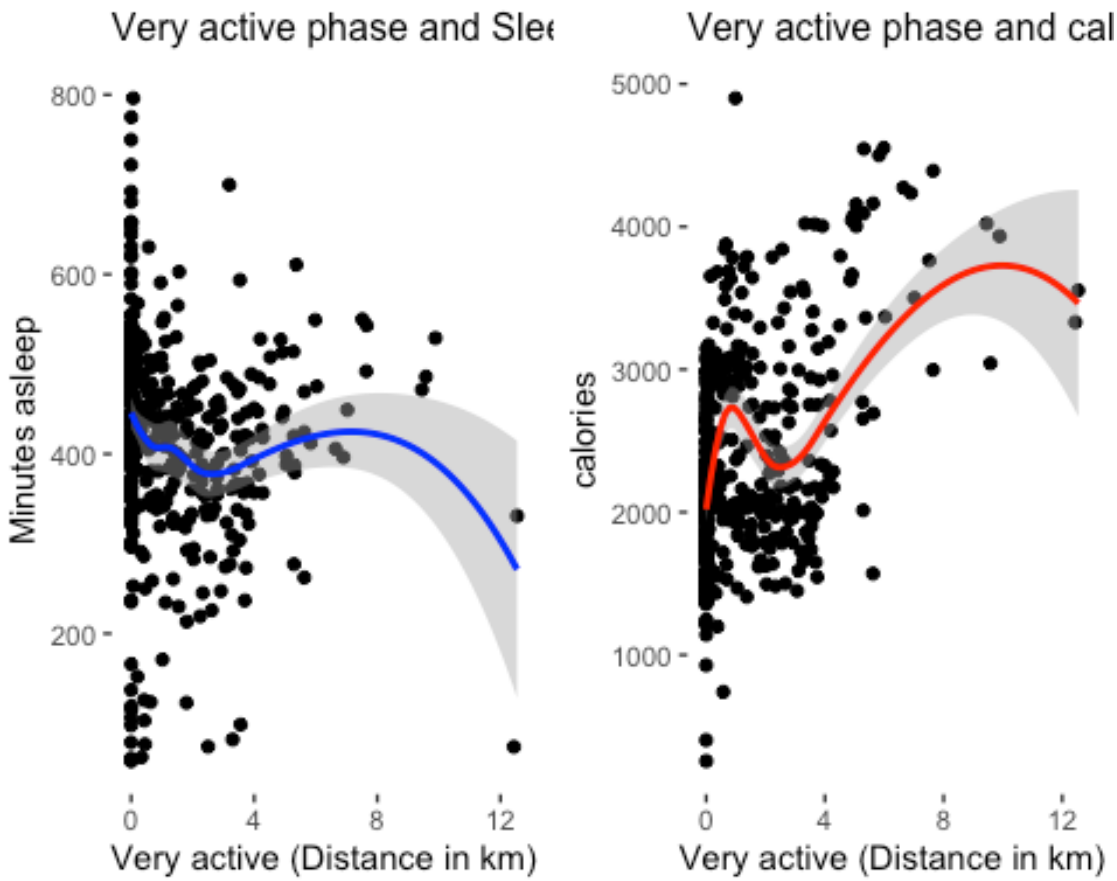
```
ggscatter(average_dist, x = "mean_steps", y = "mean_calories", add ="reg.line
") +
    geom_point(color="green") +
    geom_smooth(formula = y ~ x, method = "lm") +
    stat_regline_equation(label.y = 3600, label.x = 1500) +
    stat_cor(aes(label = paste(..rr.label..)), label.y = 3500, label.x = 1500
) +
    labs(x="Average steps", y="Average calories",
         title="Scatterplot with the average number of steps and calories") +
    theme(plot.title = element_text(hjust = 0.5, size=12))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



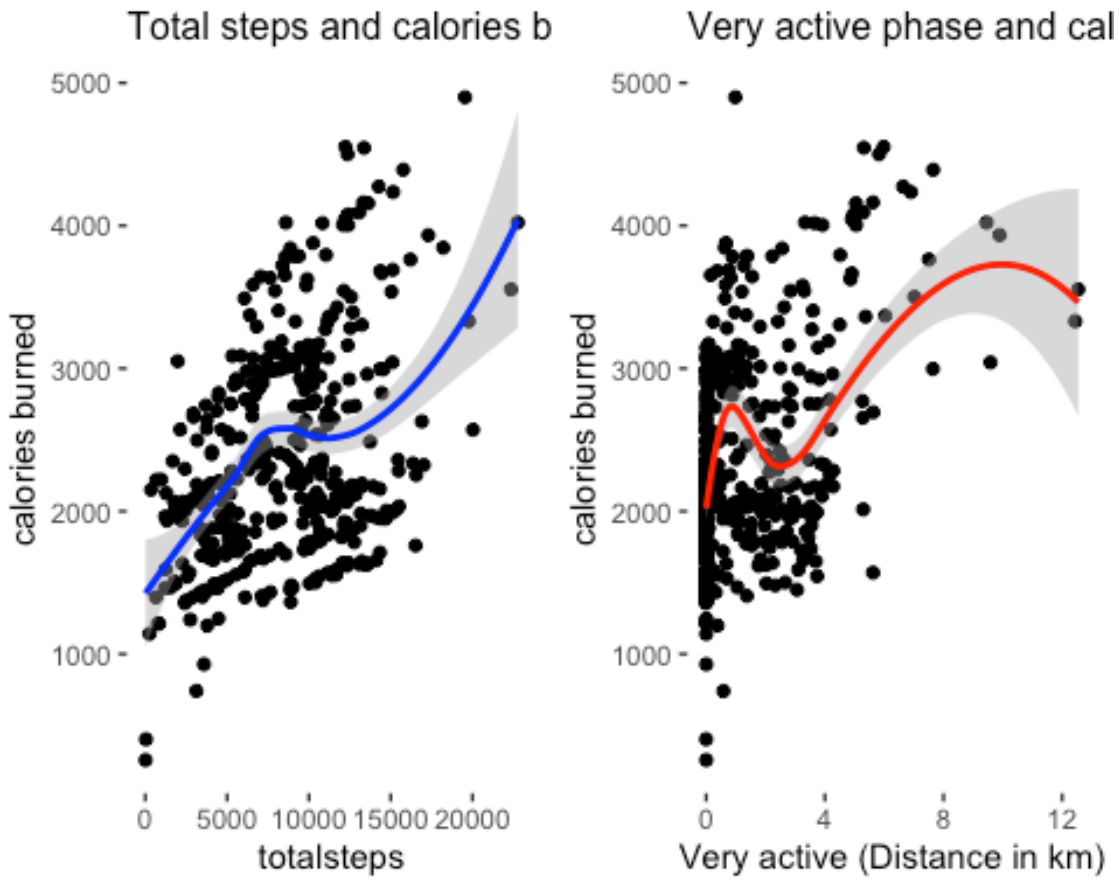
Is there a relationship between sleep minutes and steps and also between sleep minutes and very active phases?

```
ggarrange(
  ggplot(df_as, aes(x=veryactivedistance, y=totalminutesasleep)) +
    geom_jitter() +
    geom_smooth(color = "blue") +
    labs(title = "Very active phase and Sleeptime (Minutes)", x = "Very activ
e (Distance in km)",
         y= "Minutes asleep") +
    theme(panel.background = element_blank(),
          plot.title = element_text( size=12)),
  ggplot(df_as, aes(x=veryactivedistance, y=calories))+
    geom_jitter() +
    geom_smooth(color = "red") +
    labs(title = "Very active phase and calories", x = "Very active (Distance
in km)",
         y= "calories") +
    theme(panel.background = element_blank(),
          plot.title = element_text( size=12)))

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



Is it helpful to be very active when it comes to burn calories compared to only being active?

```
ggarrange(
  ggplot(df_as, aes(x=totalsteps, y=calories)) +
    geom_jitter() +
    geom_smooth(color = "blue") +
    labs(title = "Total steps and calories burned", x = "totalsteps",
         y= "calories burned") +
    theme(panel.background = element_blank(),
          plot.title = element_text( size=12)),
  ggplot(df_as, aes(x=veryactivedistance, y=calories))+
    geom_jitter() +
    geom_smooth(color = "red") +
    labs(title = "Very active phase and calories", x = "Very active (Distance
in km)",
         y= "calories burned") +
    theme(panel.background = element_blank(),
          plot.title = element_text( size=12)))

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



The last graph is a linegraph that shows the change of total steps taken over 30 days

So the first thing I have to do is to change the values of the date-formatted variable “dt” to get the number of rows for each individual

…. But I first I will create a subset with the variables that are needed for the visualization

```
df_line <- subset(df_as, select=c('id', 'totalsteps', 'calories', 'totalminut
esasleep'))
```

How many days there are for each person?

```
df_line$help <- 1
df_line$days <- ave(df_line$help, by=df_line$id, FUN=cumsum)  #days numbered
consecutively (within person)
nrow(df_line[df_line$days==1,]) #number of persons

## [1] 24
```

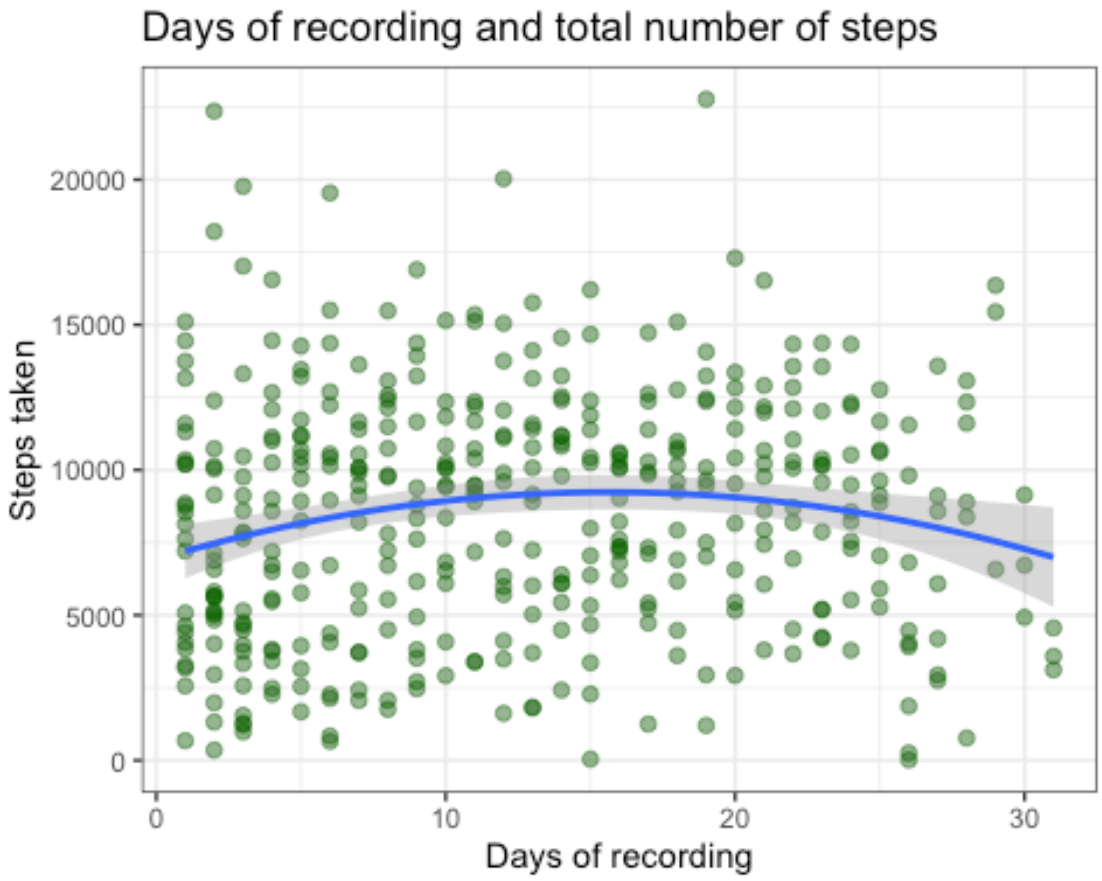
- Example: For person 1 there are 24 days. That means, that for this person exists 24 values for the variable “dt”

Now I will show the change of total steps taken over time for all individuals together

I also modified the plot with a quadratic regression-funtion that shows that the relationship is not linear.

First, the steps increase steadily until day 15 and then decrease again until day 30.

```
df_line %>%
  ggplot(aes(x=days, y=totalsteps)) +
  geom_point(size = 2, colour = "darkgreen", alpha = 0.5) +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1) +
  labs(x="Days of recording", y="Steps taken",
       title = "Days of recording and total number of steps") +
  theme(plot.title = element_text(hjust = 0.5, size=12)) +
  theme_bw()
```

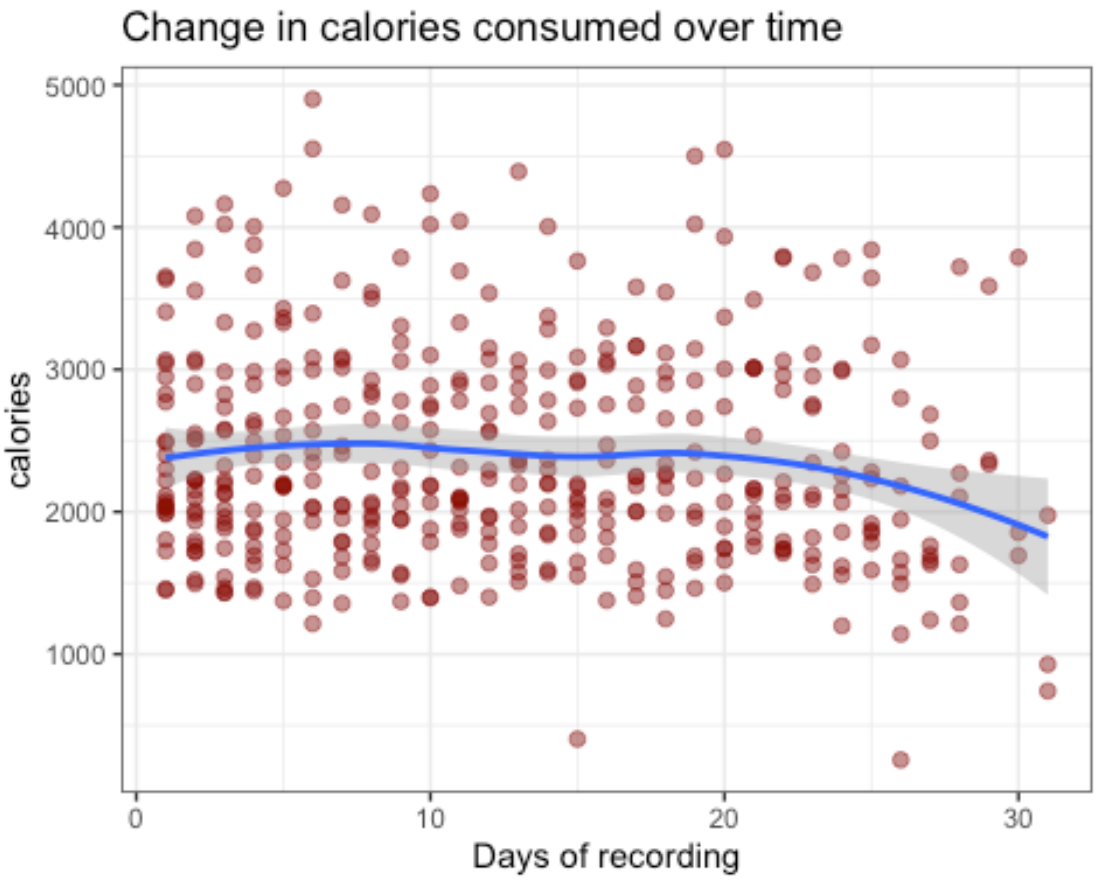



Is there also a change in calories consumed over time?

This correlation obviously corresponds to the decrease in steps taken over time and, as can be seen from the plot.

This can be seen by the decrease of the calorie consumption over time.

```
df_line %>%
  ggplot(aes(x=days, y=calories)) +
  geom_point(size = 2, colour = "darkred", alpha = 0.5) +
  geom_smooth() +
  labs(x="Days of recording", y="calories",
       title = "Change in calories consumed over time") +
  theme(plot.title = element_text(hjust = 0.5, size=12)) +
  theme_bw()
```




## Act Phase

Based on my analysis I have found different trends that may help to online campaign and improve Bellabeat app:

Recommendation Description

1. Based on the analyses, it can be determined that there are various groups that are active in different ways, ranging from very inactive to very active. Accordingly, an attempt can be made to incorporate a feature into the app that provides certain “motivational aids” for the less active users.

2. The results also show impressively that the more steps are taken, the more calories are burned. This is not surprising, of course. Possibly, based on the graphs showing the correlation between active phases and calorie consumption, it can be determined that particularly active phases could have an additional effect on calorie consumption. Therefore, it might make sense to reward people who are particularly active. What such a reward might look like must of course be discussed in more detail

3. Another result shows that the motivation to be active first increases and then decreases again. Here, too, it seems reasonable to me to work with rewards of any kind to keep the users in a good mood and to guarantee that they keep their steps constant over time or even increase them.