Bellabeat_case_study

SCENARIO

You are a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide marketing strategy for the company. You will present your analysis to the Bellabeat executive team along with your high-level recommendations for Bellabeat's marketing strategy

1. Ask

The goal of this project is to define a new marketing strategy for Bellabeat company in order to grow the sales of its Smart devices. In order to do that, Smart devices data will be explored to have an overview of how the customers use these devices and have an idea of the user profile to address the marketing campaign. The analysis results will be presented to the company co founders.

2. Prepare

The dataset that will be used for the analysis is the following one:

FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

The dataset is organized in different .csv files storing data about:

- Sleeping hours
- Activity
- Intensities
- Calories
- Weight
- Heartrate

There are minute, hour and daily level files. For the minute level files there are both the long and wide format.

I will focus on the following files for the analysis:

- dailyActivity_merged.csv → It contains information about the daily steps, walked distance, activity and calories of 33 different users (940 observations).
- weightLogInfo_merged.csv → It contains information about weight logs of different users in Kg and Pounds, as well as the Body Mass Index (BMI), a measure of the body corpulence based on the height and weight of the person (68 observations). It has only information about 8 users, which do not represent the overall population, so I am not going to use it for the analysis.

- sleepDay_merged.csv → It contains information about the daily sleep records, minutes asleep and time in bed of different users (462 observations).
- heartrate_seconds_merged.csv → It contains information about the heart rate of different users, which is measured each 5 seconds (2483658 observations).
- minuteMETsNarrow_merged.csv → It contains information about minute measures of the METs (metabolic equivalents) of 33 different users (1325580 observations). As it is defined in this article, "One MET is defined as the energy you use when you're resting or sitting still. An activity that has a value of 4 METs means you're exerting four times the energy than you would if you were sitting still.

As the dataset does not contain information about the users age and gender, I am going to consider that the samples have been taken randomly and they represent the whole population. Nevertheless, I will try to obtain a user profile from the analysis.

3. Process

We will use R for analysis and visualizatin. First of all, I have imported the different tables of the dataset: (before importing the dataset lets install and load all the important libraries:)

```
install.packages("tidyverse")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
install.packages("here")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
install.packages("skimr")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(here)
```

```
## here() starts at /cloud/project
```

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
heartrate <- read_csv('heartrate_seconds_merged.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   Time = col_character(),
##   Value = col_double()
## )
```

```
sleep <- read_csv('sleepDay_merged.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   SleepDay = col_character(),
##   TotalSleepRecords = col_double(),
##   TotalMinutesAsleep = col_double(),
##   TotalTimeInBed = col_double()
## )
```

```
activity <- read_csv('dailyActivity_merged.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityDate = col_character(),
##   TotalSteps = col_double(),
##   TotalDistance = col_double(),
##   TrackerDistance = col_double(),
##   LoggedActivitiesDistance = col_double(),
##   VeryActiveDistance = col_double(),
##   ModeratelyActiveDistance = col_double(),
##   LightActiveDistance = col_double(),
##   SedentaryActiveDistance = col_double(),
##   VeryActiveMinutes = col_double(),
##   FairlyActiveMinutes = col_double(),
```

```
##   LightlyActiveMinutes = col_double(),
##   SedentaryMinutes = col_double(),
##   Calories = col_double()
## )
```

```
MET <- read_csv('minuteMETsNarrow_merged.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityMinute = col_character(),
##   METs = col_double()
## )
```

NOW LETS TAKE A LOOK AT THE FOLLOWING TABLES:-

- **Heart Rate**

```
skim_without_charts(heartrate)
```

Table 1: Data summary

Name	heartrate
Number of rows	2483658
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Time	0	1	19	21	0	961274	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.513765e+09	2950223761.2	2022484408	4388161847	5553957443	6962181067	8877689391
Value	0	1	7.733000e+01	19.4	36	63	73	88	203

```
head(heartrate)
```

```
## # A tibble: 6 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
```

The table contains a numeric Id for the different users, a numeric value for the heart rate and the time of the measure with char format.

```
heartrate %>%
  group_by(Id) %>%
  summarize(max_rate = max(Value),min_rate = min(Value), mean_rate = mean(Value))
```

```
## # A tibble: 14 x 4
##            Id max_rate min_rate mean_rate
##         <dbl>    <dbl>    <dbl>     <dbl>
##  1 2022484408      203       38      80.2
##  2 2026352035      125       63      93.8
##  3 2347167796      195       49      76.7
##  4 4020332650      191       46      82.3
##  5 4388161847      180       39      66.1
##  6 4558609924      199       44      81.7
##  7 5553957443      165       47      68.6
##  8 5577150313      174       36      69.6
##  9 6117666160      189       52      83.7
## 10 6775888955      177       55      92.0
## 11 6962181067      184       47      77.7
## 12 7007744171      166       54      91.1
## 13 8792009665      158       43      72.5
## 14 8877689391      180       46      83.6
```

Before transforming some inconsistent data, I have taken a quick look at the table. It has information about 14 users (less than the 50% of the population) but, as I think this data is important I will perform a reduced analysis of it. The normal average heart rate for adults is between 60 and 100 bpm, so it seems that the values are coherent.

I have arranged the columns names and time and date formats, so the result table is like the following one:

```
heartrate_clean <- heartrate %>%
   rename_with(tolower) %>%
   rename(rate_value=value) %>%
   mutate(date=format(as.POSIXct(time, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone()), format = "%m/%
   mutate(time=format(as.POSIXct(time, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone()), format = "%H:%
```

now the table will look like:

```
head(heartrate_clean)
```

```
## # A tibble: 6 x 4
##           id time     rate_value date
##        <dbl> <chr>         <dbl> <chr>
## 1 2022484408 07:21:00         97 04/12/16
## 2 2022484408 07:21:05        102 04/12/16
## 3 2022484408 07:21:10        105 04/12/16
## 4 2022484408 07:21:20        103 04/12/16
## 5 2022484408 07:21:25        101 04/12/16
```

```
## 6 2022484408 07:22:05         95 04/12/16
```

- **Sleep Records**

```
skim_without_charts(sleep)
```

Table 4: Data summary

Name	sleep
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
SleepDay	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.000979e+09	2.060360e+09	1.503960e+09	4.977333e+09	4.702922e+09	6.962181e+09	8.792010e+09
TotalSleepRecords	0	1	1.120000e+00	3.150000e-01	1	1	1	1	3
TotalMinutesAsleep	0	1	4.194700e+02	1.218340e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.227100e+02	61	403	463	526	961

```
head(sleep)
```

```
## # A tibble: 6 x 5
##          Id SleepDay            TotalSleepRecor~ TotalMinutesAsle~ TotalTimeInBed
##       <dbl> <chr>                          <dbl>             <dbl>          <dbl>
## 1    1.50e9 4/12/2016 12:00:0~                 1               327            346
## 2    1.50e9 4/13/2016 12:00:0~                 2               384            407
## 3    1.50e9 4/15/2016 12:00:0~                 1               412            442
## 4    1.50e9 4/16/2016 12:00:0~                 2               340            367
## 5    1.50e9 4/17/2016 12:00:0~                 1               700            712
## 6    1.50e9 4/19/2016 12:00:0~                 1               304            320
```

The table contains a numeric Id for the different users, a numeric double value for the sleep records and the time of the measure with char format.

```
sleep%>%
   group_by(Id) %>%
   summarize(max_asleep = max(TotalMinutesAsleep),min_asleep = min(TotalMinutesAsleep), mean_asleep = me
```

```
## # A tibble: 24 x 4
##            Id max_asleep min_asleep mean_asleep
##         <dbl>      <dbl>      <dbl>       <dbl>
##  1 1503960366        700        245        360.
##  2 1644430081        796        119        294
##  3 1844505072        722        590        652
##  4 1927972279        750        166        417
##  5 2026352035        573        357        506.
##  6 2320127002         61         61         61
##  7 2347167796        556        374        447.
##  8 3977333714        424        152        294.
##  9 4020332650        501         77        349.
## 10 4319703577        692         59        477.
## # ... with 14 more rows
```

It has information about 24 users (more than the 70% of the population) but there are only 15 users that have more than 15 observations, which represent the 50% of the time range analyzed.

It can be observed that some users have really short sleep records some days, which is not normal and seem to be bad lectures, so I am going to discard the sleep records under 4h. I have also checked if there are records with less time in bed than asleep minutes but it seems that there are not errors like that. After arranging the time to date format and discarding some observations, the table looks like this:

```
sleep_clean <- sleep %>%
   rename_with(tolower) %>%
   clean_names() %>%
   rename(date=sleepday,sleep_records=totalsleeprecords,minutes_asleep=totalminutesasleep,time_bed=total
   mutate(date=format(as.POSIXct(date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone()), format = "%m/%
   filter(minutes_asleep>240)
```

-

Activity Records

```
skim_without_charts(activity)
```

Table 7: Data summary

Name	activity
Number of rows	940
Number of columns	15
Column type frequency:	
character	1
numeric	14
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDate	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.027200e+04	3.601900e+04
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	1.000000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.000000e-01	8.000000e-01	6.480000e+00
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.110000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.799000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.280000e+02
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440000e+03
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03

```
activity  %>%
  group_by(Id) %>%
   summarize(max_steps = max(TotalSteps),min_steps = min(TotalSteps), mean_steps = mean(TotalSteps))
```

```
## # A tibble: 33 x 4
##            Id max_steps min_steps mean_steps
##         <dbl>     <dbl>     <dbl>      <dbl>
##  1 1503960366     18134         0     12117.
##  2 1624580081     36019      1510      5744.
##  3 1644430081     18213      1223      7283.
##  4 1844505072      8054         0      2580.
##  5 1927972279      3790         0       916.
##  6 2022484408     18387      3292     11371.
##  7 2026352035     12357       254      5567.
##  8 2320127002     10725       772      4717.
##  9 2347167796     22244        42      9520.
## 10 2873212765      9685      2524      7556.
## # ... with 23 more rows
```

The table contains a numeric Id for the different users, numeric values for the activity observations and the time .

It is clearly seen that their are no null records. Also their are some columns which is incorrectly formatted like Activity. I am also going to ignore these observations that present a total of daily steps under 100, as all persons have some little activity every day. I am not going to analyse the different intensities distances, but the time. The result table for analysis is the following one:

```
activity_clean <- activity %>%
   select(Id, ActivityDate, TotalSteps, TotalDistance, VeryActiveMinutes, FairlyActiveMinutes, LightlyA
   rename_with(tolower) %>%
   clean_names() %>%
   rename(date=activitydate,total_steps=totalsteps,total_distance=totaldistance,very_active_min=veryacti
   mutate(date=format(as.POSIXct(date, format="%m/%d/%Y", tz=Sys.timezone()), format = "%m/%d/%y")) %>%
   filter(total_steps>100)
```

4. Analyze and 5. Share

First of all, I am going to transform the heart rate table to have daily records instead of seconds values and allow me to merge and compare these observations with the other tables.

As the heart rate dataset has less users than the others, I am going to merge first the sleep and activity observations.

```
daily_activity_merged <- merge(sleep_clean, activity_clean, by=c('id', 'date'))
 head(daily_activity_merged)
```

```
##           id     date sleep_records minutes_asleep time_bed total_steps
## 1 1503960366 04/12/16             1            327      346       13162
## 2 1503960366 04/13/16             2            384      407       10735
## 3 1503960366 04/15/16             1            412      442        9762
## 4 1503960366 04/16/16             2            340      367       12669
## 5 1503960366 04/17/16             1            700      712        9705
## 6 1503960366 04/19/16             1            304      320       15506
##   total_distance very_active_min fairly_active_min lightly_active_min
## 1           8.50              25                13                328
## 2           6.97              21                19                217
## 3           6.28              29                34                209
## 4           8.16              36                10                221
## 5           6.48              38                20                164
## 6           9.88              50                31                264
##   sedentary_min calories
## 1           728     1985
## 2           776     1797
## 3           726     1745
## 4           773     1863
## 5           539     1728
## 6           775     2035
```

Let's take a quick look at the table statistics:

```
summary(daily_activity_merged)
```

```
##        id                 date           sleep_records   minutes_asleep
##  Min.   :1.504e+09   Length:381         Min.   :1.000   Min.   :245.0
##  1st Qu.:3.977e+09   Class :character   1st Qu.:1.000   1st Qu.:383.0
##  Median :4.703e+09   Mode  :character   Median :1.000   Median :441.0
##  Mean   :5.066e+09                      Mean   :1.126   Mean   :442.1
##  3rd Qu.:6.962e+09                      3rd Qu.:1.000   3rd Qu.:498.0
##  Max.   :8.792e+09                      Max.   :3.000   Max.   :796.0
##     time_bed      total_steps    total_distance   very_active_min
##  Min.   :257.0   Min.   :  254   Min.   : 0.160   Min.   :  0.00
##  1st Qu.:417.0   1st Qu.: 5325   1st Qu.: 3.620   1st Qu.:  0.00
##  Median :471.0   Median : 8954   Median : 6.370   Median :  9.00
```

```
##  Mean   :480.9   Mean   : 8560   Mean   : 6.051   Mean   : 25.71
##  3rd Qu.:535.0   3rd Qu.:11193   3rd Qu.: 7.920   3rd Qu.: 38.00
##  Max.   :961.0   Max.   :22770   Max.   :17.540   Max.   :210.00
##  fairly_active_min lightly_active_min sedentary_min       calories
##  Min.   :  0.00    Min.   : 17.0      Min.   : 125.0   Min.   : 741
##  1st Qu.:  0.00    1st Qu.:159.0      1st Qu.: 623.0   1st Qu.:1861
##  Median : 11.00    Median :209.0      Median : 711.0   Median :2220
##  Mean   : 16.92    Mean   :219.3      Mean   : 693.3   Mean   :2410
##  3rd Qu.: 26.00    3rd Qu.:266.0      3rd Qu.: 772.0   3rd Qu.:2924
##  Max.   :116.00    Max.   :518.0      Max.   :1058.0   Max.   :4900
```

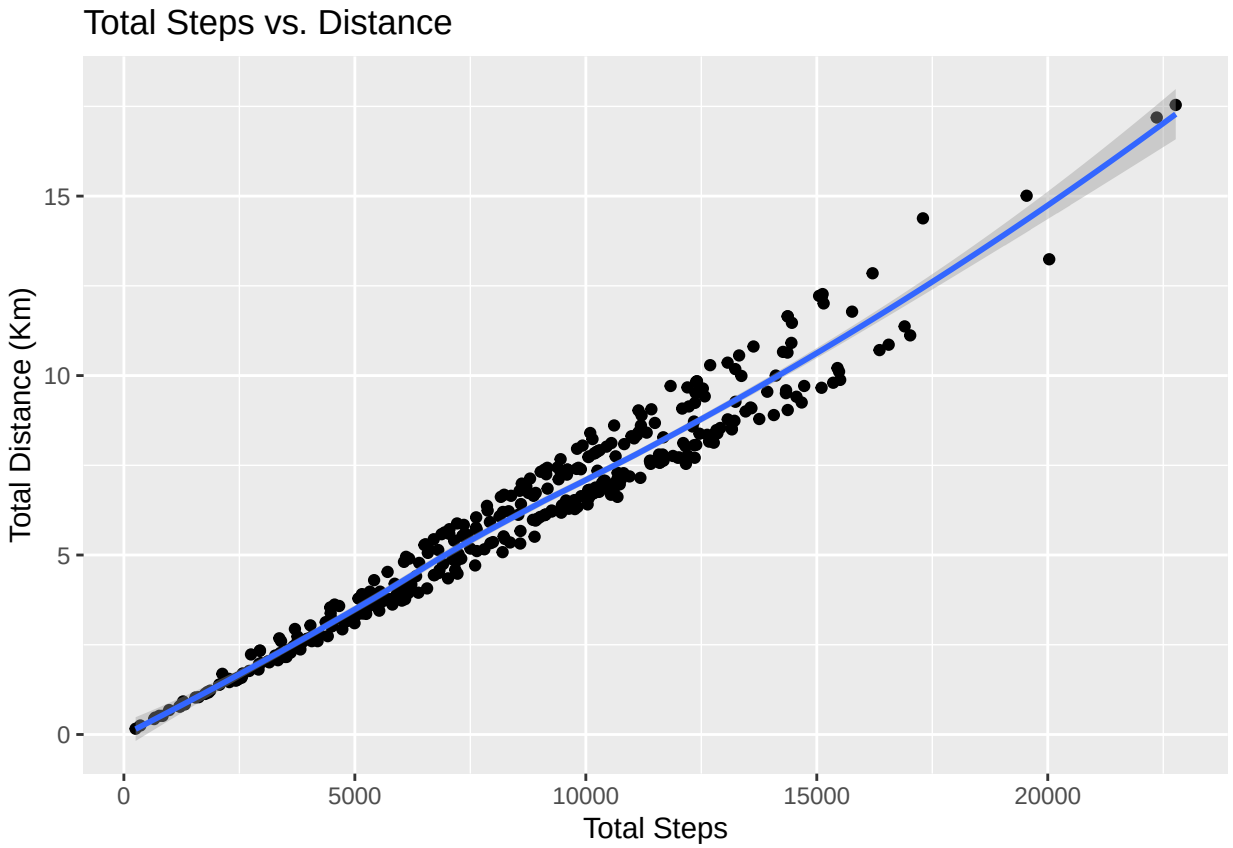
With this first summary it can be observed that:

- The users sleep an average of 7.3 hours and are in bed an average of 8 hours.
- The users walk an average of 8000 steps and 6 Km, which is the recommended.
- The average sedentary time of the users is around 11 hours, which seems a lot, and 40 min of active time.

I am going to first compare the total steps with different activity parameters.

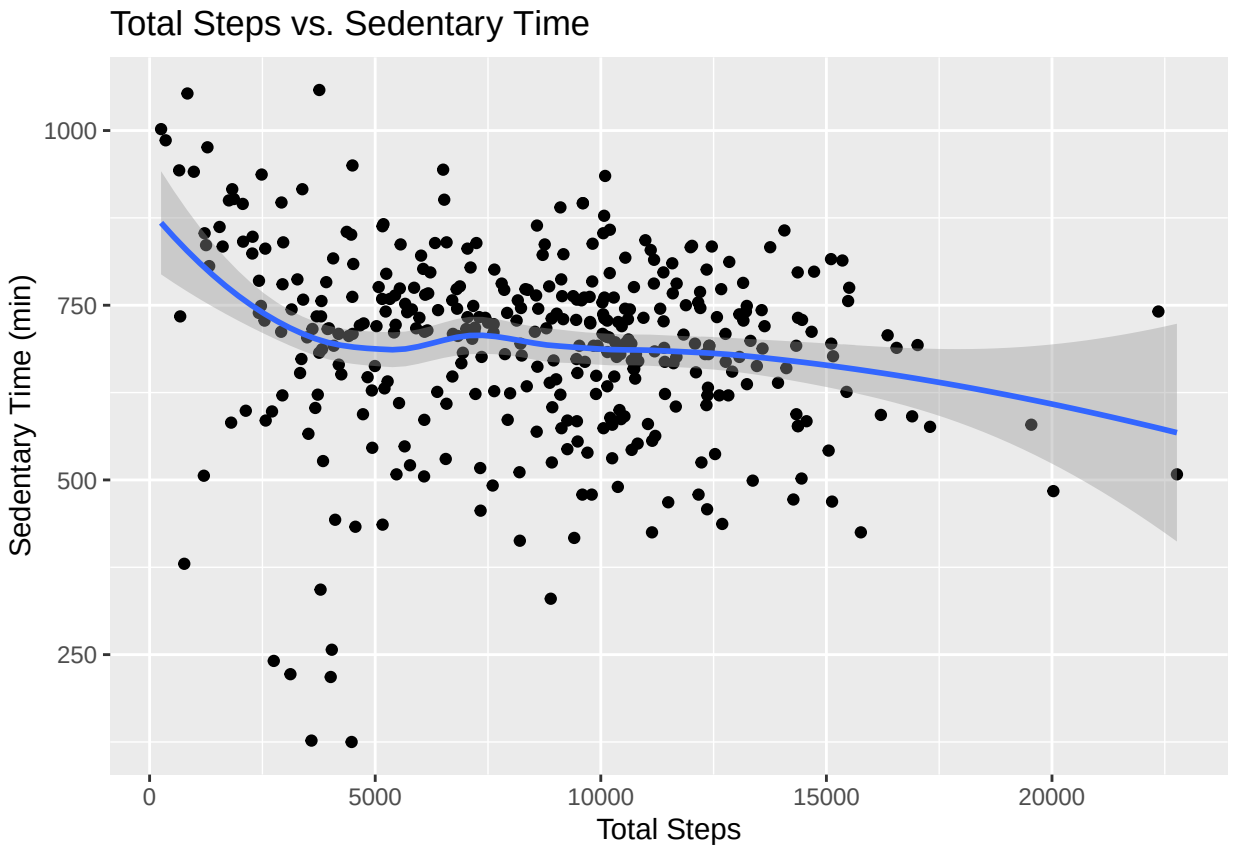
```
ggplot(data=daily_activity_merged, aes(x=total_steps, y=total_distance)) +
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Distance",x="Total Steps" ,y="Total Distanc
```

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



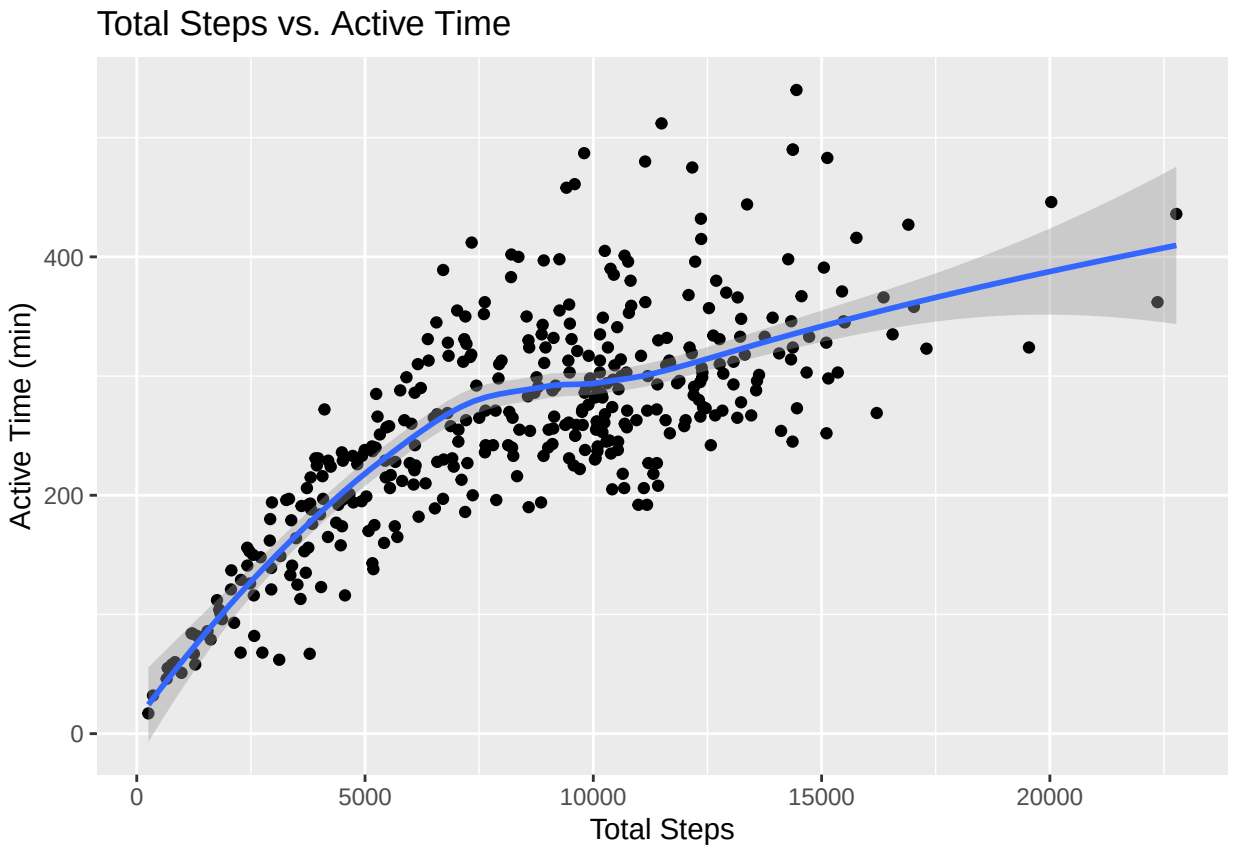

```
ggplot(data=daily_activity_merged, aes(x=total_steps, y=sedentary_min)) +
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Sedentary Time",x="Total Steps" ,y="Sedenta
```

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




```
ggplot(data=daily_activity_merged, aes(x=total_steps, y=very_active_min+fairly_active_min+lightly_activ
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Active Time",x="Total Steps" ,y="Active Ti
```

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




```
ggplot(data=daily_activity_merged, aes(x=total_steps, y=calories)) +
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Calories",x="Total Steps" ,y="Calories" )
```

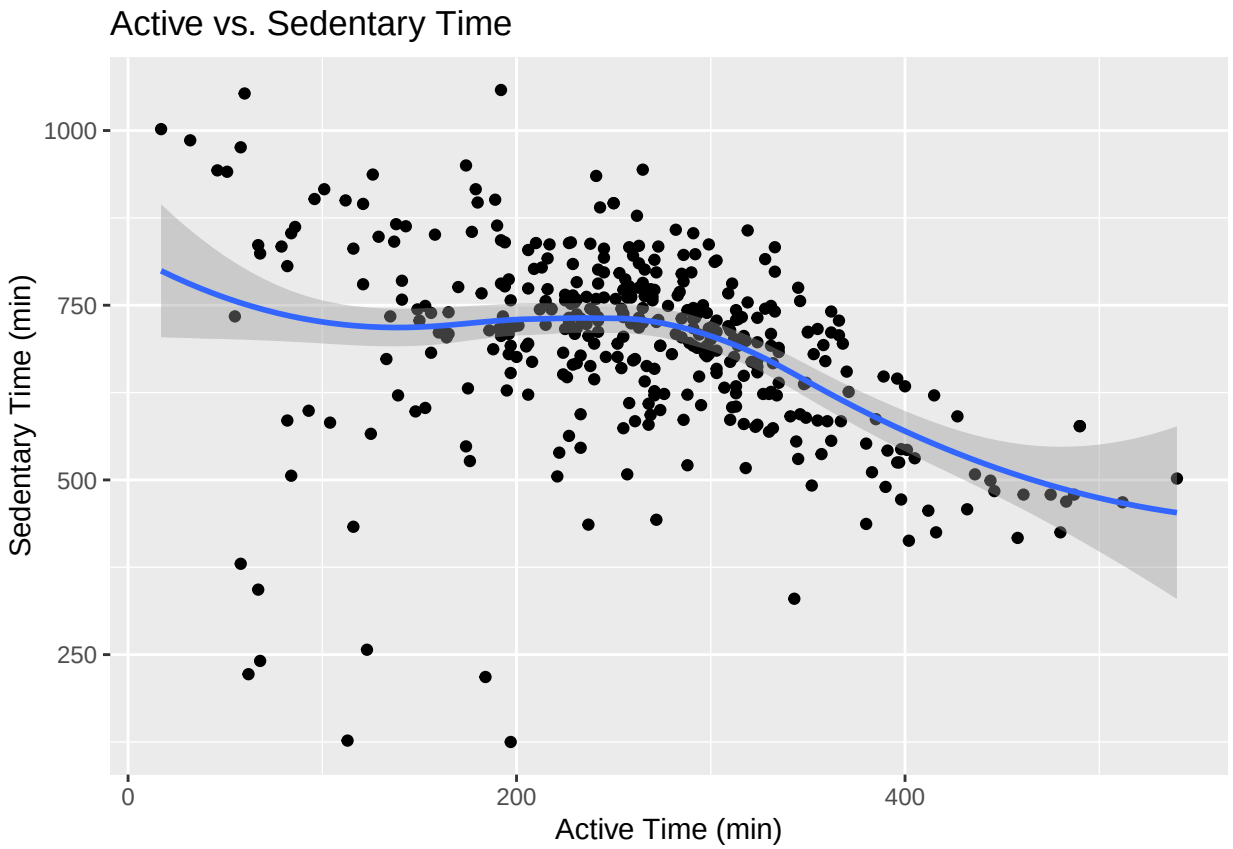
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

The number of steps taken by a user daily is proportional with the walked distance as expected. It seems that the active time also increases with the steps taken but it's not as linear and the same happens with the calories. In the case of the sedentary time, it shows an inverse relation with the steps but it's very scattered.

Let's see which relationships appear comparing the active time with the sedentary time and calories.

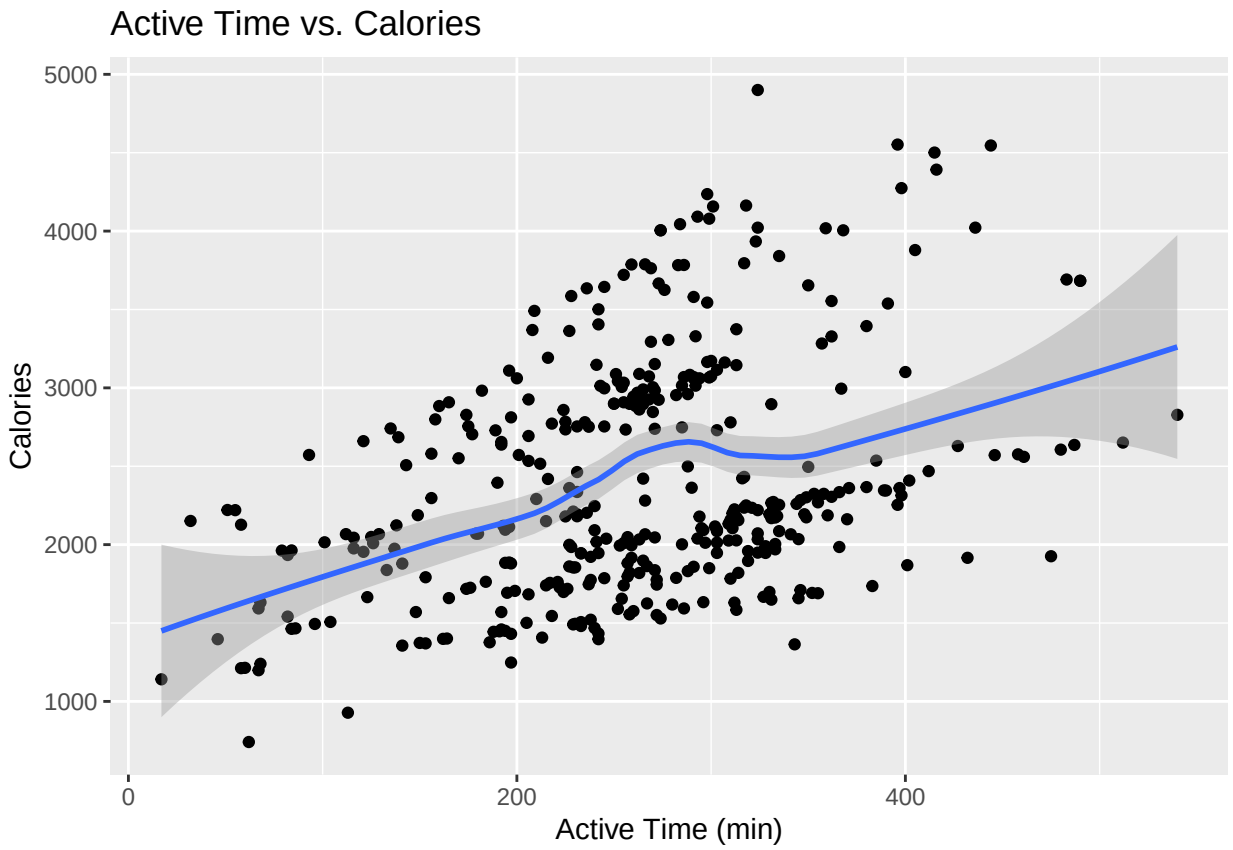
```
ggplot(data=daily_activity_merged, aes(x=very_active_min+fairly_active_min+lightly_active_min, y=sedenta
  geom_point() + geom_smooth() + labs(title="Active vs. Sedentary Time",x="Active Time (min)" ,y="Seden
```

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




```
ggplot(data=daily_activity_merged, aes(x=very_active_min+fairly_active_min+lightly_active_min, y=calori
  geom_point() + geom_smooth() + labs(title="Active Time vs. Calories",x="Active Time (min)" ,y="Calori
```

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

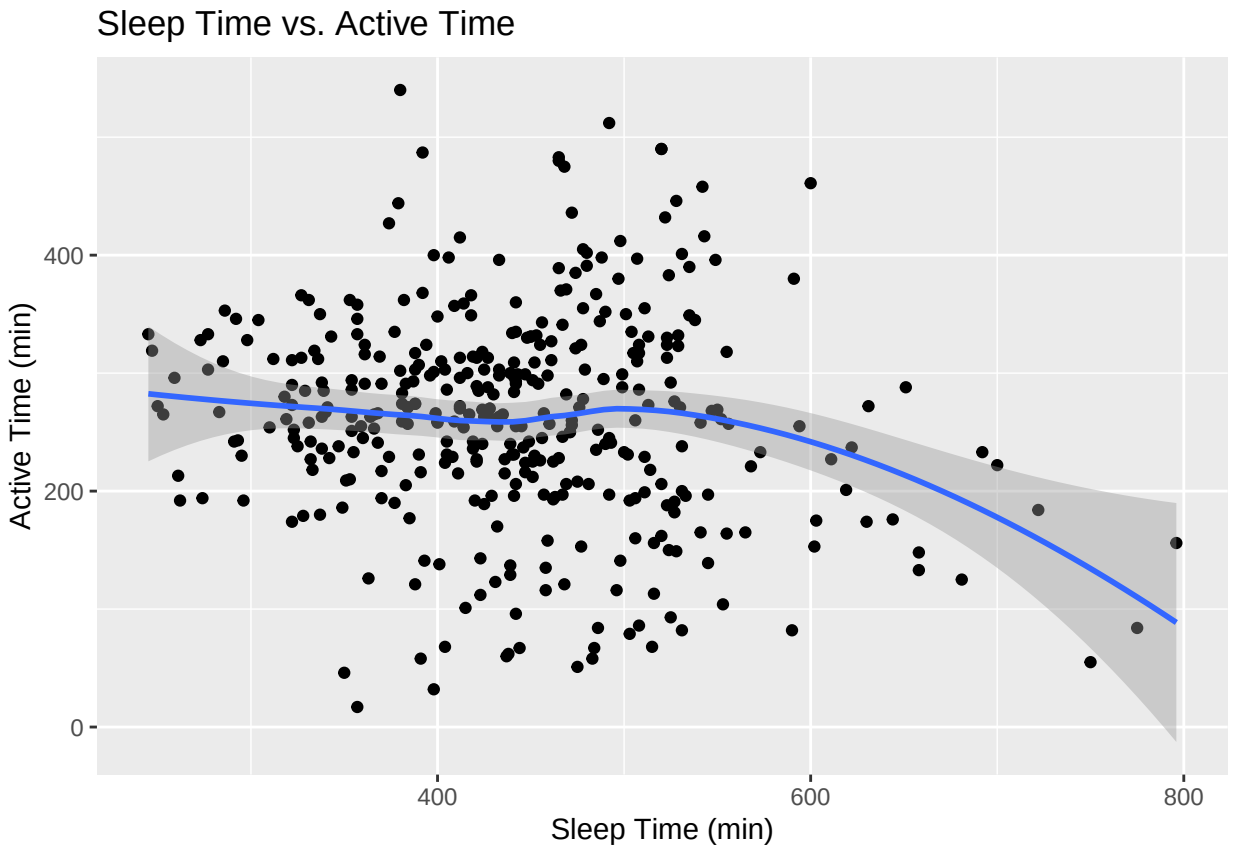


It is also expected that the sedentary time decreases when the active time is higher and the calories increase with the active time.

Now that we have checked that the activity trends are the expected ones, let's see which relationship appears between the sleep time and the activity time.

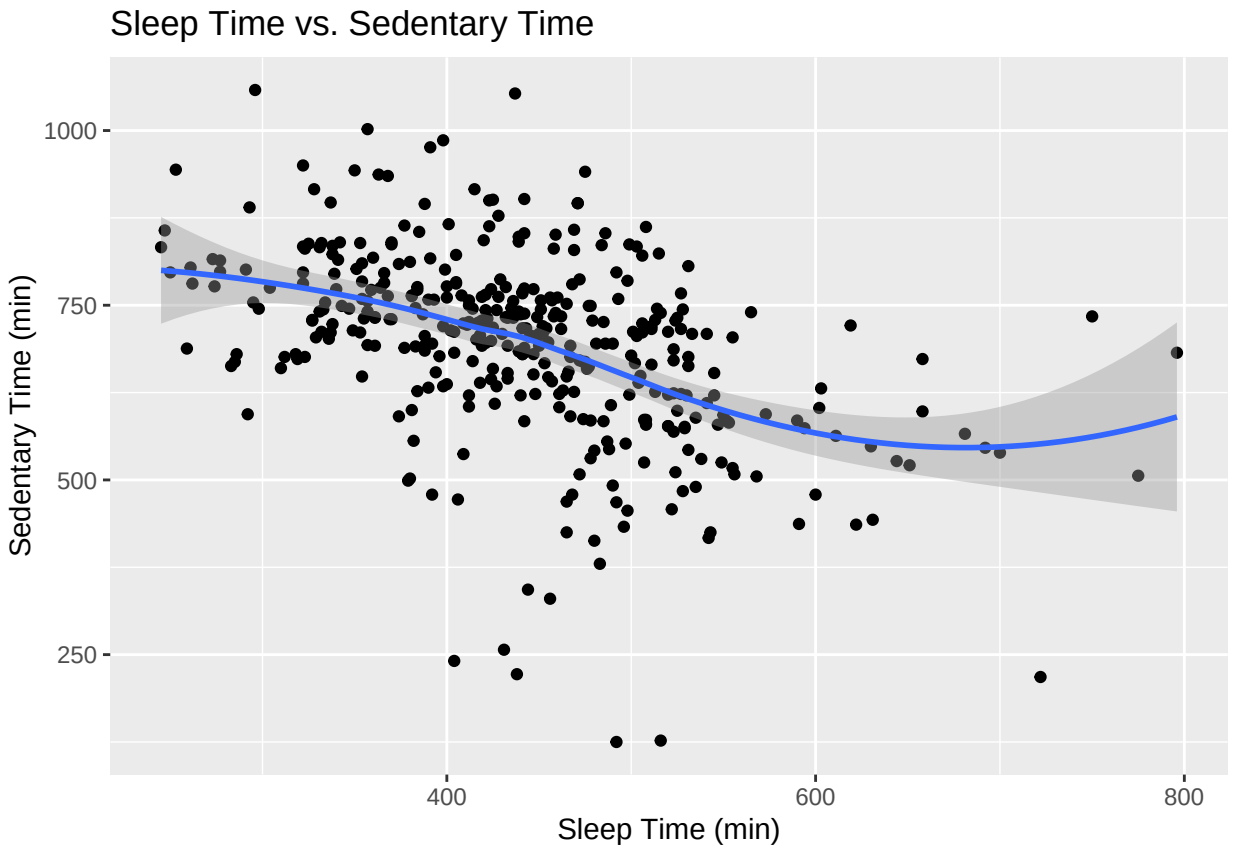
```
ggplot(data=daily_activity_merged, aes(x=minutes_asleep, y=very_active_min+fairly_active_min+lightly_ac
  geom_point() + geom_smooth() + labs(title="Sleep Time vs. Active Time",x="Sleep Time (min)" ,y="Activ
```

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




```
ggplot(data=daily_activity_merged, aes(x=minutes_asleep, y=sedentary_min)) +
  geom_point() + geom_smooth() + labs(title="Sleep Time vs. Sedentary Time",x="Sleep Time (min)" ,y="Se
```

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



From these plots we can observe that users who sleep over 10 hours are less active and a bit more sedentary. Sleeping under 10 hours does not present a direct relation with the active time but it seems that the sedentary lifestyle decreases when users sleep up to 10 hours.

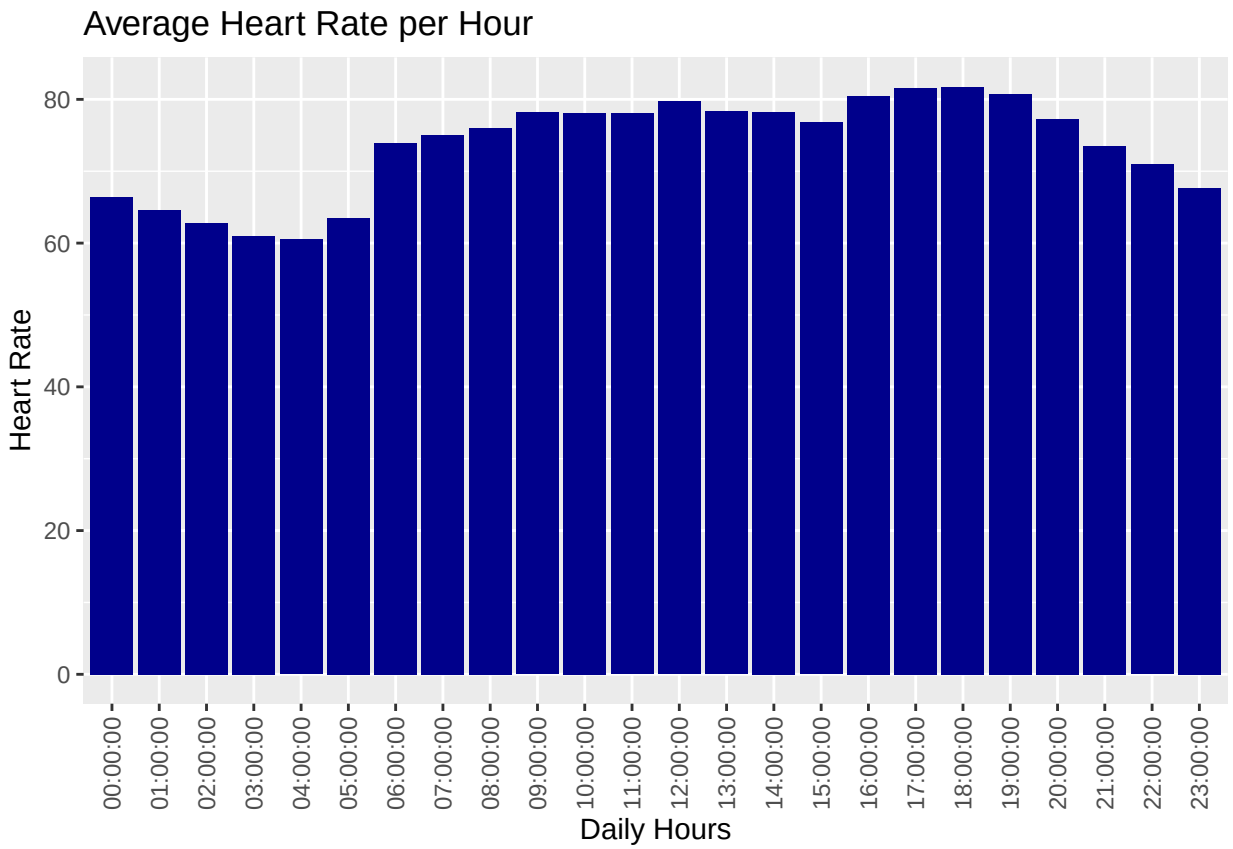
Let's take a look at the heart rate statistics:

```
heartrate_hour <- heartrate_clean %>%
   mutate(time=format(strptime(time,"%H:%M:%S"),'%H:00:00')) %>%
   group_by(id, date, time) %>%
   summarize(rate_value=mean(rate_value))
```

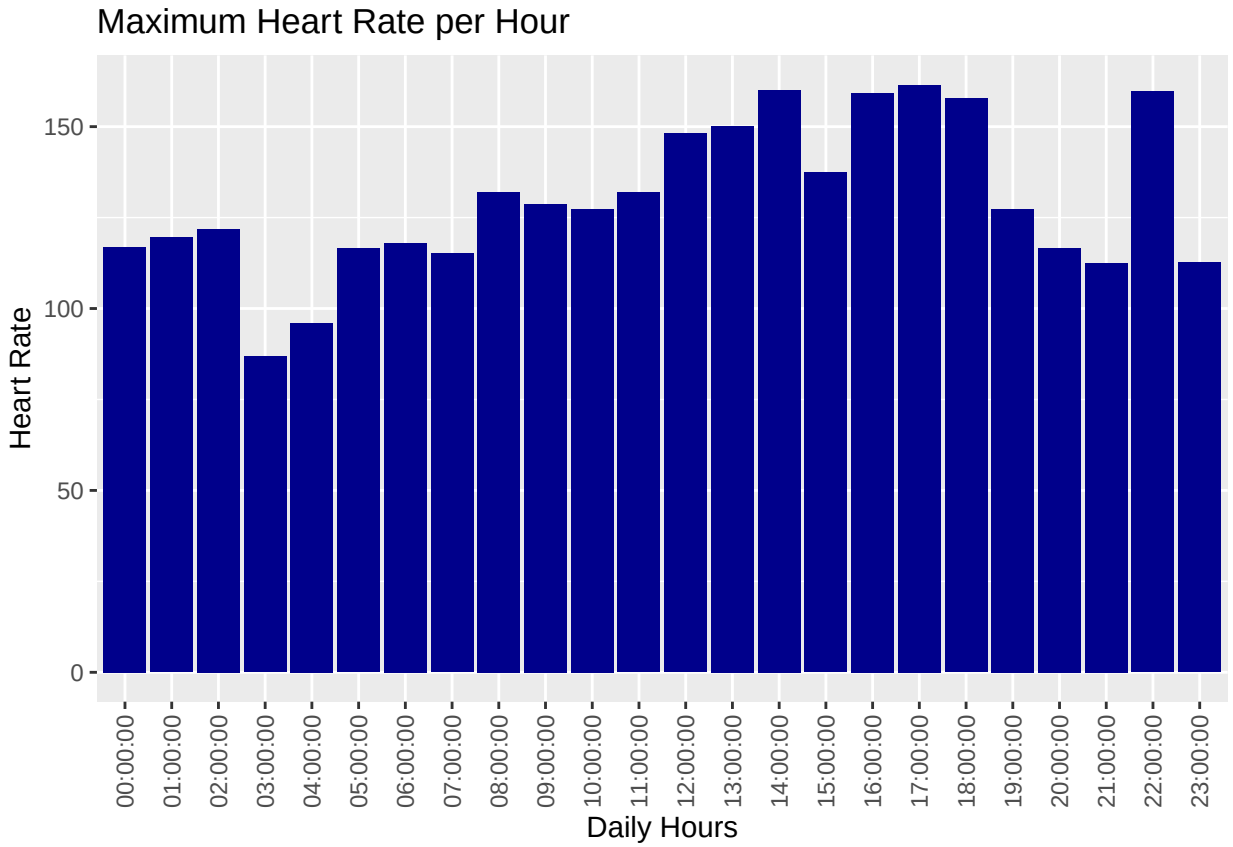
```
## `summarise()` has grouped output by 'id', 'date'. You can override using the `.groups` argument.
```

```
summary_heartrate_hour <- heartrate_hour %>%
   group_by(time) %>%
   summarize(avg_rate = mean(rate_value),min_rate = min(rate_value), max_rate = max(rate_value))
```

```
ggplot(data=summary_heartrate_hour, aes(x=time,y=avg_rate)) +
  geom_bar(stat="identity", fill='darkblue') + labs(title="Average Heart Rate per Hour",x="Daily Hours"
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
ggplot(data=summary_heartrate_hour, aes(x=time,y=max_rate)) +
  geom_bar(stat="identity", fill='darkblue') + labs(title="Maximum Heart Rate per Hour",x="Daily Hours"
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



The average Heart Rate makes sense, as it presents lower values for the sleeping hours (from 23:00 to 5:00). The higher maximum rates appear during the morning (from 12:00 to 14:00), the afternoon (from 16:00 to 18:00) and at the evening (22:00) which can be the time ranges at which the users exercise.

Let's now merge the daily average heart rates with the activity table.

```
heartrate_daily <- heartrate_clean %>%
   group_by(id,date) %>%
   summarize(max_rate = max(rate_value),min_rate = min(rate_value), mean_rate = mean(rate_value))
```

```
## `summarise()` has grouped output by 'id'. You can override using the `.groups` argument.
```

```
daily_merged <- merge(daily_activity_merged, heartrate_daily , by=c('id', 'date'))
head(daily_merged)
```

```
##           id     date sleep_records minutes_asleep time_bed total_steps
## 1 2026352035 04/17/16             1            437      498         838
## 2 2026352035 04/25/16             1            506      531        6017
## 3 2026352035 05/02/16             1            511      543        7018
## 4 2026352035 05/09/16             1            531      556       10685
## 5 2347167796 04/13/16             1            467      531       10352
## 6 2347167796 04/14/16             1            445      489       10129
##   total_distance very_active_min fairly_active_min lightly_active_min
## 1           0.52               0                 0                 60
## 2           3.73               0                 0                260
## 3           4.35               0                 0                355
## 4           6.62               0                 0                401
## 5           7.01              19                32                195
## 6           6.70               1                48                206
```

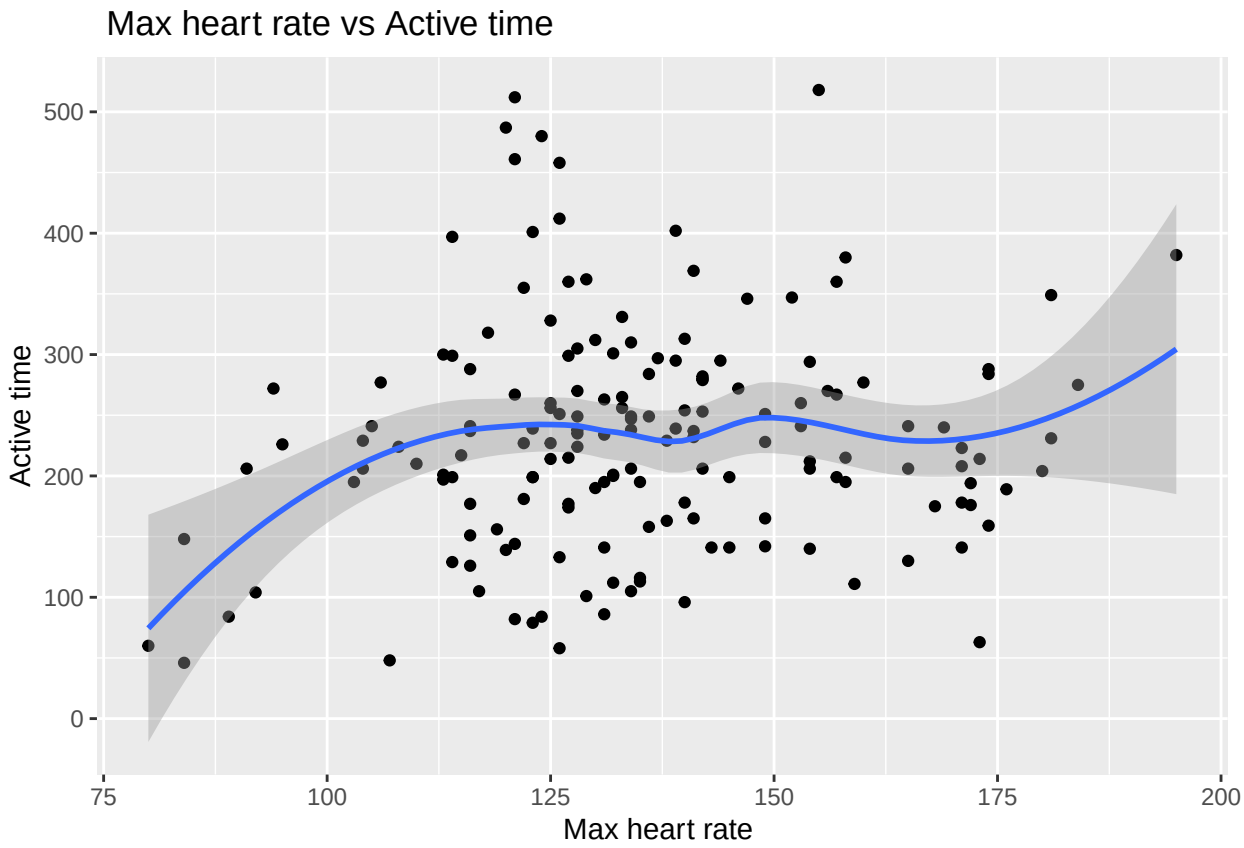
```
##   sedentary_min calories max_rate min_rate mean_rate
## 1          1053     1214       80       63  68.65625
## 2           821     1576      125       70  99.50581
## 3           716     1690      122       70  84.13457
## 4           543     1869      123       70  98.23390
## 5           676     2038      158       55  73.81290
## 6           705     2010      154       52  72.57948
```

```
ggplot(data=daily_merged, aes(x=max_rate, y=calories)) +
  geom_point() + geom_smooth() + labs(title=" Max heart rate vs calories",x="Max heart rate " ,y="calor
```

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


```
ggplot(data=daily_merged, aes(x=max_rate, y=lightly_active_min
)) +
  geom_point() + geom_smooth() + labs(title=" Max heart rate vs Active time",x="Max heart rate " ,y="Ac
```

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



It seems that there exists a relationship between the heart rate and consumed calories. Users consume more calories these days that have higher maximum heart rates, which is also related with the activity.

Let's finally analyse which percentage of the day users usually spend for each type of activity.

```
daily_activity_summary <- daily_merged %>%
   summarise(time_in_bed = mean(time_bed), sedentary_minutes = mean(sedentary_min), lightly_active_minu
```

```
daily_summary_long <- daily_activity_summary*100/(daily_activity_summary$time_in_bed + daily_activity_su
```

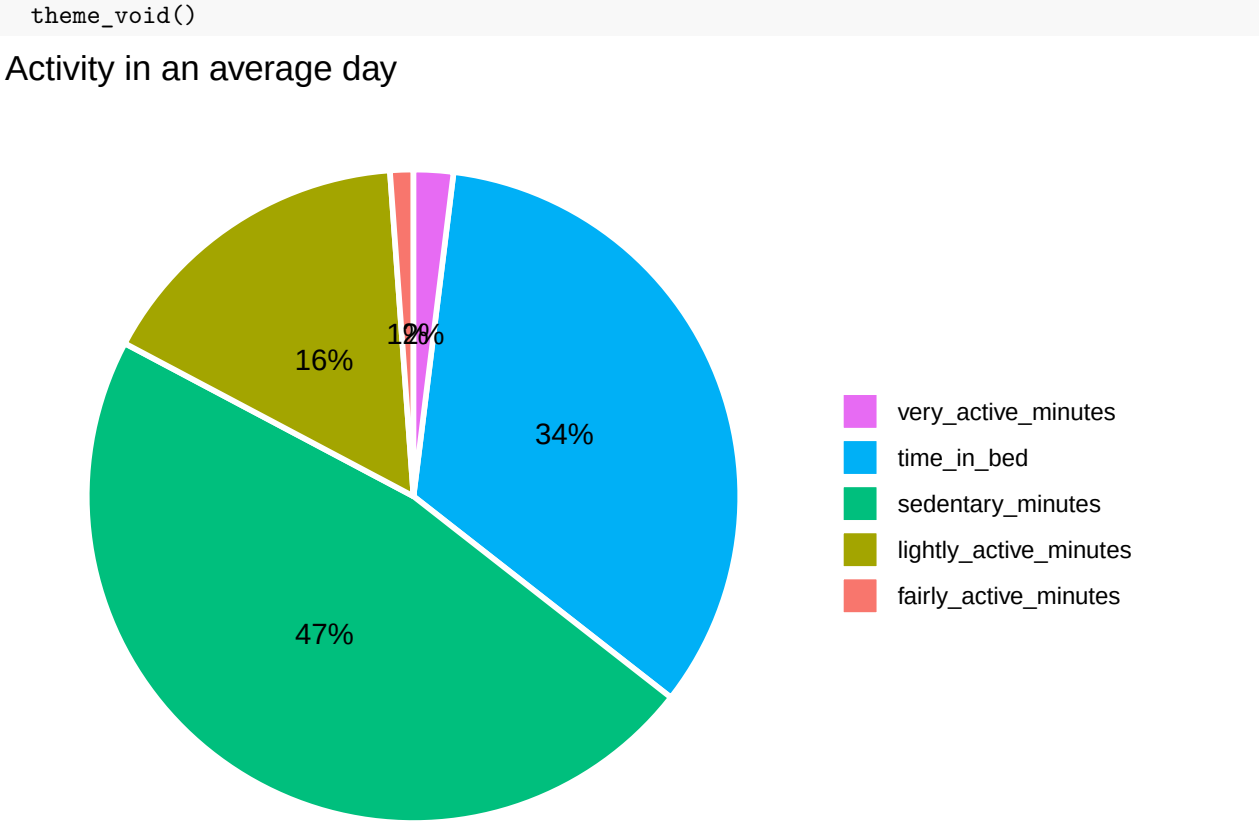
```
gather(daily_summary_long)
```

```
##                       key     value
## 1            time_in_bed 33.597678
## 2      sedentary_minutes 47.192688
## 3 lightly_active_minutes 16.095865
## 4  fairly_active_minutes  1.152028
## 5    very_active_minutes  1.961741
```

```
daily_summ_long <- gather(daily_summary_long)
```

```
ggplot(daily_summ_long, aes(x="", y=value, fill=factor(key))) +
  geom_bar(width = 1, size = 1, color = "white", stat = "identity") +
  coord_polar("y", start=0) +
geom_text(aes(label = paste0(round(value), "%")),
              position = position_stack(vjust = 0.5)) +
    labs(x = NULL, y = NULL, fill = NULL,
         title = "Activity in an average day") +
  guides(fill = guide_legend(reverse = TRUE)) +
```

```
  theme_void()
```



On average, users spend 47% of the day doing sedentary activities and a 34% in bed, which leaves only 19% of real activity time.

We can check that the sum of the different activities is around 24 hours.

On average, users spend 47% of the day doing sedentary activities and a 34% in bed, which leaves only 19% of real activity time.

6. Act

Now on the basis of my data analysis I have found that :-

Users sleep on average 8 hours a day in night hours, between 22:00 and 6:00, which is the time range at which they are more relaxed based on their heart rate. Users seem to exercise more in the afternoon, which could be because they work/study in the morning. Users walk an average of 8000 steps and 6 Km per day, which is recommended for a quite active lifestyle. Nevertheless, users spend on average 47% of the day doing sedentary activities. From this bullet points, we can construct an average profile of FitBit users:

It seems that the average users are adult people, who work or study in static positions (which implies sitting a lot of hours), and exercise in their free time.

Bellabeat marketing strategy can be focused on showing to women the advantages of having knowledge of its healthy lifestyle.

- Walking influences daily activity and calories consumed, so trying to reach a goal or having low activity alarms can help improve these good habits.
- Sleeping between 7-10 hours results in a more active day, so knowing the sleeping habits can help redirect them to be more efficient.
- High heart rates, which are not related with high intensity activities may be a sign of stress or anxiety, which can be alerted with Bellabeat products.