Bellabeat Smart Device Fitness Data Analysis

Abeer

2022-12-19

Business task:

Analyze non Bellabeat smart device usage data to identify trends. Then, using this information, make high-level recommendations for how these trends can inform Bellabeat marketing strategy.

Data Source:

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 Data contains 18 files which are written in long format.

Data Limitations:

Bellabeats products are for women and the Fitbit data doesn't specify gender, there is no demographic information and the data is limited (30 users only) therefor there could be a sampling bias.

Setting up the environment

setting up my R environment by loading the following packages:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readr)
library(janitor)
```

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

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

Importing the data

after inspecting the data in excel it appears that the `dailyCalories_merged.csv`, `dailyIntensities_merged.csv`, `dailySteps_merged.csv` have been merged into `dailyActivity_merged.csv`. So, I will only use the following:

```
daily_activity <- read.csv("/Users/Abeer/Desktop/Google Data Analytics/Bellabeat-Smart-Device-Fitness-Da
daily_sleep <- read.csv("/Users/Abeer/Desktop/Google Data Analytics/Bellabeat-Smart-Device-Fitness-Data-
weight <- read.csv("/Users/Abeer/Desktop/Google Data Analytics/Bellabeat-Smart-Device-Fitness-Data-Analy
hourly_steps <- read.csv("/Users/Abeer/Desktop/Google Data Analytics/Bellabeat-Smart-Device-Fitness-Data
hourly_intensities <- read.csv("/Users/Abeer/Desktop/Google Data Analytics/Bellabeat-Smart-Device-Fitnes
```

Data cleaning

Take a look at the data & clean the columns names

```
daily_activity <- daily_activity %>% clean_names()
daily_sleep <- daily_sleep %>% clean_names()
weight <- weight %>% clean_names()
hourly_steps <- hourly_steps %>% clean_names()
hourly_intensities <- hourly_intensities %>% clean_names()
head(daily_activity)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 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
##   logged_activities_distance very_active_distance moderately_active_distance
## 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
##   light_active_distance sedentary_active_distance very_active_minutes
## 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
##   fairly_active_minutes lightly_active_minutes sedentary_minutes 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(daily_sleep)
```

```
##           id             sleep_day total_sleep_records total_minutes_asleep
## 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
##   total_time_in_bed
## 1               346
## 2               407
## 3               442
## 4               367
## 5               712
## 6               320
```

```
head(weight)
```

```
##           id                  date weight_kg weight_pounds 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
##   is_manual_report       log_id
## 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(hourly_steps)
```

```
##           id         activity_hour step_total
## 1 1503960366 4/12/2016 12:00:00 AM        373
## 2 1503960366  4/12/2016 1:00:00 AM        160
## 3 1503960366  4/12/2016 2:00:00 AM        151
## 4 1503960366  4/12/2016 3:00:00 AM          0
## 5 1503960366  4/12/2016 4:00:00 AM          0
## 6 1503960366  4/12/2016 5:00:00 AM          0
```

```
head(hourly_intensities)
```

```
##           id         activity_hour total_intensity average_intensity
## 1 1503960366 4/12/2016 12:00:00 AM              20          0.333333
## 2 1503960366  4/12/2016 1:00:00 AM               8          0.133333
## 3 1503960366  4/12/2016 2:00:00 AM               7          0.116667
## 4 1503960366  4/12/2016 3:00:00 AM               0          0.000000
## 5 1503960366  4/12/2016 4:00:00 AM               0          0.000000
## 6 1503960366  4/12/2016 5:00:00 AM               0          0.000000
```

how many users in each dataset?

```
n_distinct(daily_activity$id)
```

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

```
n_distinct(daily_sleep$id)
```

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

```
n_distinct(weight$id) # only 8 users therefore i will not use it
```

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

```
n_distinct(hourly_steps$id)
```

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

check for missing and duplicate observations

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

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

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

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

```
sum(is.na(hourly_steps))
```

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

```
sum(is.na(hourly_intensities))
```

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

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

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

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

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

```
sum(duplicated(hourly_steps))
```

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

```
sum(duplicated(hourly_intensities))
```

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

remove duplicates from `daily_sleep`

```
daily_sleep <- daily_sleep %>% distinct()
sum(duplicated(daily_sleep))
```

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

we can see that the data type for date columns is char

```
str(daily_activity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ id                        : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_date             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ total_steps               : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ total_distance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ tracker_distance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ logged_activities_distance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_distance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ moderately_active_distance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ light_active_distance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ sedentary_active_distance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_minutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ fairly_active_minutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ lightly_active_minutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ sedentary_minutes         : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ calories                  : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(daily_sleep)
```

```
## 'data.frame':    410 obs. of  5 variables:
##  $ id                  : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ sleep_day           : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM
##  $ total_sleep_records : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ total_minutes_asleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ total_time_in_bed   : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
str(hourly_steps)
```

```
## 'data.frame':    22099 obs. of  3 variables:
##  $ id           : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_hour: chr  "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/20
##  $ step_total   : int  373 160 151 0 0 0 0 0 250 1864 ...
```

```
str(hourly_intensities)
```

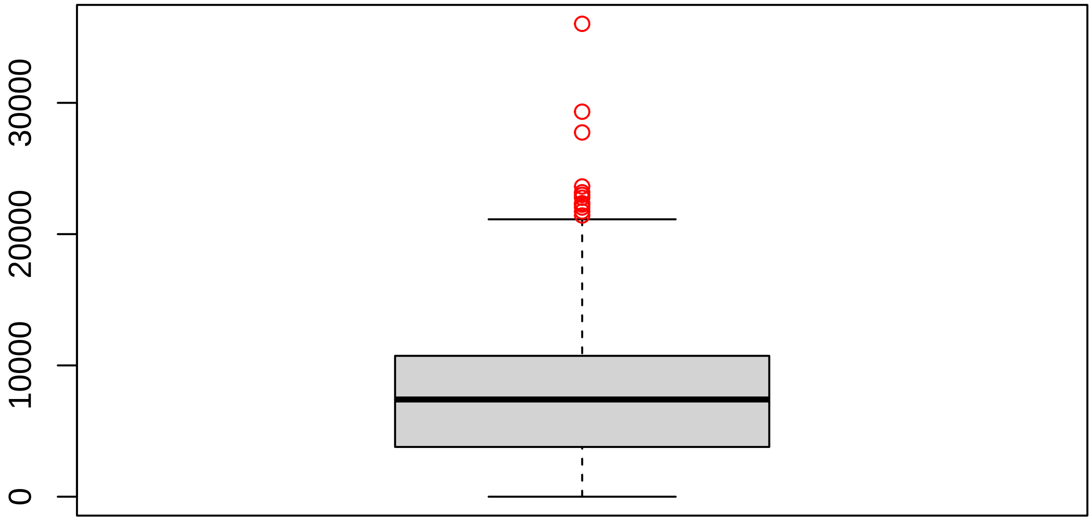
```
## 'data.frame':    22099 obs. of  4 variables:
##  $ id               : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_hour    : chr  "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/1
##  $ total_intensity  : int  20 8 7 0 0 0 0 0 13 30 ...
##  $ average_intensity: num  0.333 0.133 0.117 0 0 ...
```

fix the date columns format (i will ignore the time in sleep_day since all observations are 12:00:00)

```
daily_activity$activity_date <- as.Date(daily_activity$activity_date, "%m/%d/%y")

daily_sleep$sleep_day <- as.Date(daily_sleep$sleep_day, "%m/%d/%y")

hourly_steps$activity_hour <- strptime(hourly_steps$activity_hour, "%m/%d/%Y %I:%M:%S %p")
hourly_steps$hour <- strftime(hourly_steps$activity_hour, "%H:%M")

hourly_intensities$activity_hour <- strptime(hourly_intensities$activity_hour, "%m/%d/%Y %I:%M:%S %p")
hourly_intensities$hour <- strftime(hourly_steps$activity_hour, "%H:%M")
```

To identify outliers i will create a boxplot then i will use the IQR method to remove outliers, since the dataset is small and I'm not sure it's representative of the population of interest i decided to be more conservative and remove only extreme outliers

```
boxplot(daily_activity$total_steps, outcol="red")
```



IQR method

```
#daily activity total steps
summary(daily_activity$total_steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3790    7406    7638   10727   36019
```

```
IQR(daily_activity$total_steps)
```

```
## [1] 6937.25
```

```
Tmin = 3790 - (3 * 6937.25)
Tmax = 10727 + (3 * 6937.25)

#outliers
daily_activity$total_steps[which(daily_activity$total_steps < Tmin | daily_activity$total_steps > Tmax)]
```

```
## [1] 36019
```

```
#remove outliers
daily_activity <- daily_activity[(daily_activity$total_steps > Tmin & daily_activity$total_steps < Tmax)
```

Analysis

```
#daily usage
```

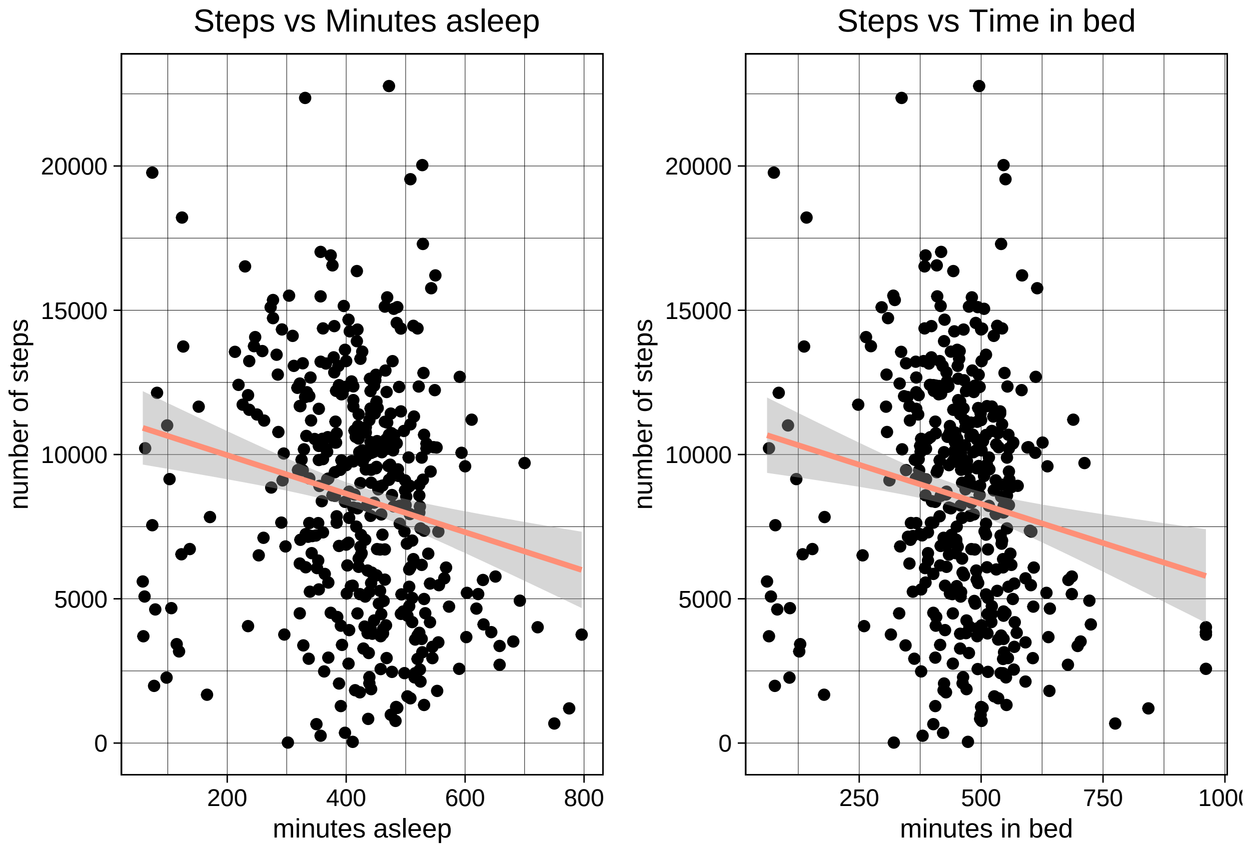
```
#minutes asleep vs steps

#first i will use rowSums function to sum across rows and create total_intensities_distance column
daily_activity <- daily_activity %>% mutate(total_intensities_distance = rowSums(across(c(light_active_d
#now i will inner_join daily_sleep and daily_activity to create the plot
daily_sleep <- inner_join(daily_sleep, daily_activity[ , c("id", "activity_date", "total_steps", "seden

g1 <- ggplot(data = daily_sleep, mapping = aes(x = total_minutes_asleep, y = total_steps)) + geom_point
         x = "minutes asleep", y = "number of steps") +
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))

#steps vs time in bed
g2 <- ggplot(data = daily_sleep,mapping = aes(x= total_time_in_bed, y = total_steps)) + geom_point() + g
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))




g1grob <- ggplotGrob(g1)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
g2grob <- ggplotGrob(g2)
```

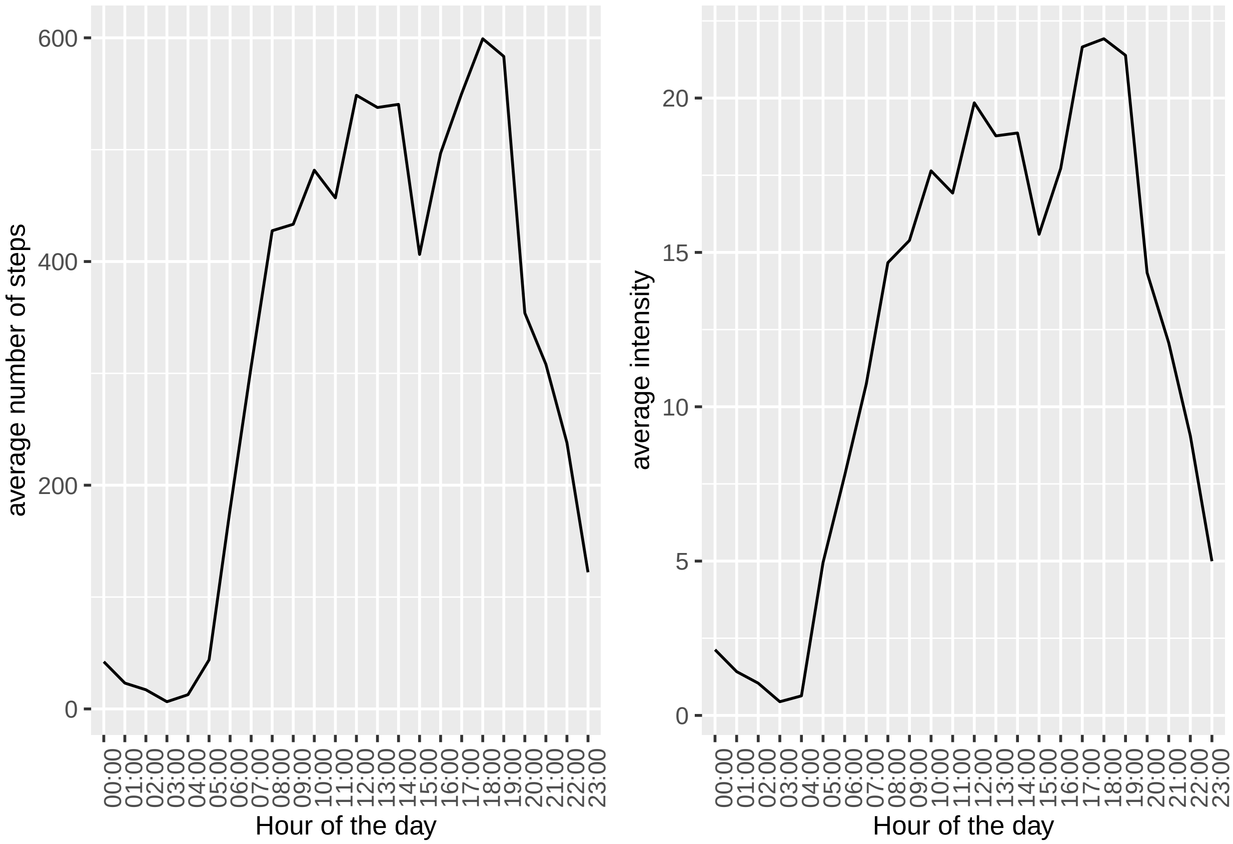
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
grid.arrange(g1grob, g2grob, nrow = 1)
```



Here i wanted to see if there is a correlation between number of steps with minutes asleep and minutes in bed, we can see that the data points follow no direction. This means there is no correlation.

```
#Steps vs time of the day
steps_time_trends <- hourly_steps %>% group_by(hour) %>%
  summarise(avg_steps_per_hour = mean(step_total)) %>%
  arrange(hour)

g3 <- ggplot(data = steps_time_trends, mapping = aes(x= hour, y = avg_steps_per_hour, group = 1 )) + ge
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))

#intensities vs time of the day
intensity_time_trends <- hourly_intensities %>% group_by(hour) %>%
  summarise(avg_intensity_per_hour = mean(total_intensity)) %>%
  arrange(hour)

g4 <- ggplot(data = intensity_time_trends, mapping = aes(x= hour, y= avg_intensity_per_hour, group = 1)
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))

g3grob <- ggplotGrob(g3)
g4grob <- ggplotGrob(g4)

grid.arrange(g3grob, g4grob, nrow = 1)
```



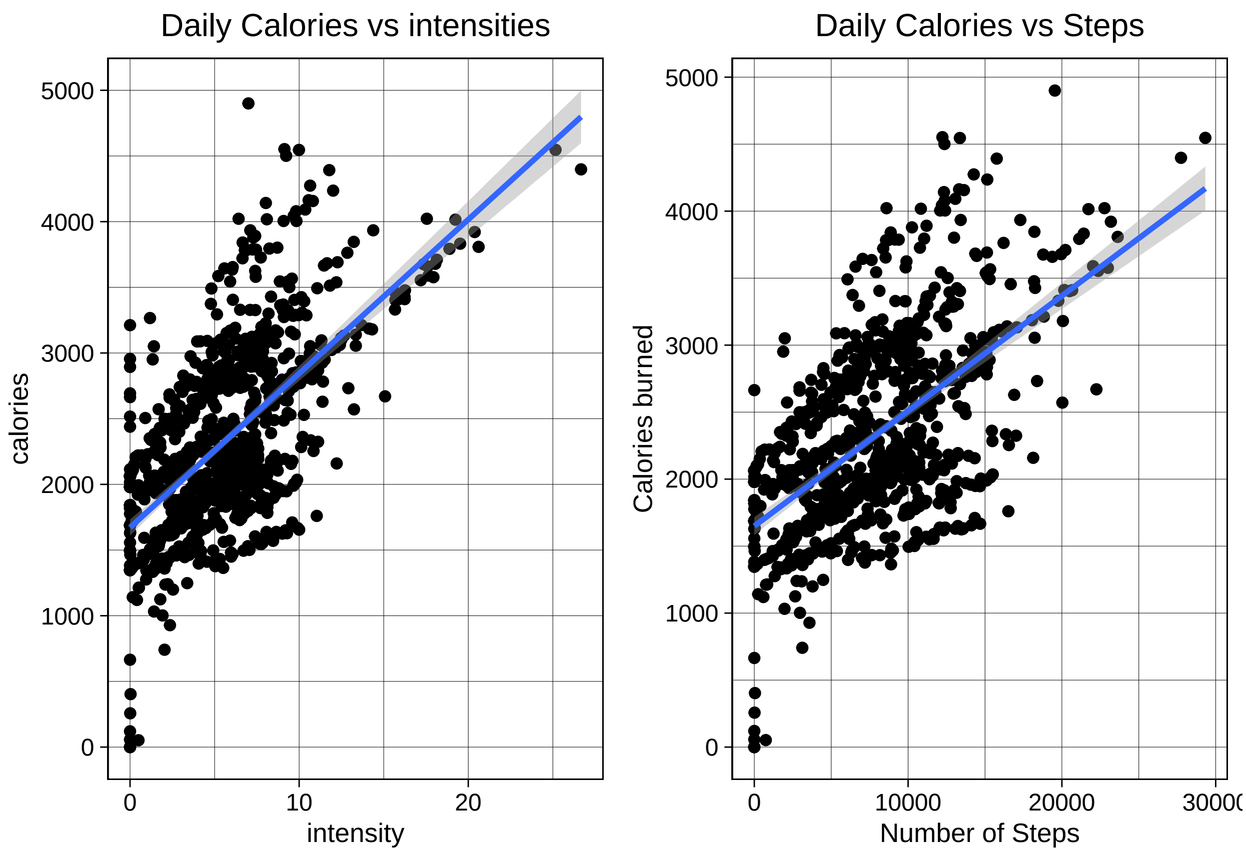
```
#intensities vs calories
g5 <- ggplot(data = daily_activity, mapping = aes(x = total_intensities_distance, y = calories)) + geom_
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))


#daily steps vs calories
g6 <- ggplot(data = daily_activity, mapping = aes(x = total_steps, y = calories)) + geom_point() + geom_
         x = "Number of Steps", y = "Calories burned") +
    theme(plot.title = element_text(size = 12, hjust = 0.5),
          axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))
g5grob <- ggplotGrob(g5)
```

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

```
g6grob <- ggplotGrob(g6)
```

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

```
grid.arrange(g5grob, g6grob, nrow = 1)
```



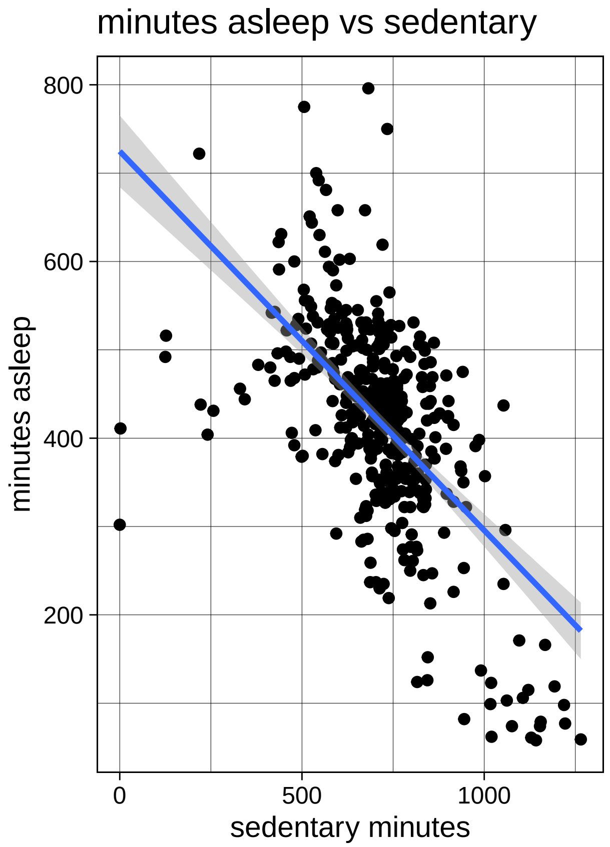
```
# minutes asleep vs sedentary
g7 <- ggplot(data = daily_sleep, mapping = aes(x = sedentary_minutes, y = total_minutes_asleep)) + geom_

#intensities vs sleep

g8 <- ggplot(data = daily_sleep, mapping = aes(x = total_intensities_distance, y = total_minutes_asleep)

g7grob <- ggplotGrob(g7)
```

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

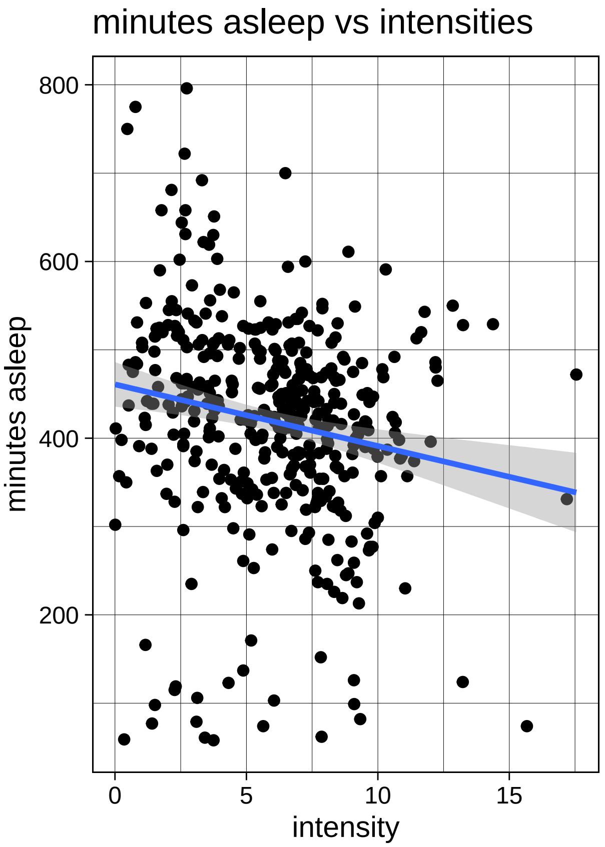
```
g8grob <- ggplotGrob(g8)
```

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

```
grid.arrange(g7grob, g8grob, nrow = 1)
```



minutes asleep vs sedentary



minutes asleep vs intensities

Conclusion