

Case_Study_2_Bellabeat

Jade Nguyen

2022-06-27

R Markdown

1 Load necessary packages

```
library(tidyverse)
```

```
## — Attaching packages ————— tidyverse 1.3.1 —
```

```
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4  
## ✓ tibble  3.1.7      ✓ dplyr   1.0.9  
## ✓ tidyr   1.2.0      ✓ stringr 1.4.0  
## ✓ readr   2.1.2      ✓ forcats 0.5.1
```

```
## — Conflicts ————— tidyverse_conflicts() —
```

```
## ✗ dplyr::filter() masks stats::filter()  
## ✗ dplyr::lag()     masks stats::lag()
```

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

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

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

```
library(dplyr)  
library(lubridate)
```

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

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

2 Load data

```

setwd("Dataset_Bellabeat")

weight_log <- read.csv("weightLogInfo_merged.csv")
sleep_day <- read.csv("sleepDay_merged.csv")
minute_steps_wide <- read.csv("minuteStepsWide_merged.csv")
minute_steps_narrow <- read.csv("minuteStepsNarrow_merged.csv")
minute_sleep <- read.csv("minuteSleep_merged.csv")
minute_METs_narrow <- read.csv("minuteMETsNarrow_merged.csv")
minute_intensities_wide <- read.csv("minuteIntensitiesWide_merged.csv")
minute_intensities_narrow <- read.csv("minuteIntensitiesNarrow_merged.csv")
minute_calories_wide <- read.csv("minuteCaloriesWide_merged.csv")
minute_calories_narrow <- read.csv("minuteCaloriesNarrow_merged.csv")
hourly_steps <- read.csv("hourlySteps_merged.csv")
hourly_intensities <- read.csv("hourlyIntensities_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
heartrate_seconds <- read.csv("heartrate_seconds_merged.csv")
daily_steps <- read.csv("dailySteps_merged.csv")
daily_intensities <- read.csv("dailyIntensities_merged.csv")
daily_calories <- read.csv("dailyCalories_merged.csv")
daily_activity <- read.csv("dailyActivity_merged.csv")

```

3 The sqldf package is loaded to use SQL syntax to determine if the values of daily_calories, daily_intensities, and daily_steps are contained in daily_activity.

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
check1_daily_activity <- daily_activity %>% select (Id,ActivityDate,Calories)
head(check1_daily_activity)
```

```
##           Id ActivityDate  Calories
## 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
```

```
sql_check_calories <- sqldf('SELECT * FROM daily_calories INTERSECT SELECT * FROM daily_calor
ies')
head(sql_check_calories)
```

```
##           Id ActivityDay Calories
## 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
```

```
check2_daily_activity <- daily_activity %>% select (Id,ActivityDate,TotalSteps)
head(check2_daily_activity)
```

```
##           Id ActivityDate TotalSteps
## 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
```

```
sql_check_steps <- sqldf('SELECT * FROM daily_steps INTERSECT SELECT * FROM daily_steps')
head(sql_check_steps)
```

```
##           Id ActivityDay StepTotal
## 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
```

Use data from daily_activity to analyse calories, intensities and steps in place of 3 other dfs

4 Check number of participants for each log

```
library(dplyr)
n_distinct(daily_activity$Id)
```

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

```
n_distinct(weight_log$Id)
```

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

```
n_distinct(sleep_day$Id)
```

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

```
n_distinct(minute_steps_wide$Id)
```

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

```
n_distinct(minute_steps_narrow$Id)
```

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

```
n_distinct(minute_sleep$Id)
```

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

```
n_distinct(minute_METs_narrow$Id)
```

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

```
n_distinct(minute_intensities_wide$Id)
```

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

```
n_distinct(minute_intensities_narrow$Id)
```

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

```
n_distinct(heartrate_seconds$Id)
```

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

Users mainly use their device to track steps, calories, and sleep. Tracking `weight_log` and `heartrate_seconds` have low number of participants so it's not very reliable for analysis

5 The `summary()` function is used to pull key statistics about the data frames.

```
daily_activity %>%select(TotalSteps, TotalDistance, VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance, SedentaryActiveDistance, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes, Calories) %>%summary()
```

```
##      TotalSteps      TotalDistance      VeryActiveDistance ModeratelyActiveDistance
## Min.      :    0      Min.      : 0.000      Min.      : 0.000      Min.      :0.0000
## 1st Qu.: 3790      1st Qu.: 2.620      1st Qu.: 0.000      1st Qu.:0.0000
## Median : 7406      Median : 5.245      Median : 0.210      Median :0.2400
## Mean      : 7638      Mean      : 5.490      Mean      : 1.503      Mean      :0.5675
## 3rd Qu.:10727      3rd Qu.: 7.713      3rd Qu.: 2.053      3rd Qu.:0.8000
## Max.      :36019      Max.      :28.030      Max.      :21.920      Max.      :6.4800
## LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## Min.      : 0.000      Min.      :0.000000      Min.      : 0.00
## 1st Qu.: 1.945      1st Qu.:0.000000      1st Qu.: 0.00
## Median : 3.365      Median :0.000000      Median : 4.00
## Mean      : 3.341      Mean      :0.001606      Mean      : 21.16
## 3rd Qu.: 4.782      3rd Qu.:0.000000      3rd Qu.: 32.00
## Max.      :10.710      Max.      :0.110000      Max.      :210.00
## FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes      Calories
## Min.      : 0.00      Min.      : 0.0      Min.      : 0.0      Min.      : 0
## 1st Qu.: 0.00      1st Qu.:127.0      1st Qu.: 729.8      1st Qu.:1828
## Median : 6.00      Median :199.0      Median :1057.5      Median :2134
## Mean      : 13.56      Mean      :192.8      Mean      : 991.2      Mean      :2304
## 3rd Qu.: 19.00      3rd Qu.:264.0      3rd Qu.:1229.5      3rd Qu.:2793
## Max.      :143.00      Max.      :518.0      Max.      :1440.0      Max.      :4900
```

Average steps: 7638, Average distance: 5490, Sedentary minutes: 991.2, Calories: 2304, VeryActiveMinutes: 21.16

```
minute_METs_narrow %>% select(METs) %>% summary()
```

```
##      METs
## Min.      : 0.00
## 1st Qu.: 10.00
## Median : 10.00
## Mean      : 14.69
## 3rd Qu.: 11.00
## Max.      :157.00
```

Average MET: 14.69

```
sleep_day %>% select(TotalMinutesAsleep,      TotalTimeInBed) %>% summary()
```

```
##      TotalMinutesAsleep TotalTimeInBed
## Min.      : 58.0      Min.      : 61.0
## 1st Qu.:361.0      1st Qu.:403.0
## Median :433.0      Median :463.0
## Mean      :419.5      Mean      :458.6
## 3rd Qu.:490.0      3rd Qu.:526.0
## Max.      :796.0      Max.      :961.0
```

Average TotalMinutesAsleep: 419.5, Average TotalTimeInBed:458.6

```
heartrate_seconds %>% select(Value) %>% summary()
```

```
##      Value
## Min.   : 36.00
## 1st Qu.: 63.00
## Median : 73.00
## Mean   : 77.33
## 3rd Qu.: 88.00
## Max.   :203.00
```

Average: 77.33

```
weight_log %>% select(WeightKg,WeightPounds,BMI) %>% summary ()
```

```
##      WeightKg      WeightPounds      BMI
## Min.   : 52.60   Min.   :116.0   Min.   :21.45
## 1st Qu.: 61.40   1st Qu.:135.4   1st Qu.:23.96
## Median : 62.50   Median :137.8   Median :24.39
## Mean   : 72.04   Mean   :158.8   Mean   :25.19
## 3rd Qu.: 85.05   3rd Qu.:187.5   3rd Qu.:25.56
## Max.   :133.50   Max.   :294.3   Max.   :47.54
```

Average Weight(kg) : 72.04 Average Weight(pounds) :158.8 Average BMI :25.19

6 show level of activeness

6.1 Calculate total minutes across different activity levels

```
daily_activity <- daily_activity %>%
  mutate(TotalActiveMinutes = VeryActiveMinutes + FairlyActiveMinutes +
    LightlyActiveMinutes + SedentaryMinutes)
```

6.2 Calculate dailyActivityRatio

```
dailyActivityRatio <- daily_activity %>%
  summarise(sedentary=mean(SedentaryMinutes/TotalActiveMinutes),
    lightlyActive=mean(LightlyActiveMinutes/TotalActiveMinutes),
    fairlyActive=mean(FairlyActiveMinutes/TotalActiveMinutes),
    veryActive=mean(VeryActiveMinutes/TotalActiveMinutes)) %>%
  summarise(sedentary = round(sedentary*100,2), lightlyActive = round(lightlyActive*100,2), f
    airlyActive = round(fairlyActive*100,2), veryActive=round(veryActive*100,2))
```

6.3 create donut chart using plot_ly() function

```
# Load library plotly
library(plotly)
```

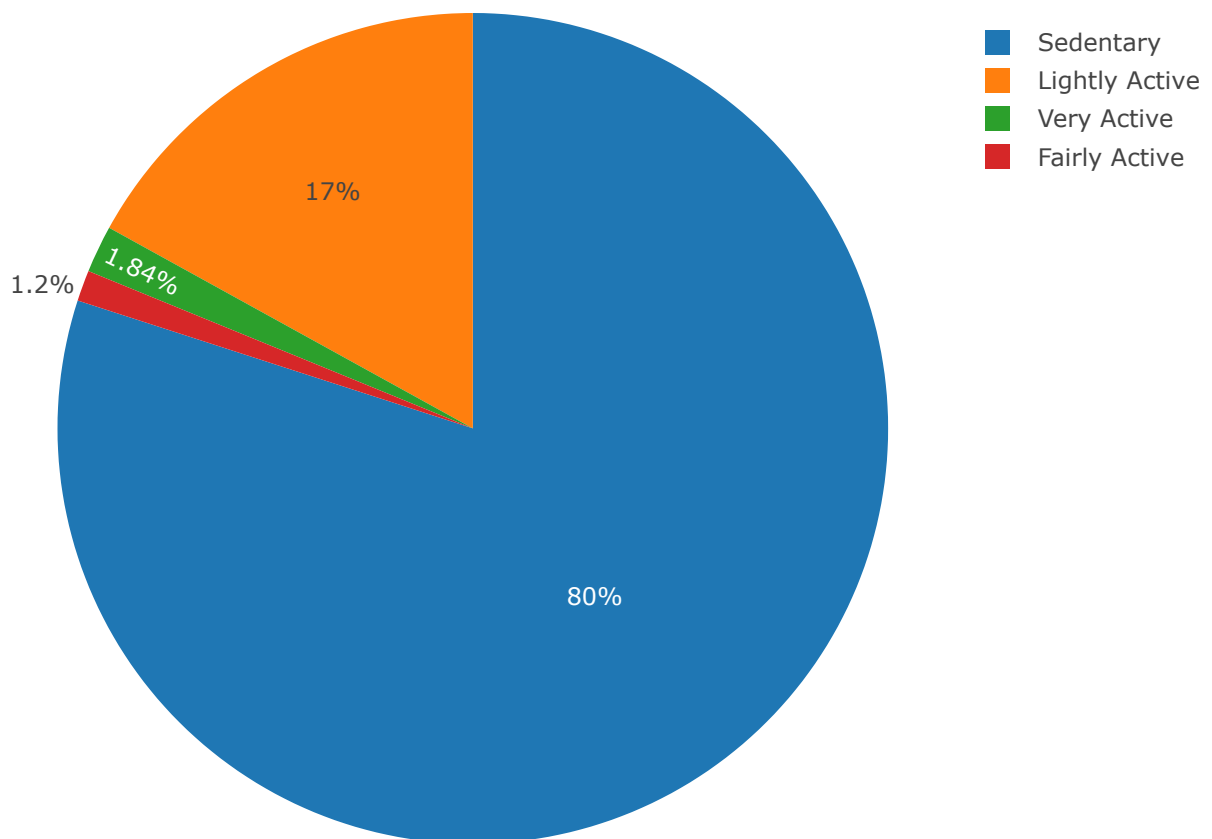
```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##      last_plot
```

```
## The following object is masked from 'package:stats':  
##  
##   filter
```

```
## The following object is masked from 'package:graphics':  
##  
##   layout
```

```
pie_dailyActivityRatio <- data.frame(group= c('Sedentary','Lightly Active', 'Fairly Active',  
                                              'Very Active'),  
                                     value= c(79.98,  
                                              16.98,  
                                              1.2,  
                                              1.84))  
  
plot_ly(pie_dailyActivityRatio) %>%  
  add_pie(pie_dailyActivityRatio, labels = ~`group`, values = ~`value`)
```

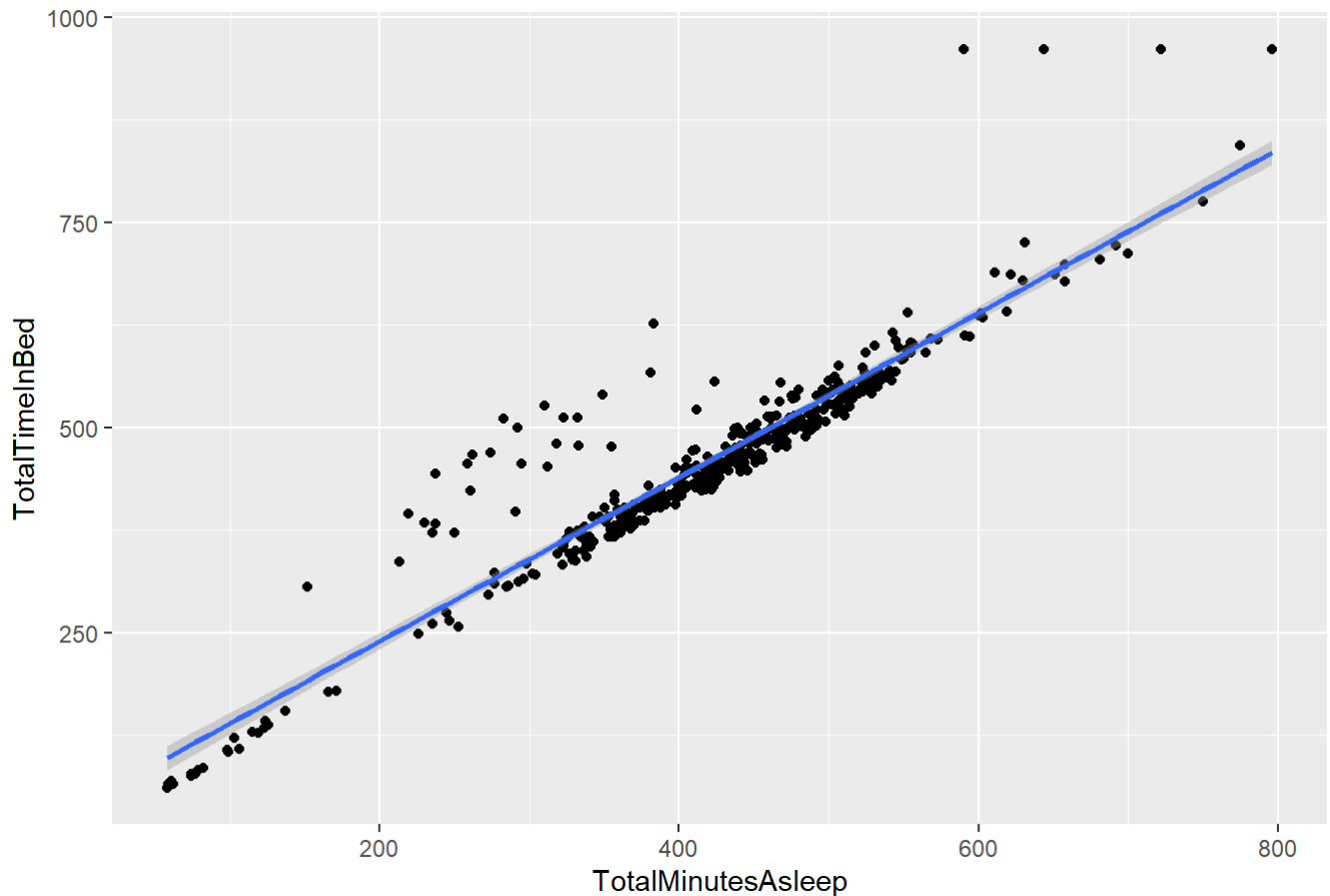


7 Visualize relationship between total Minutes Asleep and Total time in bed

```
ggplot(sleep_day, aes(x=TotalMinutesAsleep, y =TotalTimeInBed)) + geom_point() + stat_smooth  
(method=lm) + labs(title = "The Relationship Between Total Minutes Asleep and Total Time In Be  
d")
```

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

The Relationship Between Total Minutes Asleep and Total Time In Bed



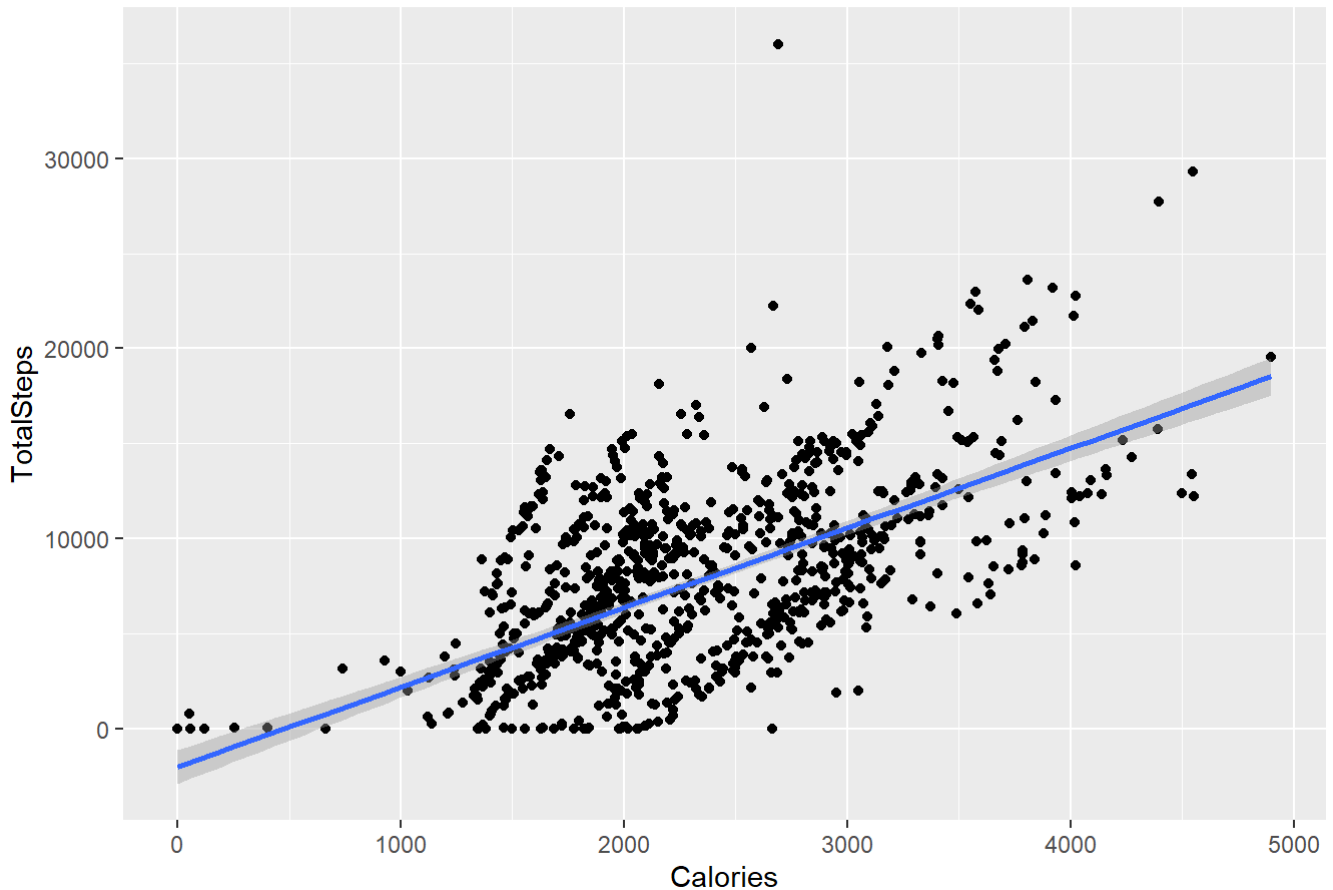
There is a clear similarity between the time participants spent asleep and the time they spent in bed.

8 Visualize relationship between total steps and Total calories burnt

```
ggplot(daily_activity, aes(x=Calories, y =TotalSteps)) + geom_point() + stat_smooth(method="lm") + labs(title = "The Relationship Between Total Steps and Calories Burnt")
```

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


The Relationship Between Total Steps and Calories Burnt



-> The more steps they take, the more calories they burn

9. Visualize relationship between total Minutes Asleep and sedantaryminutes

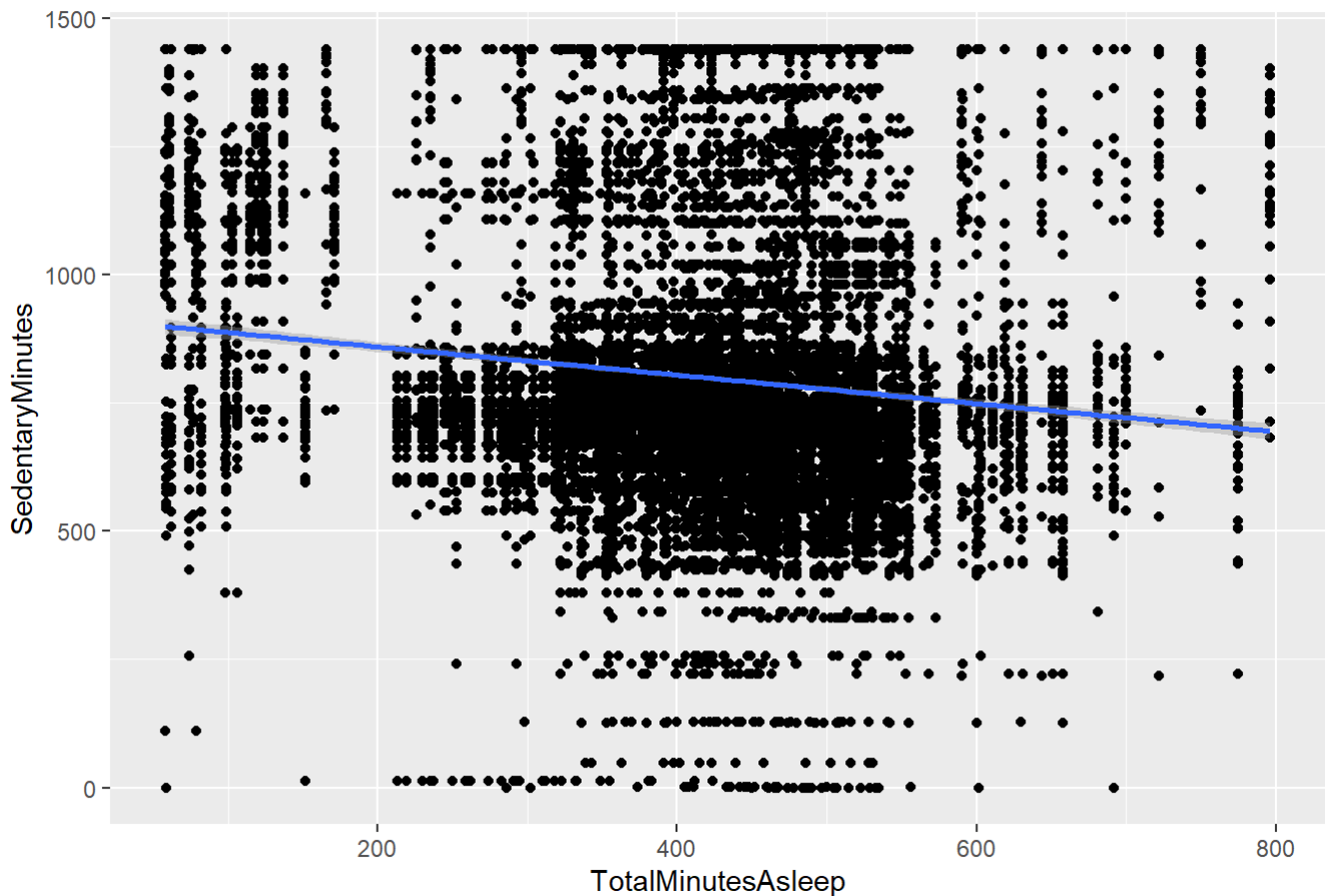
```
merged_data <- merge(daily_activity, sleep_day, by = "Id", all.x = TRUE, all.y=TRUE)
ggplot(merged_data, aes(x=TotalMinutesAsleep, y =SedantaryMinutes)) + geom_point() + stat_smooth(method=lm) + labs(title = "The Relationship Between Total Minutes Asleep and Sedantary Minutes")
```

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

```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 227 rows containing missing values (geom_point).
```

The Relationship Between Total Minutes Asleep and Sedantary Minutes



-> The more sedentary minutes they have, the less sleep they have.

10 The relationship between good BMI and active level

Filter the participants with good BMI

```
average_BMI_df <- weight_log %>% filter( BMI > 18.5 & BMI < 24.9)
merged_data_2 <- merge(daily_activity, average_BMI_df, by = "Id", all.x = TRUE, all.y=TRUE)
merged_data_2 %>% select(VeryActiveMinutes, TotalSteps) %>% summary()
```

```
## VeryActiveMinutes  TotalSteps
## Min.   : 0.00      Min.   : 0
## 1st Qu.: 0.00      1st Qu.: 5454
## Median : 13.00     Median : 9799
## Mean   : 22.11     Mean   : 8730
## 3rd Qu.: 36.00     3rd Qu.: 11835
## Max.   : 210.00    Max.   : 36019
```

On average, People with good BMI tend to walk more and be more active

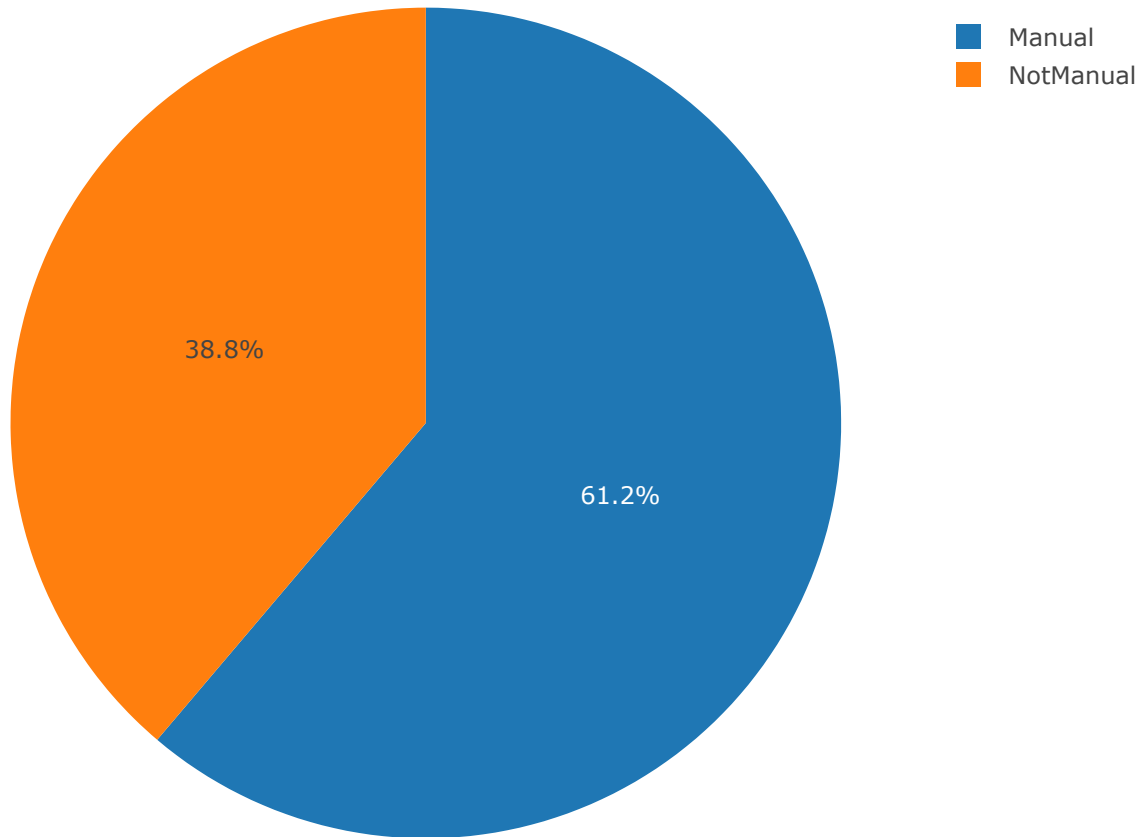
11 Check how many weight logs are manually reported

```
weight_log %>% count(IsManualReport)
```

```
## IsManualReport  n
## 1             False 26
## 2              True 41
```

Visualize data

```
pie_weight_log <- data.frame(group= c('Manual','NotManual'),  
                             value= c(41, 26))  
plot_ly(pie_weight_log) %>%  
  add_pie(pie_weight_log, labels = ~`group`, values = ~`value`)
```



61.2% weight data is manually reported, which may explain the low participation rate

12 Recommendations

- (1). Connect weight_log function with electric scales to reduce the need for manual log
- (2). Add Reminders for users to be more active
- (3). In-app articles about healthy BMI level, activity level, healthy sleep pattern
- (4). Enable goals setting and remind users to reach their goal.
- (5). Enable alert notifications if user's resting heart rate varies significantly from their normal.
- (6). Enable notifications to encourage activity if a user has been sedentary for an extended period of time.
- (7). Enable users to make friends to track each other's progress and send congratulations when reaching a goal