Correlating Fitbit activity data and sleeping patterns

Final capstone project for the Google Data Analytics Professional Certificate

Fabrizio Leo

2023-01-21

Introduction and background

Goals

The aim of this project is to analyze Fitbit usage data to find out patterns and trend in physical activity, physiological and sleep data. In particular, the main goals are:

- define and identify active and sedentary Fitbit users in a Fitbit dataset
- find out how these groups differ in terms of pattern of activity, number of steps and physiological variables such as heart rate and calories
- find out whether these groups differ in terms of sleeping habits, in particular in the amount of sleep and time in bed.

The final goal is to inform sedentary people by providing them with information that can be used by them to start an healthier and more active lifestyle.

The Data

I use the dataset publicy available at the following link:

https://www.kaggle.com/arashnic/fitbit

It includes thirty Fitbit users data collected between 03/12/2016 and 05/12/2016 including minute-level output for physical activity, heart rate and sleep monitoring. These data are organized in several csv files. Kaggle reports a usability score of 10 for this dataset.

First, I download and install some required libraries

```
#install.packages(c("lubridate","tidyverse","plotrix","rstatix","here"))

library(lubridate)
library(tidyverse)
library(plotrix)
library(rstatix)
library(here)
```

Loading your CSV files

I load a dataframe with the daily activity.

```
daily_activity <- read_csv(here("data","dailyActivity_merged.csv"))
```

```
## Rows: 940 Columns: 15
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

I also load a different csv for sleep data.

```
sleep_day <- read_csv(here("data","sleepDay_merged.csv"))
```

```
## Rows: 413 Columns: 5
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Exploring a few key tables

Take a look at the daily_activity data.

```
head(daily_activity)
```

```
## # A tibble: 6 x 15
##       Id Activ~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8 Seden~9
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~   13162    8.5     8.5        0    1.88   0.550    6.06       0
## 2 1.50e9 4/13/2~   10735    6.97    6.97       0    1.57   0.690    4.71       0
## 3 1.50e9 4/14/2~   10460    6.74    6.74       0    2.44   0.400    3.91       0
## 4 1.50e9 4/15/2~    9762    6.28    6.28       0    2.14   1.26     2.83       0
## 5 1.50e9 4/16/2~   12669    8.16    8.16       0    2.71   0.410    5.04       0
## 6 1.50e9 4/17/2~    9705    6.48    6.48       0    3.19   0.780    2.51       0
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>, and abbreviated variable names
## #   1: ActivityDate, 2: TotalSteps, 3: TotalDistance, 4: TrackerDistance,
## #   5: LoggedActivitiesDistance, 6: VeryActiveDistance,
## #   7: ModeratelyActiveDistance, 8: LightActiveDistance,
## #   9: SedentaryActiveDistance
```

Identify all the columsn in the daily_activity data.

```
colnames(daily_activity)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

Take a look at the sleep_day data.

```
head(sleep_day)
```

```
## # A tibble: 6 x 5
##           Id SleepDay              TotalSleepRecords TotalMinutesAsleep TotalT~1
##        <dbl> <chr>                             <dbl>              <dbl>    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327      346
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384      407
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412      442
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340      367
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700      712
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304      320
## # ... with abbreviated variable name 1: TotalTimeInBed
```

Identify all the columsn in the daily_activity data.

```
colnames(sleep_day)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"
```

Understanding some summary statistics

I check the number of unique participants in each dataframe.

```
n_distinct(daily_activity$Id)
```

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

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

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

How many observations are there in each dataframe?

```
nrow(daily_activity)
```

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

```
nrow(sleep_day)
```

```
## [1] 413
```

I compute some quick summary statistics for each dataframe.

For the daily activity dataframe:

```
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    SedentaryMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8
##  Median : 7406   Median : 5.245   Median :1057.5
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5
##  Max.   :36019   Max.   :28.030   Max.   :1440.0
```

For the sleep dataframe:

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

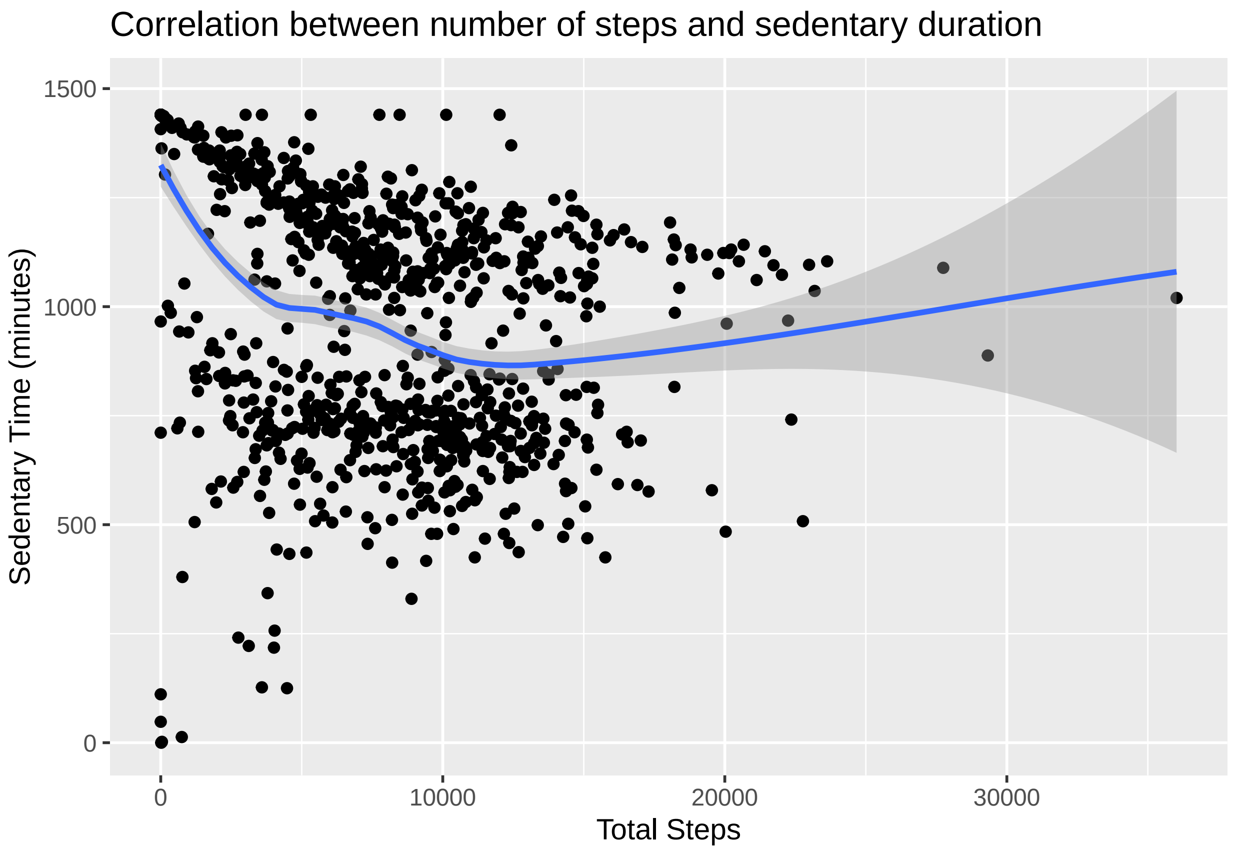
```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

It looks like there might be outliers for the sleep duration and total time in bed. I will explore this issue afterwards.

Plotting a few explorations

What's the relationship between steps taken in a day and sedentary minutes?

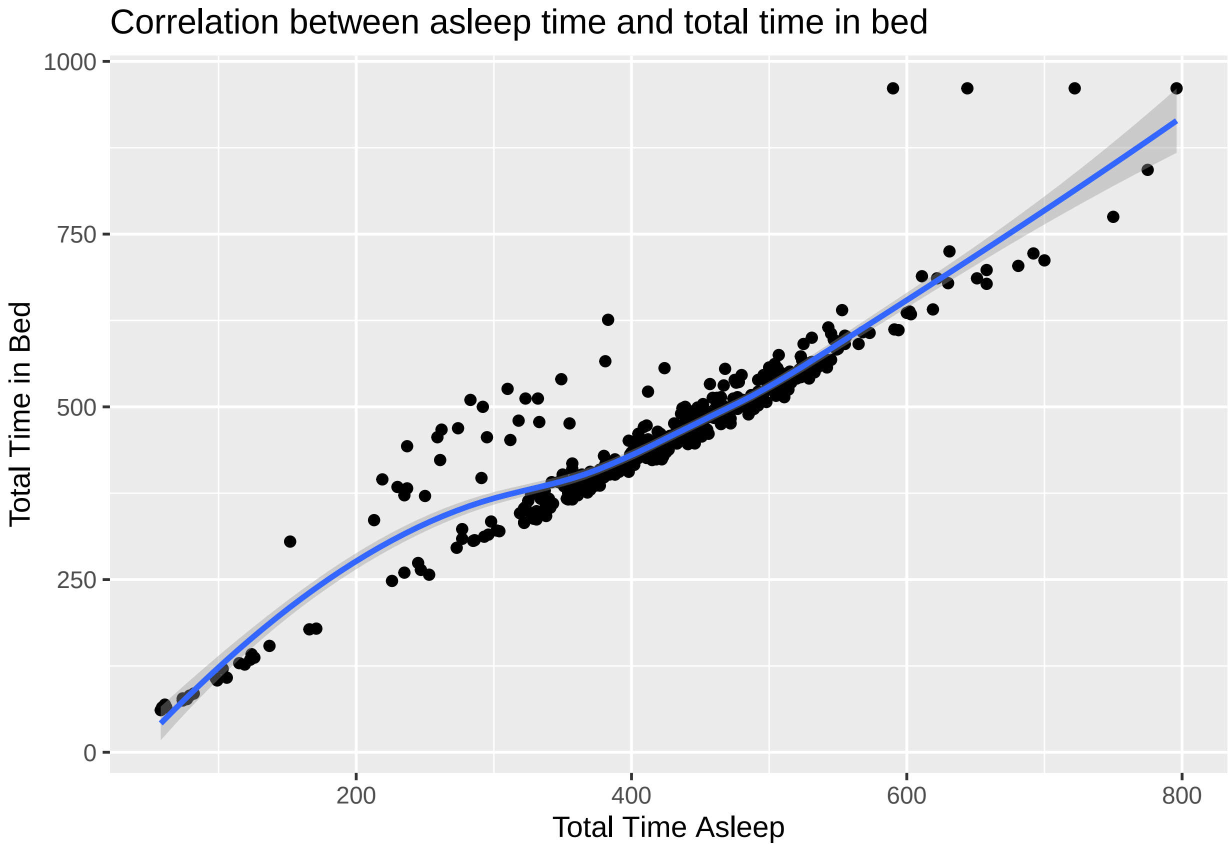
```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Correlation between number of steps and sedentary duration", x = "Total Steps", y = "Se
```



The relationship between steps and sedentary time seems to differ depending on the number of steps. Until about 10000 steps, the more the steps the less the duration of the sedentary time. After the 10000 steps threshold, the duration of the sedentary time tends to increase with increasing the number of steps.

What's the relationship between minutes asleep and time in bed? We might expect it to be almost completely linear - are there any unexpected trends?

```
ggplot(data=sleep_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Correlation between asleep time and total time in bed", y = "Total Time in Bed", x = "T
```



As expected, the relationship between sleep duration and total time in bed is almost perfectly linear.

Merging these two datasets together

```
combined_data <- merge(sleep_day, daily_activity, by="Id")
```

Take a look at how many participants are in this data set (after inner join).

```
n_distinct(combined_data$Id)
```

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

Cleaning

Let's start doing some cleaning. In particular, I change the format of date from string to proper dates.

```
#separate date from time
combined_data <- combined_data %>% separate(SleepDay, into = c('Sleep_day','Sleep_time'), sep = ' ', ex

#remove useless column
combined_data$Sleep_time <- NULL
```

```
#convert dates from strings to proper dates
output_1 <- combined_data %>% select(Sleep_day) %>% mutate(Sleep_date = mdy(Sleep_day))
output_2 <- combined_data %>% select(ActivityDate) %>% mutate(Activity_date = mdy(ActivityDate))

#add columns with proper dates
combined_data$Sleep_date <- output_1$Sleep_date
combined_data$Activity_date <- output_2$Activity_date

#remove old dolumns
combined_data$Sleep_day <- NULL
combined_data$ActivityDate <- NULL
rm(output_1,output_2)

#relocate columns
combined_data <- combined_data %>% relocate(Sleep_date, .after = Id)
combined_data <- combined_data %>% relocate(Activity_date, .before = TotalSteps)
```

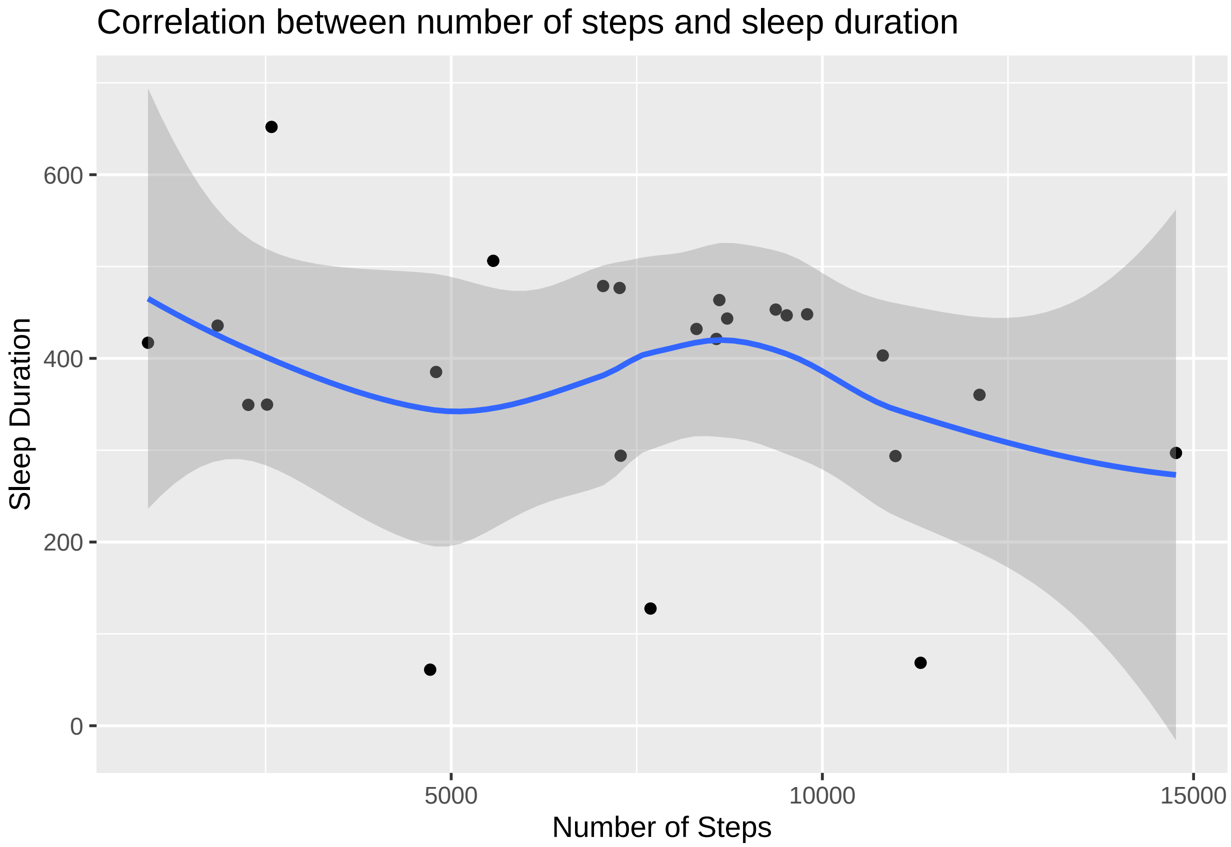
Other trends

Now I can explore some different relationships between activity and sleep as well. For example, we can investigate whether participants who sleep more also take more steps or fewer steps per day. Is there a relationship at all? To answer to these question, I first compute the average sleep duration and number of steps per day by subject.

```
#calculate average sleep by subject
sleep_summary <- combined_data %>% group_by(Id) %>% summarise(avg_sleep_duration = mean(TotalMinutesAsl

#calculate average number of steps by subject
steps_summary <- combined_data %>% group_by(Id) %>% summarise(avg_steps = mean(TotalSteps))

#and I merge for displaying purposes
corr_sleep_steps <- merge(sleep_summary, steps_summary, by='Id')
```

Good. Now, I'm ready to plot the correlation between number of steps and sleep duration.

```
ggplot(data = corr_sleep_steps, mapping = aes(x=avg_steps, y=avg_sleep_duration)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Correlation between number of steps and sleep duration", x = "Number of Steps", y = "Sl
```



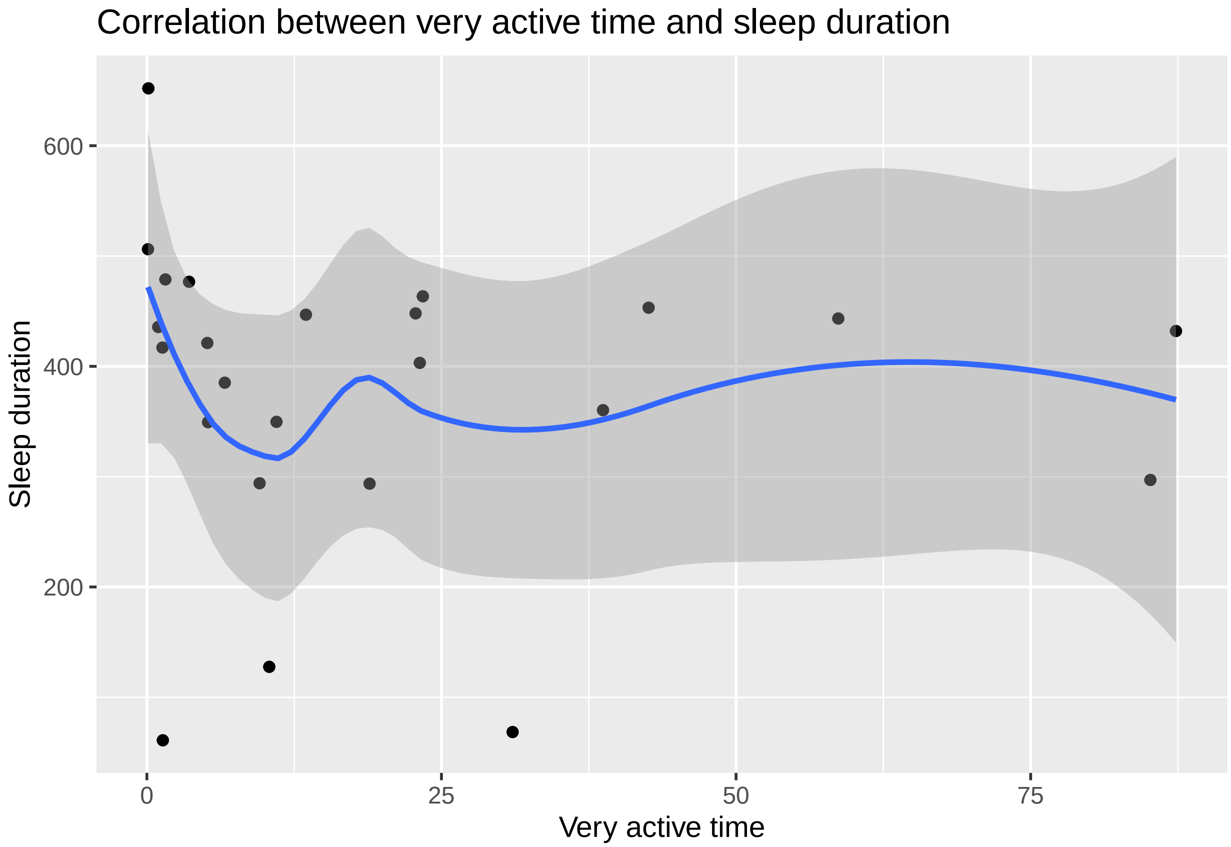
There is no clear correlation between number of steps per day and sleep duration.

Now, let's see whether there is a correlation between sleep and amount of time a subject is very active.

```
#calculate duration of strong activity by subject
very_active_summary <- combined_data %>% group_by(Id) %>% summarise(avg_very_active_time = mean(VeryActi

#and I merge with other data
summary_data <- merge(corr_sleep_steps,very_active_summary,by='Id')
```

Now, I can plot relation between sleep time and amount of very active minutes.

```
ggplot(data = summary_data, mapping = aes(x=avg_very_active_time,y=avg_sleep_duration)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Correlation between very active time and sleep duration",
       x = "Very active time", y = "Sleep duration")
```



Also in this case there is no clear correlation between time spent doing vigorous physical activity and sleep duration.

Analyses

Let's start investigating the research goals described at the beginning of this document. First, I will address the first aim, that is, dividing the dataset in sedentary and very active subjects. To do so, I will work with a new csv file.

```
hourly_activity <- read_csv(here("data", "hourlyIntensities_merged.csv"))
```

Let's see how it is structured.

```
head(hourly_activity)
```

```
## # A tibble: 6 x 4
##           Id ActivityHour          TotalIntensity AverageIntensity
##        <dbl> <chr>                          <dbl>            <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM             20            0.333
## 2 1503960366 4/12/2016 1:00:00 AM               8            0.133
## 3 1503960366 4/12/2016 2:00:00 AM               7            0.117
## 4 1503960366 4/12/2016 3:00:00 AM               0            0
## 5 1503960366 4/12/2016 4:00:00 AM               0            0
## 6 1503960366 4/12/2016 5:00:00 AM               0            0
```

Then, I clean it.

```
hourly_activity <- hourly_activity %>% separate(ActivityHour, into = c('Activity_date','Activity_time')

new_times <- hour(parse_time(gsub("\\.", "", hourly_activity$Activity_time), "%I:%M:%S %p"))

hourly_activity$Activity_time <- new_times
```

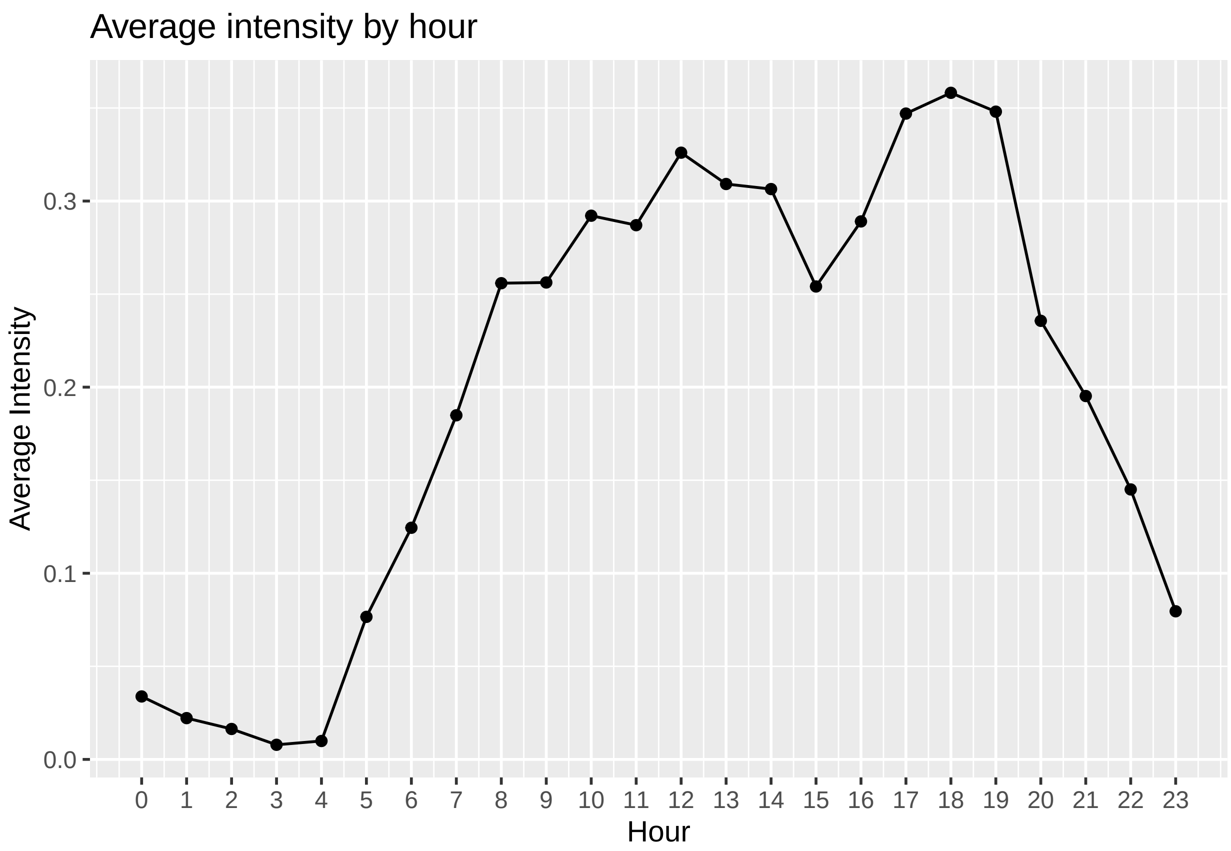
I visualize the mean intesnity by hour. Before that, I compute the required summary stats

```
by_hour <- hourly_activity %>%
  group_by(Id,Activity_time) %>%
  summarise(avg_intensity = mean(AverageIntensity))

head(by_hour)
```

```
## # A tibble: 6 x 3
## # Groups:   Id [1]
##           Id Activity_time avg_intensity
##        <dbl>         <int>         <dbl>
## 1 1503960366             0       0.123
## 2 1503960366             1       0.0583
## 3 1503960366             2       0.0294
## 4 1503960366             3       0.0156
## 5 1503960366             4       0.00556
## 6 1503960366             5       0.00278
```

Now, I plot it.

```
by_hour %>%
  group_by(Activity_time) %>%
  summarise(avg_intensity = mean(avg_intensity)) %>%
  ggplot(mapping=aes(x=Activity_time,y=avg_intensity))+
  geom_line()+
  geom_point()+
  scale_x_continuous(breaks=seq(0,23,1))+
  labs(title='Average intensity by hour', x='Hour', y='Average Intensity')
```



As expected, the average activity is low in late evening and night and it reaches two peaks: one around 12 and one around 18.

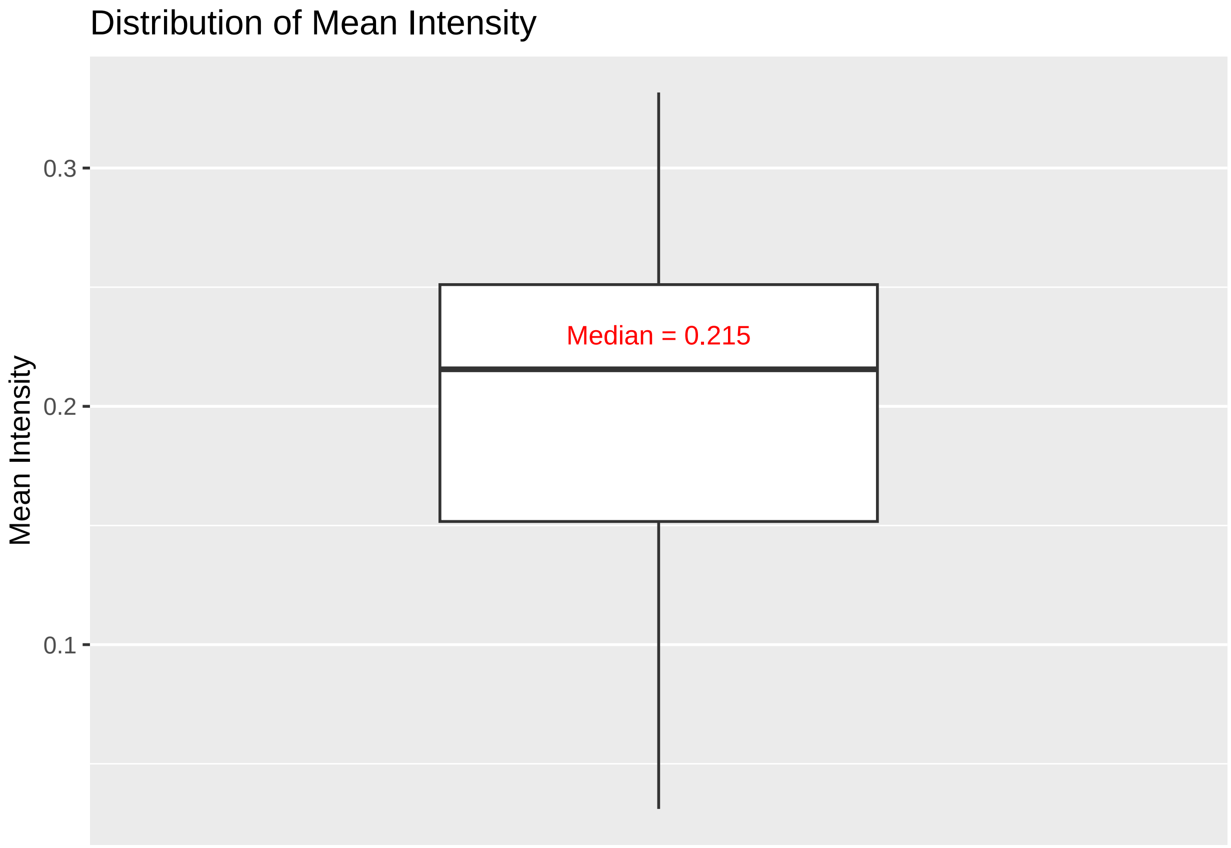
Now, I compute the mean intensity by subject to find active and sedentary persons.

```
by_subj <- by_hour %>%
  group_by(Id) %>%
  summarise(avg_intensity = mean(avg_intensity))

head(by_subj)
```

```
## # A tibble: 6 x 2
##           Id avg_intensity
##        <dbl>         <dbl>
## 1 1503960366        0.270
## 2 1624580081        0.134
## 3 1644430081        0.177
## 4 1844505072        0.0845
## 5 1927972279        0.0311
## 6 2022484408        0.283
```

I visualize the distribution ot mean intensity.

```
ggplot(data=by_subj, mapping=aes(y=avg_intensity))+
  geom_boxplot()+
  #geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 0.5)+
  annotate("text", x=0, y=0.23, label="Median = 0.215", color='red', size=3.5) +
  labs(title='Distribution of Mean Intensity',
       y = 'Mean Intensity',
       x = NULL) +
  scale_x_discrete(breaks=NULL)
```



I also save the median intensity which is necesssary to separate the groups.

```
median_intensity <- median(by_subj$avg_intensity)
```

Now, I can define the two groups.

```
by_subj <- by_subj %>% mutate(group = ifelse(avg_intensity < median_intensity, 'sedentary', 'active'))
by_subj <- by_subj %>% relocate(group, .after = Id)

head(by_subj)
```

```
## # A tibble: 6 x 3
##           Id group     avg_intensity
##        <dbl> <chr>             <dbl>
## 1 1503960366 active           0.270
## 2 1624580081 sedentary        0.134
## 3 1644430081 sedentary        0.177
## 4 1844505072 sedentary        0.0845
## 5 1927972279 sedentary        0.0311
## 6 2022484408 active           0.283
```

To proceed with the analyses I need to extract the active and sedentary subjects Ids.

```
active_Id <- by_subj %>% select(Id,group) %>% filter(group=='active')
active_Id$group = NULL #do not need the group
active_Id <- as.list(active_Id)

sedentary_Id <- by_subj %>% select(Id,group) %>% filter(group=='sedentary')
sedentary_Id$group = NULL
sedentary_Id <- as.list(sedentary_Id)
```

I need to add the column "group" to the dataframe of daily activities

```
daily_activity <- daily_activity %>% mutate(group = ifelse(Id %in% sedentary_Id$Id, 'sedentary', 'activ
daily_activity <- daily_activity %>% relocate(group, .after = Id)

head(daily_activity)
```

```
## # A tibble: 6 x 16
##         Id group Activ~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8
##      <dbl> <chr> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1   1.50e9 acti~ 4/12/2~   13162    8.5     8.5        0    1.88   0.550    6.06
## 2   1.50e9 acti~ 4/13/2~   10735    6.97    6.97       0    1.57   0.690    4.71
## 3   1.50e9 acti~ 4/14/2~   10460    6.74    6.74       0    2.44   0.400    3.91
## 4   1.50e9 acti~ 4/15/2~    9762    6.28    6.28       0    2.14   1.26     2.83
## 5   1.50e9 acti~ 4/16/2~   12669    8.16    8.16       0    2.71   0.410    5.04
## 6   1.50e9 acti~ 4/17/2~    9705    6.48    6.48       0    3.19   0.780    2.51
## # ... with 6 more variables: SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>, and
## #   abbreviated variable names 1: ActivityDate, 2: TotalSteps,
## #   3: TotalDistance, 4: TrackerDistance, 5: LoggedActivitiesDistance,
## #   6: VeryActiveDistance, 7: ModeratelyActiveDistance, 8: LightActiveDistance
```

I can now extract some summary stats. I start with the averages

```
activity_summary_avg <- daily_activity %>%
  select(group,TotalSteps,VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes,SedentaryMinutes,C
  group_by(group) %>%
  summarise(avg_VeryActiveMinutes = mean(VeryActiveMinutes),
            avg_FairlyActiveMinutes = mean(FairlyActiveMinutes),
            avg_LightlyActiveMinutes = mean(LightlyActiveMinutes),
            avg_SedentaryMinutes = mean(SedentaryMinutes))
```

I also need the standard errors of those averages

```
activity_summary_se <- daily_activity %>%
  select(group,TotalSteps,VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes,SedentaryMinutes,C
  group_by(group) %>%
  summarise(se_VeryActiveMinutes = std.error(VeryActiveMinutes),
            se_FairlyActiveMinutes = std.error(FairlyActiveMinutes),
            se_LightlyActiveMinutes = std.error(LightlyActiveMinutes),
            se_SedentaryMinutes = std.error(SedentaryMinutes))
```

Before plotting, I need to convert data from wide to long format

```
#averages first
activity_summary_avg_long <- activity_summary_avg %>% pivot_longer(cols = starts_with('avg_'), names_to

#then, se
activity_summary_se_long <- activity_summary_se %>% pivot_longer(cols = starts_with('se_'), names_to =

head(activity_summary_avg_long)
```

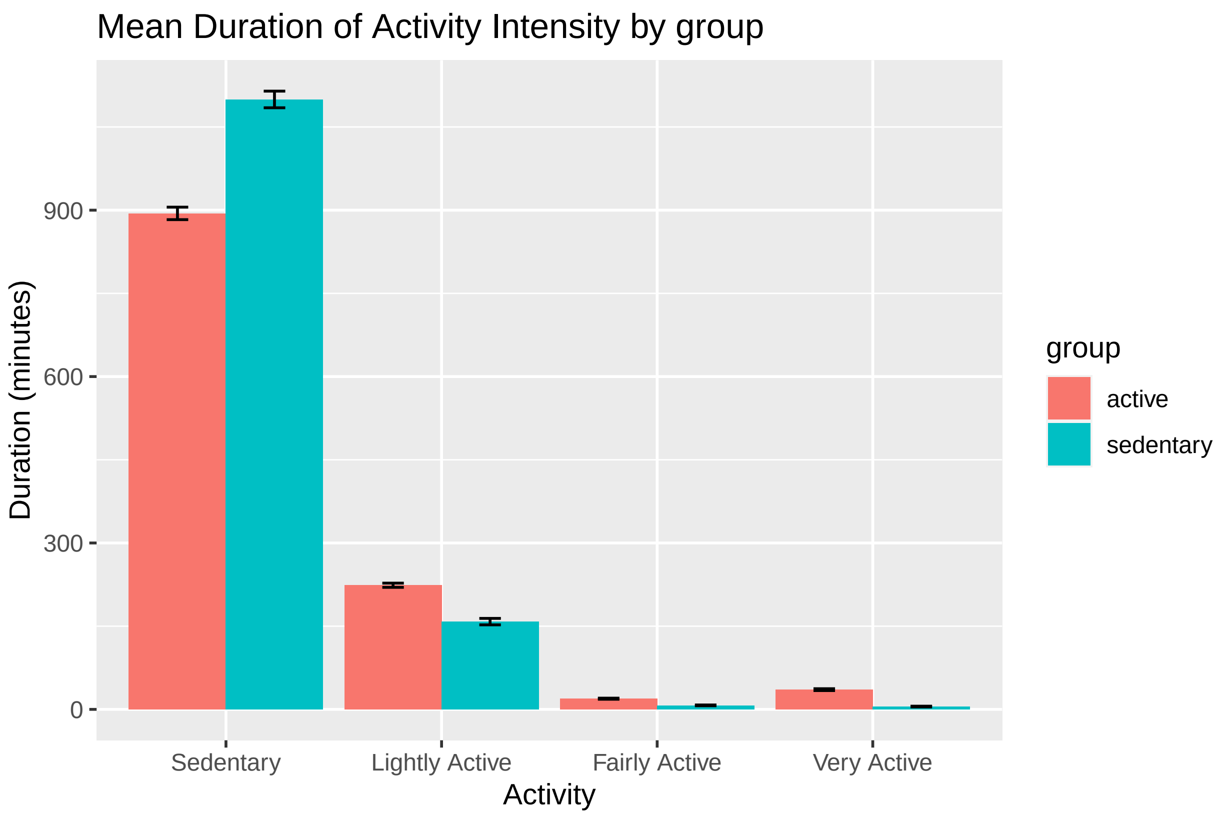
```
## # A tibble: 6 x 3
##   group     variable                  value
##   <chr>     <chr>                     <dbl>
## 1 active    avg_VeryActiveMinutes     35.6
## 2 active    avg_FairlyActiveMinutes   19.2
## 3 active    avg_LightlyActiveMinutes 224.
## 4 active    avg_SedentaryMinutes     894.
## 5 sedentary avg_VeryActiveMinutes      5.08
## 6 sedentary avg_FairlyActiveMinutes    7.23
```

I do some extra cleaning.

```
#rename column value
activity_summary_se_long <- rename(activity_summary_se_long, se = value)

#add standard error to the df with the averages
activity_summary_avg_long$se = activity_summary_se_long$se

#I factorize to specify the order of conditions to show in the bar plot
activity_summary_avg_long$variable <- factor(activity_summary_avg_long$variable,levels = c("avg_Sedenta

#rename df
activity_summary_long = activity_summary_avg_long

#finally, I delete redundant info
rm(activity_summary_avg_long,activity_summary_se_long,activity_summary_avg,activity_summary_se)
```

Now, I can plot

```
ggplot(data = activity_summary_long, mapping = aes(variable, value, fill=group))+
  geom_bar(position=position_dodge(), stat="identity")+
  geom_errorbar(aes(ymin=value-se, ymax=value+se),
              width=.2,                    # Width of the error bars
              position=position_dodge(.9))+
  labs(title = 'Mean Duration of Activity Intensity by group',
      y = 'Duration (minutes)',
      x = 'Activity',
      caption="SEM are indicated")+
  scale_x_discrete(labels=c("avg_SedentaryMinutes"="Sedentary",
                            "avg_LightlyActiveMinutes"="Lightly Active",
                            "avg_FairlyActiveMinutes"= "Fairly Active",
                            "avg_VeryActiveMinutes"="Very Active"))+
  theme(plot.caption.position = "plot",
        plot.caption = element_text(hjust = 0))
```



SEM are indicated

For completeness, I show also the raw data in table format.

```
head(activity_summary_long, n=dim(activity_summary_long))
```

```
## # A tibble: 8 x 4
##   group     variable                   value     se
##   <chr>     <fct>                      <dbl>  <dbl>
## 1 active    avg_VeryActiveMinutes      35.6   1.69
## 2 active    avg_FairlyActiveMinutes    19.2   0.971
## 3 active    avg_LightlyActiveMinutes  224.    3.81
## 4 active    avg_SedentaryMinutes      894.   11.3
## 5 sedentary avg_VeryActiveMinutes       5.08  0.676
## 6 sedentary avg_FairlyActiveMinutes     7.23  0.747
## 7 sedentary avg_LightlyActiveMinutes  158.    5.81
## 8 sedentary avg_SedentaryMinutes     1100.   15.0
```

It is clear that the two groups differ in terms of pattern of activity. The active group spend less time in sedentary state and spend more time in all the other active states than the sedentary group.

Now, I investigate how the two groups differ in number of daily steps

```
steps_summary <- daily_activity %>%
  select(Id,group,TotalSteps) %>%
  group_by(Id,group) %>%
  summarise(avg_steps = mean(TotalSteps))
```

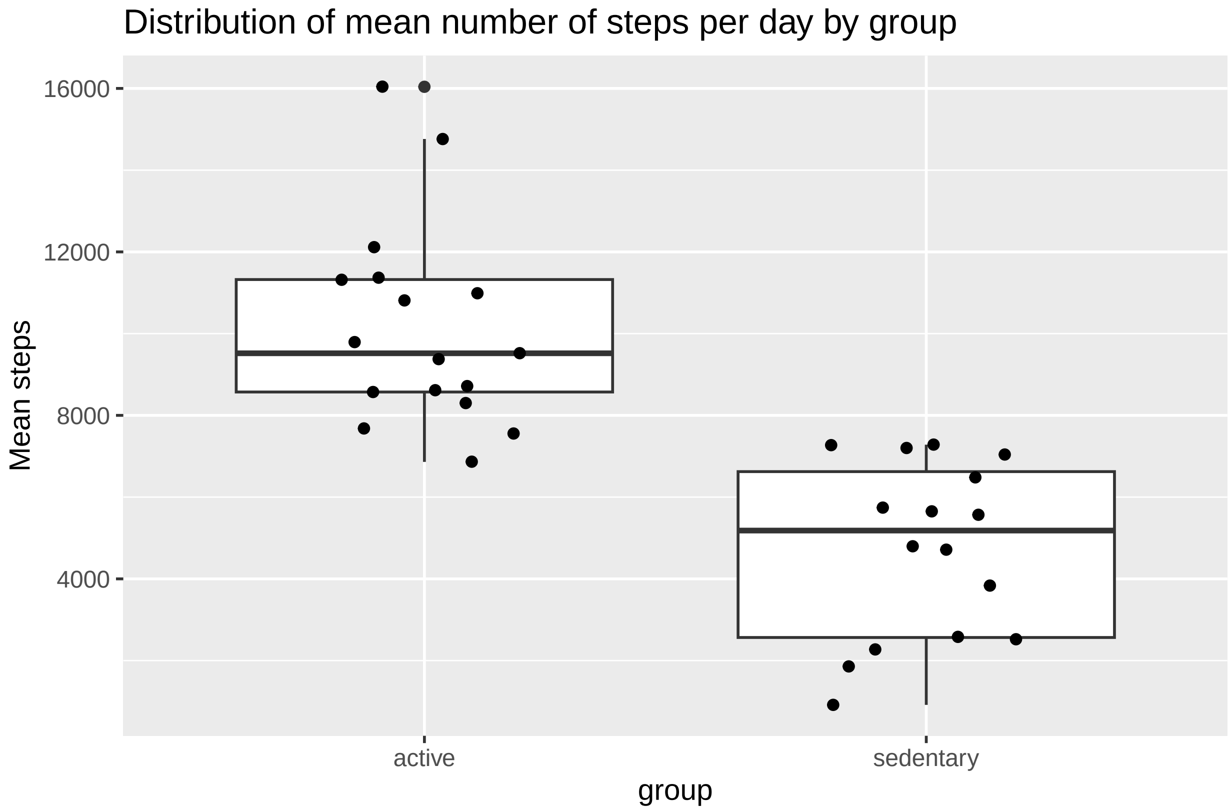
```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

```
head(steps_summary)
```

```
## # A tibble: 6 x 3
## # Groups:   Id [6]
##           Id group     avg_steps
##        <dbl> <chr>         <dbl>
## 1 1503960366 active       12117.
## 2 1624580081 sedentary     5744.
## 3 1644430081 sedentary     7283.
## 4 1844505072 sedentary     2580.
## 5 1927972279 sedentary      916.
## 6 2022484408 active       11371.
```

and I visualize it

```
ggplot(data=steps_summary, aes(x=group,y=avg_steps))+
  geom_boxplot()+
  geom_jitter(width = 0.2)+
  labs(title='Distribution of mean number of steps per day by group',
       y='Mean steps', caption = "Dots indicate single datapoints")+
  theme(plot.caption.position = "plot",
        plot.caption = element_text(hjust = 0))
```



Dots indicate single datapoints

Same for number of calories

```
calories_summary <- daily_activity %>%
  select(Id,group,Calories) %>%
  group_by(Id,group) %>%
  summarise(avg_calories = mean(Calories))
```

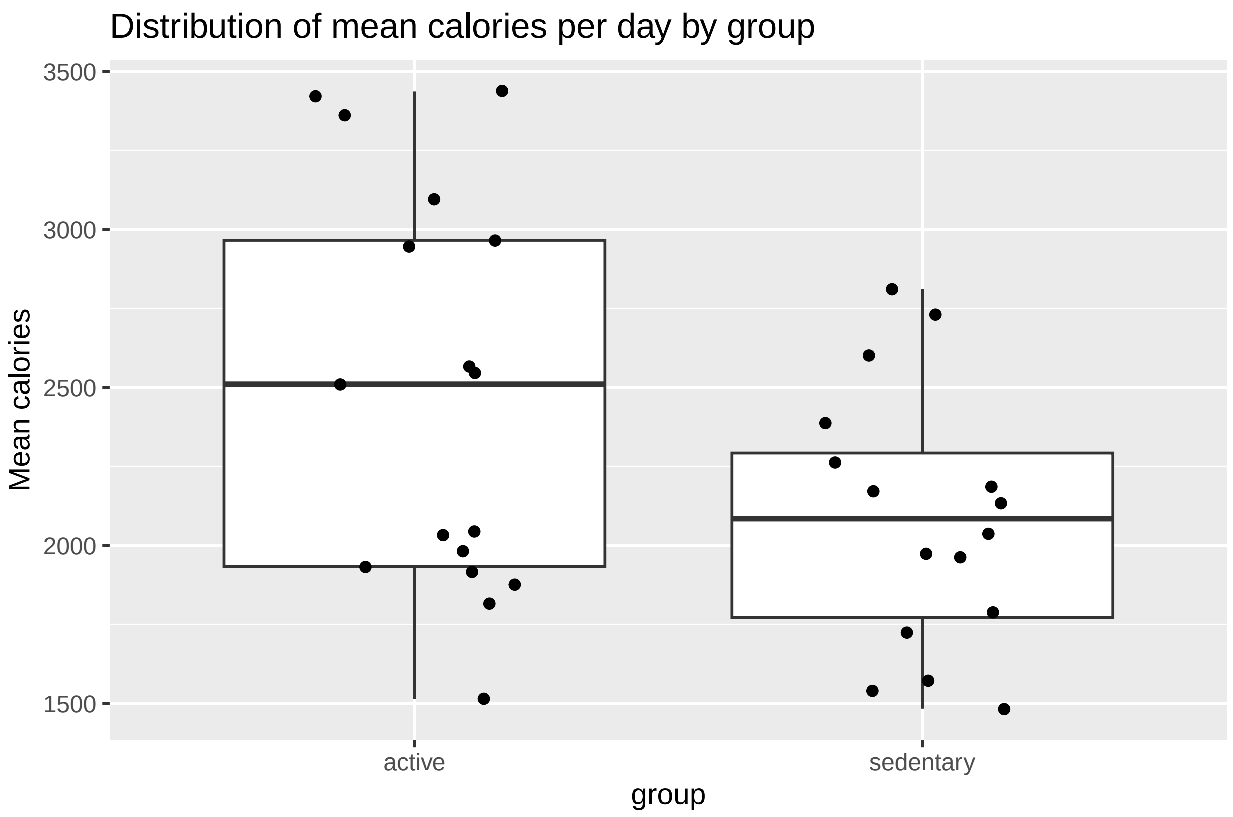
```
## ‘summarise()‘ has grouped output by ’Id’. You can override using the ‘.groups‘
## argument.
```

```
head(calories_summary)
```

```
## # A tibble: 6 x 3
## # Groups:   Id [6]
##           Id group     avg_calories
##        <dbl> <chr>            <dbl>
## 1 1503960366 active           1816.
## 2 1624580081 sedentary        1483.
## 3 1644430081 sedentary        2811.
## 4 1844505072 sedentary        1573.
## 5 1927972279 sedentary        2173.
## 6 2022484408 active           2510.
```

I visualize it

```
ggplot(data=calories_summary, aes(x=group,y=avg_calories))+
  geom_boxplot()+
  geom_jitter(width = 0.2)+
  labs(title='Distribution of mean calories per day by group',
       y='Mean calories', caption = "Dots indicate single datapoints")+
  theme(plot.caption.position = "plot",
        plot.caption = element_text(hjust = 0))
```



Dots indicate single datapoints

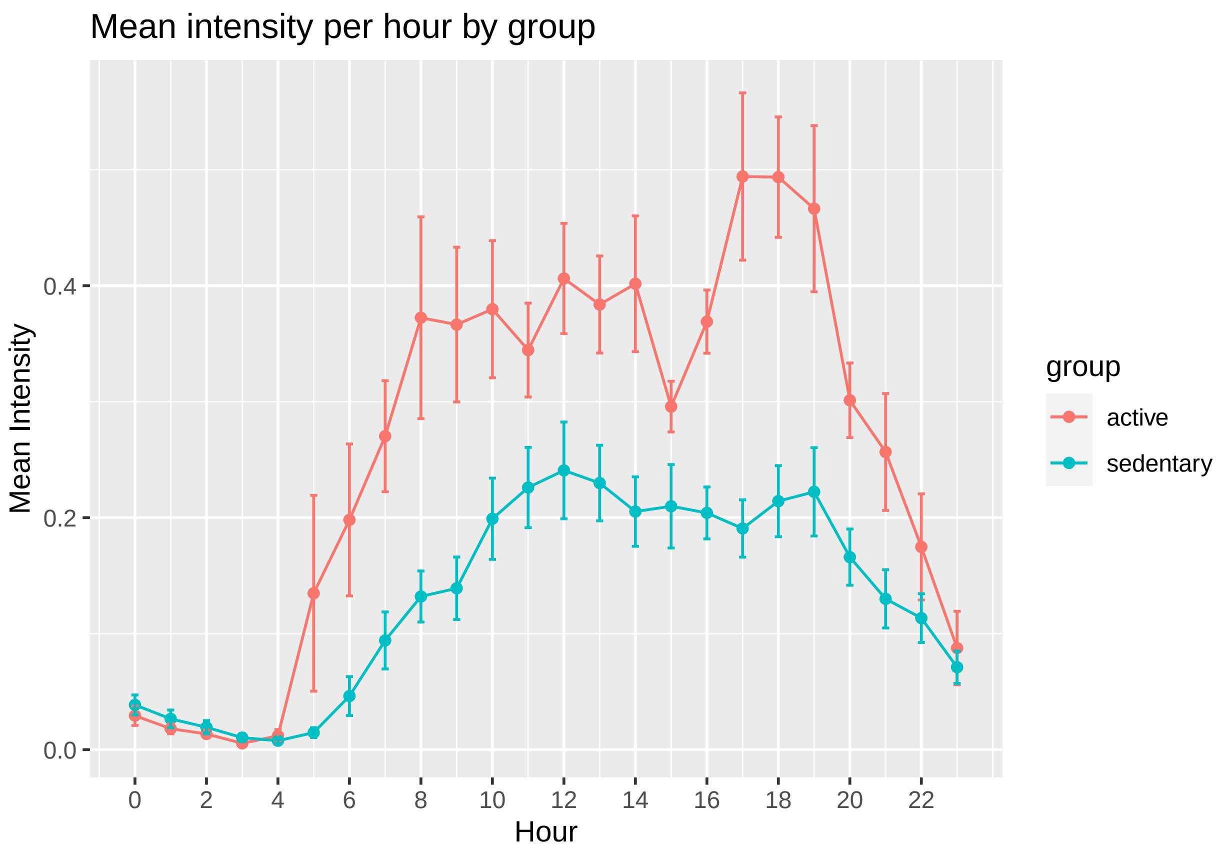
Now, I want to plot the mean intensity by hour by group. To do so, I first add the 'group' column to the df with hour data.

```
by_hour <- by_hour %>% mutate(group = ifelse(Id %in% sedentary_Id$Id,"sedentary","active"))
by_hour <- relocate(by_hour, group, .after=Id)
```

Now, I can plot it

```
by_hour %>%
  group_by(group,Activity_time) %>%
  summarise(intensity = mean(avg_intensity), se = std.error(avg_intensity)) %>%
  ggplot(mapping=aes(x=Activity_time, y=intensity, group=group, color=group))+
  geom_line()+
  geom_point()+
  geom_errorbar(aes(ymin=intensity-se, ymax=intensity+se),
              width=.2)+
  labs(title = 'Mean intensity per hour by group',
       y='Mean Intensity',
       x='Hour')+
  scale_x_continuous(breaks=seq(0,23,2))
```

```
## ‘summarise()‘ has grouped output by ’group’. You can override using the
## ‘.groups‘ argument.
```



We can clearly see that the active group is more active in daytime and also in the evening. It is not only due to higher, isolated, peaks of activity but generally, they have a higher baseline of activity.

Let's now see what happens with heart rate data.

First, I load the right csv file

```
heart_rate <- read_csv(here("data","heartrate_seconds_merged.csv"))
```

```
## Rows: 2483658 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(heart_rate)
```

```
## # 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
```

I do some cleaning. Date formatting first.

```
heart_rate <- heart_rate %>% separate(Time, into = c("Date", "Time"), sep = " ", extra = "merge")
new_times <- hour(parse_time(gsub("\\.", "", heart_rate$Time), "%I:%M:%S %p"))

heart_rate <- heart_rate %>% mutate(Time = new_times)
```

Let'see if it is correct

```
head(heart_rate)
```

```
## # A tibble: 6 x 4
##           Id Date       Time Value
##        <dbl> <chr>     <int> <dbl>
## 1 2022484408 4/12/2016     7    97
## 2 2022484408 4/12/2016     7   102
## 3 2022484408 4/12/2016     7   105
## 4 2022484408 4/12/2016     7   103
## 5 2022484408 4/12/2016     7   101
## 6 2022484408 4/12/2016     7    95
```

Let's add the 'group' column

```
heart_rate <- heart_rate %>% mutate(group = ifelse(Id %in% sedentary_Id$Id, "sedentary", "active"))
heart_rate <- relocate(heart_rate, group, .after=Id)
head(heart_rate)
```

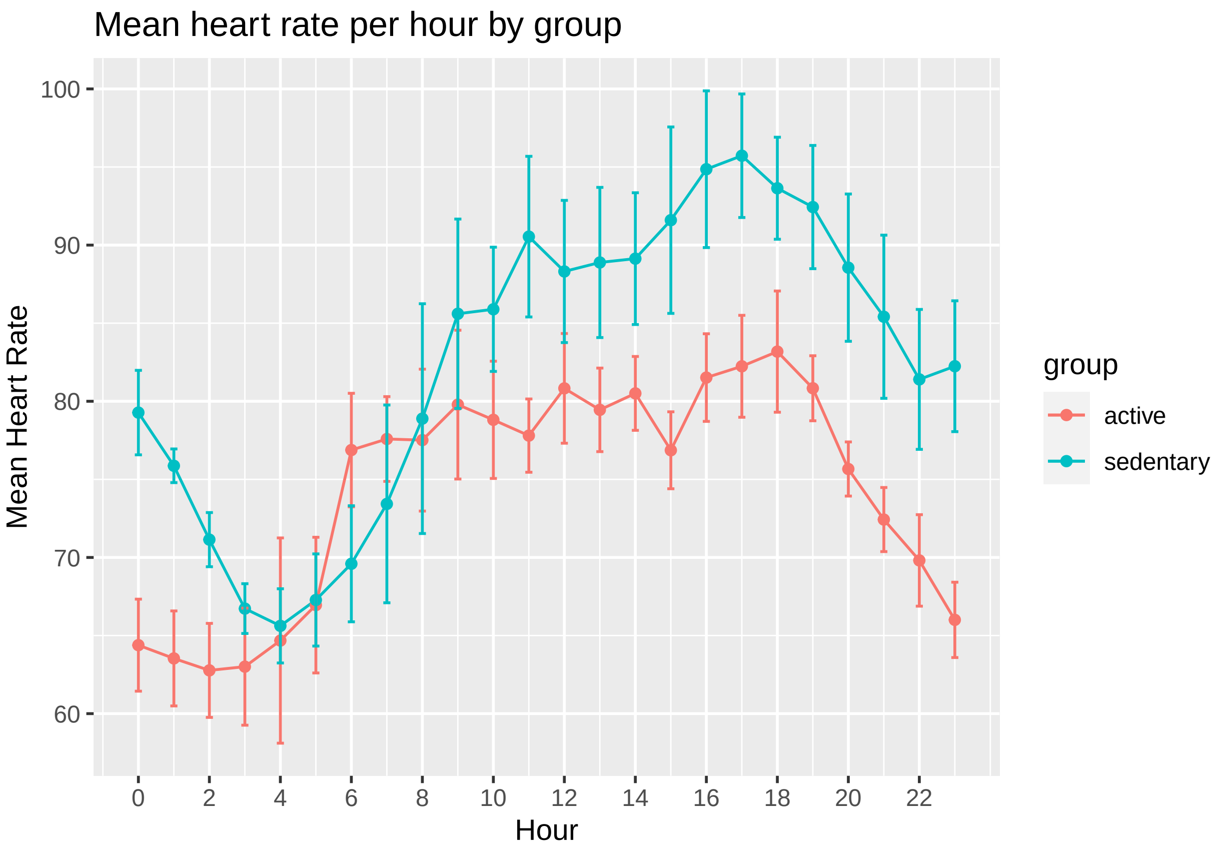
```
## # A tibble: 6 x 5
##           Id group  Date       Time Value
##        <dbl> <chr>  <chr>     <int> <dbl>
## 1 2022484408 active 4/12/2016     7    97
## 2 2022484408 active 4/12/2016     7   102
## 3 2022484408 active 4/12/2016     7   105
## 4 2022484408 active 4/12/2016     7   103
## 5 2022484408 active 4/12/2016     7   101
## 6 2022484408 active 4/12/2016     7    95
```

Let's see how the two groups differ in heart rate by hour. I prepare the data first.

```
heartrate_byhour <- heart_rate %>%
  select(Id,group,Time,Value) %>%
  group_by(Id, group, Time) %>%
  summarise(avg_heartrate = mean(Value))

heartrate_byhour <- heartrate_byhour %>%
  group_by(group,Time) %>%
  summarise(heartrate = mean(avg_heartrate,na.rm = TRUE), se = std.error(avg_heartrate,na.rm = TRUE))
```

I can plot now

```
ggplot(data = heartrate_byhour,mapping= aes(x=Time,y=heartrate, group = group, color = group))+
  geom_line()+
  geom_point()+
  geom_errorbar(aes(ymin=heartrate-se, ymax=heartrate+se),
              width=.2)+
  labs(title = 'Mean heart rate per hour by group',
      y='Mean Heart Rate',
      x='Hour')+
  scale_x_continuous(breaks=seq(0,23,2))
```



This looks very interesting. Heart rate is generally higher in the sedentary group even tough they are less active. This is also true during the night when the persons are sleeping!

Let's do some analyses to find out whether this trend is statistically significant. First, I compute the mean heart rate by group

```
mean_heartrate_active <- mean(heartrate_byhour$heartrate[heartrate_byhour$group=="active"])
mean_heartrate_sedentary <- mean(heartrate_byhour$heartrate[heartrate_byhour$group=="sedentary"])

mean_heartrate_active
```

```
## [1] 74.29233
```

```
mean_heartrate_sedentary
```

```
## [1] 82.58536
```

Compare these means

```
t.test(heartrate_byhour$heartrate[heartrate_byhour$group=="active"], heartrate_byhour$heartrate[heartra
```

```
##
##  Welch Two Sample t-test
##
## data:  heartrate_byhour$heartrate[heartrate_byhour$group == "active"] and heartrate_byhour$heartrate
## t = -3.4171, df = 42.469, p-value = 0.001406
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.189165  -3.396892
## sample estimates:
## mean of x mean of y
##  74.29233  82.58536
```

Yes, the difference is significant!

I am ready to approach the final part of the analyses, that is, the sleep data.

I add the 'group' column to sleep df.

```
sleep_day <- sleep_day %>% mutate(group = ifelse(Id %in% sedentary_Id$Id, "sedentary", "active"))
sleep_day <- relocate(sleep_day, group, .after=Id)
```

I calculate some summary stats

```
sleep_summary <- sleep_day %>% select(Id, group, TotalMinutesAsleep, TotalTimeInBed) %>%
  group_by(Id,group) %>%
  summarise(time_asleep = mean(TotalMinutesAsleep),
            time_inbed = mean(TotalTimeInBed),
            se_asleep = std.error(TotalMinutesAsleep),
            se_inbed = std.error(TotalTimeInBed))
```

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

Visualize it

```
head(sleep_summary)
```

```
## # A tibble: 6 x 6
## # Groups:   Id [6]
##           Id group     time_asleep time_inbed se_asleep se_inbed
##        <dbl> <chr>           <dbl>      <dbl>     <dbl>    <dbl>
## 1 1503960366 active           360.       383.     20.1     19.6
## 2 1644430081 sedentary         294         346     167.     205.
## 3 1844505072 sedentary        652        961      38.3      0
## 4 1927972279 sedentary        417       438.      98.0    100.
## 5 2026352035 sedentary        506.      538.       7.99     8.01
## 6 2320127002 sedentary         61         69      NA       NA
```

A previous visualization identified potential outliers. Let's investigate this issue in a more detailed way. Let's check asleep time first

```
sleep_summary %>%
  group_by(group) %>%
  identify_outliers(time_asleep)
```

```
## # A tibble: 2 x 8
##   group             Id time_asleep time_inbed se_asleep se_inbed is.ou~1 is.ex~2
##   <chr>          <dbl>       <dbl>      <dbl>     <dbl>    <dbl> <lgl>   <lgl>
## 1 active    7007744171        68.5       71.5      10.5     10.5 TRUE    FALSE
## 2 sedentary 2320127002        61         69        NA       NA   TRUE    FALSE
## # ... with abbreviated variable names 1: is.outlier, 2: is.extreme
```

Now, the same for the time in bed

```
sleep_summary %>%
  group_by(group) %>%
  identify_outliers(time_inbed)
```

```
## # A tibble: 4 x 8
##   group             Id time_asleep time_inbed se_asleep se_inbed is.ou~1 is.ex~2
##   <chr>          <dbl>       <dbl>      <dbl>     <dbl>    <dbl> <lgl>   <lgl>
## 1 active    4558609924       128.       140          11.6     10.1 TRUE    FALSE
## 2 active    7007744171        68.5       71.5      10.5     10.5 TRUE    TRUE
## 3 sedentary 1844505072       652        961          38.3      0   TRUE    TRUE
## 4 sedentary 2320127002        61         69        NA       NA   TRUE    FALSE
## # ... with abbreviated variable names 1: is.outlier, 2: is.extreme
```

Let's see if there is something strange with those Ids. Perhaps, too little data?

```
filter(sleep_day,Id==7007744171)
```

```
## # A tibble: 2 x 6
##           Id group  SleepDay              TotalSleepRecords TotalMinut~1 Total~2
##        <dbl> <chr>  <chr>                             <dbl>        <dbl>   <dbl>
## 1 7007744171 active 4/16/2016 12:00:00 AM                 1           79      82
## 2 7007744171 active 5/1/2016 12:00:00 AM                  1           58      61
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed
```

```
filter(sleep_day,Id==2320127002)
```

```
## # A tibble: 1 x 6
##           Id group     SleepDay              TotalSleepRecords TotalMi~1 Total~2
##        <dbl> <chr>     <chr>                             <dbl>     <dbl>   <dbl>
## 1 2320127002 sedentary 4/23/2016 12:00:00 AM                 1        61      69
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed
```

```r
filter(sleep_day,Id==4558609924)
```

```
## # A tibble: 5 x 6
##           Id group  SleepDay              TotalSleepRecords TotalMinut~1 Total~2
##        <dbl> <chr>  <chr>                             <dbl>        <dbl>   <dbl>
## 1 4558609924 active 4/21/2016 12:00:00 AM                 1          126     137
## 2 4558609924 active 4/26/2016 12:00:00 AM                 1          103     121
## 3 4558609924 active 4/29/2016 12:00:00 AM                 1          171     179
## 4 4558609924 active 5/1/2016 12:00:00 AM                  1          115     129
## 5 4558609924 active 5/8/2016 12:00:00 AM                  1          123     134
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed
```

```r
filter(sleep_day,Id==1844505072)
```

```
## # A tibble: 3 x 6
##           Id group     SleepDay              TotalSleepRecords TotalMi~1 Total~2
##        <dbl> <chr>     <chr>                             <dbl>     <dbl>   <dbl>
## 1 1844505072 sedentary 4/15/2016 12:00:00 AM                 1       644     961
## 2 1844505072 sedentary 4/30/2016 12:00:00 AM                 1       722     961
## 3 1844505072 sedentary 5/1/2016 12:00:00 AM                  1       590     961
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed
```

There are in particular two Ids with 2 or less night data. I remove these two Ids.

```r
sleep_cleaned <- sleep_day %>%
  filter(!Id %in% c(7007744171,2320127002))
```

and I redo the stats with the cleaned df

```r
sleep_summary_clean <- sleep_cleaned %>% select(Id, group, TotalMinutesAsleep, TotalTimeInBed) %>%
  group_by(Id,group) %>%
  summarise(time_asleep = mean(TotalMinutesAsleep),
            time_inbed = mean(TotalTimeInBed),
            se_asleep = std.error(TotalMinutesAsleep),
            se_inbed = std.error(TotalTimeInBed))
```

```
## ‘summarise()‘ has grouped output by ’Id’. You can override using the ‘.groups‘
## argument.
```

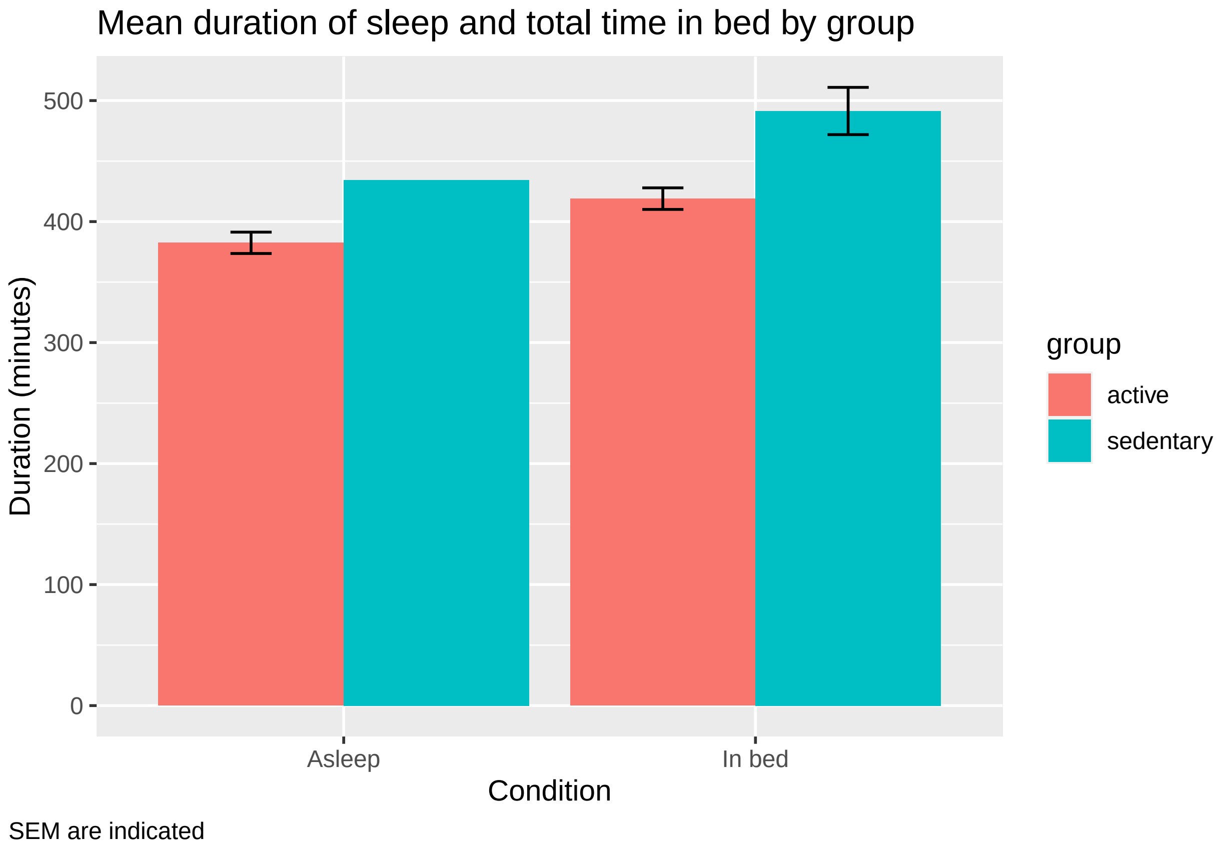
```r
sleep_summary_clean <- sleep_summary_clean %>%
  group_by(group) %>%
  summarise(time_asleep = mean(time_asleep),
            time_inbed = mean(time_inbed),
            se_asleep = std.error(se_asleep),
            se_inbed = std.error(se_inbed))
```

I change the format to long for visualization purposes

```
sleep_summary_clean_long <- pivot_longer(sleep_summary_clean, cols=starts_with("time_"), names_to="condi
#adding standard errors
sleep_summary_clean_long$se <- c(sleep_summary_clean$se_asleep[1], sleep_summary_clean_long$se_inbed[1]
#removing redundant cols
sleep_summary_clean_long$se_asleep = NULL
sleep_summary_clean_long$se_inbed = NULL
```

Ready to plot

```
ggplot(data = sleep_summary_clean_long, mapping=aes(x=condition,y=value, fill = group))+
  geom_bar(position=position_dodge(), stat="identity")+
  geom_errorbar(aes(ymin=value-se, ymax=value+se),
                width=.2,                    # Width of the error bars
                position=position_dodge(.9))+
  labs(title = 'Mean duration of sleep and total time in bed by group',
       y = 'Duration (minutes)',
       x = 'Condition',
       caption="SEM are indicated")+
  scale_x_discrete(labels=c("time_asleep"="Asleep",
                            "time_inbed"="In bed"))+
  theme(plot.caption.position = "plot",
        plot.caption = element_text(hjust = 0))
```



It is evident how the sedentary group tends to sleep more and to stay in bed for longer than the active group. Let's examine whether this trend is statistically significant or not.

To chose the right statistical tool, I need to verify whether data are normally distributed or not.

I start with the sleep data of the active group

```
active_asleep <- sleep_cleaned %>%
  select(Id,group,TotalMinutesAsleep) %>%
  filter(group=="active") %>%
  group_by(Id) %>%
  summarise(time_asleep = mean(TotalMinutesAsleep)) %>%
  select(time_asleep)

shapiro.test(active_asleep$time_asleep)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  active_asleep$time_asleep
## W = 0.77423, p-value = 0.004842
```

Not normal.

Sleep data of sedentary group

```
sedentary_asleep <- sleep_cleaned %>%
  select(Id,group,TotalMinutesAsleep) %>%
  filter(group=="sedentary") %>%
  group_by(Id) %>%
  summarise(time_asleep = mean(TotalMinutesAsleep)) %>%
  select(time_asleep)

shapiro.test(sedentary_asleep$time_asleep)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sedentary_asleep$time_asleep
## W = 0.94505, p-value = 0.6104
```

Normal

Time in bed data of the active group

```
active_inbed <- sleep_cleaned %>%
  select(Id,group,TotalTimeInBed) %>%
  filter(group=="active") %>%
  group_by(Id) %>%
  summarise(time_inbed = mean(TotalTimeInBed)) %>%
  select(time_inbed)

shapiro.test(active_inbed$time_inbed)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  active_inbed$time_inbed
## W = 0.73768, p-value = 0.001993
```

Not normal.

Time in bed data of the sedentary group

```
sedentary_inbed <- sleep_cleaned %>%
  select(Id,group,TotalTimeInBed) %>%
  filter(group=="sedentary") %>%
  group_by(Id) %>%
  summarise(time_inbed = mean(TotalTimeInBed)) %>%
  select(time_inbed)

shapiro.test(sedentary_inbed$time_inbed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sedentary_inbed$time_inbed
## W = 0.7108, p-value = 0.001191
```

I proceed with non-parametric tests. First, I compare the median asleep time of the two groups

```
wilcox.test(active_asleep$time_asleep, sedentary_asleep$time_asleep)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  active_asleep$time_asleep and sedentary_asleep$time_asleep
## W = 48, p-value = 0.4562
## alternative hypothesis: true location shift is not equal to 0
```

Then, I compare the median total time in bed of the two groups

```
wilcox.test(active_inbed$time_inbed, sedentary_inbed$time_inbed)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  active_inbed$time_inbed and sedentary_inbed$time_inbed
## W = 55, p-value = 0.7713
## alternative hypothesis: true location shift is not equal to 0
```

Both comparison are not significant maybe because of the small sample sizes.

Summary

This project was aimed at analyzing Fitbit data to understand how very active users differ compared to more sedentary users. Importantly, these two groups of users were defined based on Fitbit data rather than their self-evaluation.

We have discriminated active from sedentary persons based on the mean intensity per day. The following analyses demonstrated that active people spent less time in sedentary state and more time doing from light to strong physical activity than the sedentary group. Importantly, with the exception of nigh time, they were always more active than sedentary people. This indicates that the higher mean intensity in the active group was not due to isolated peaks of activity (e.g., going to the gym at 18 pm) but rather to a constantly higher level of activity across all day. The active persons showed also a higher number of steps, higher consumption of calories and lower heart rate. Interestingly, the lower hear rate was also present during sleep.

As for the relationship between physical activity and sleep, the analyses showed little evidence of an effect of such activity on sleep. Although, there is a visual trend towards lower duration of the total time in bed as well as the asleep time compared to the sedentary group, this trend was not significant. It is possible this was due to the small sample size of Fitbit users we could evaluate.

Since, because of privacy concerns, sensitive data such as age, gender and health conditions were not available, it is possible that such variables might have contributed to the differences between groups.

Conclusion

This project has showed that:

- an active lifestyle, as defined by higher number of steps, number of calories and mean intensity is associated with lower heart rate. A lower heart rate is tipically a sign the heart is working well since it pumps more blood with each contraction;
- there is a trend towards better sleep efficiency in active users compared to more sedentary users which need to be confirmed by future investigations with larger sample sizes.

This may be helpful as a guidance for person aiming at developing a more active lifestyle.