R Notebook

Loading and installing packages for working

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.0
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflicts to becom
e errors
```

```
library(lubridate)
library(ggplot2)
```

#load the dataset, after analyzing the data available, most of data was irrelevant or incomplete, dailityactivity_merged, had most of the useful data already in a single CSV, the same happens with sleepday. I could not give any use to heartrate, and also the data is incomplete and does not represent all population.

#Also renamed DailyActivity to work easily.

```
daily_activity <- read.csv("dailyActivity.csv")
```

```
sleep_day <- read.csv("sleepDay.csv")
```

Exploring a little of the data table in R, specially to see the datatypes, also it looks like the "ActivityDate" is not a date type, also I need to check why ID is a float, it might be an Integer.

```
head(daily_activity)
```

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistanc
	<dbl>	<chr>	<int>	<dbl>	<dbl>	<db
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	

6 rows | 1-7 of 16 columns

Checking the columns

```
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"
```

Same work for Sleep_day, also the same issues reappear.

```
head(sleep_day)
```

	Id	SleepDay	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
	<dbl>	<chr>	<int>	<int>	<int>
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

6 rows

```
colnames(sleep_day)
```

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

Understanding the ammount of parcitipants in the data.

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

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

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

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

Analyzing rows of data or sucesses.

```
nrow(daily_activity)
```

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

```
nrow(sleep_day)
```

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

Some stats, of the most important data.

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:

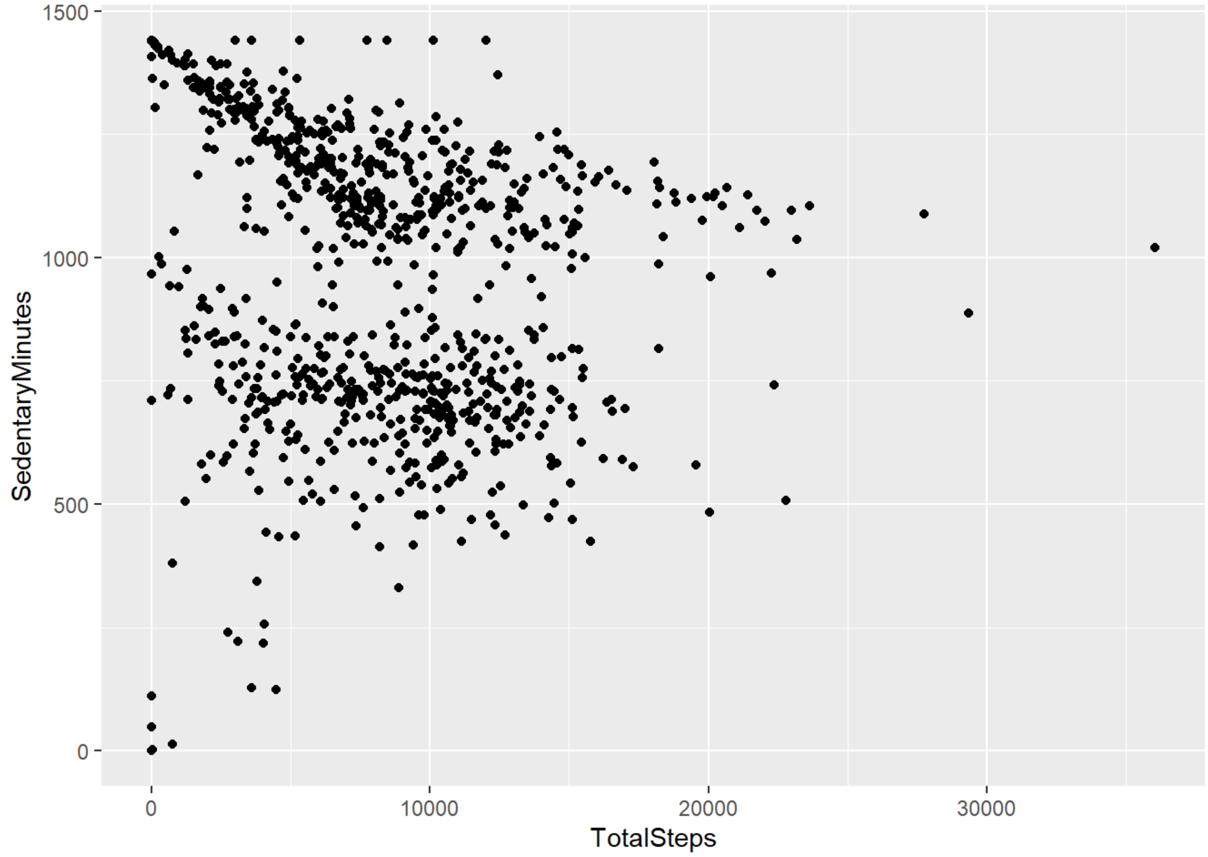
```
sleep_day %>%
  select(TotalSleepRecords,
  TotalMinutesAsleep,
  TotalTimeInBed) %>%
  summary()
```

```
##  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  
```

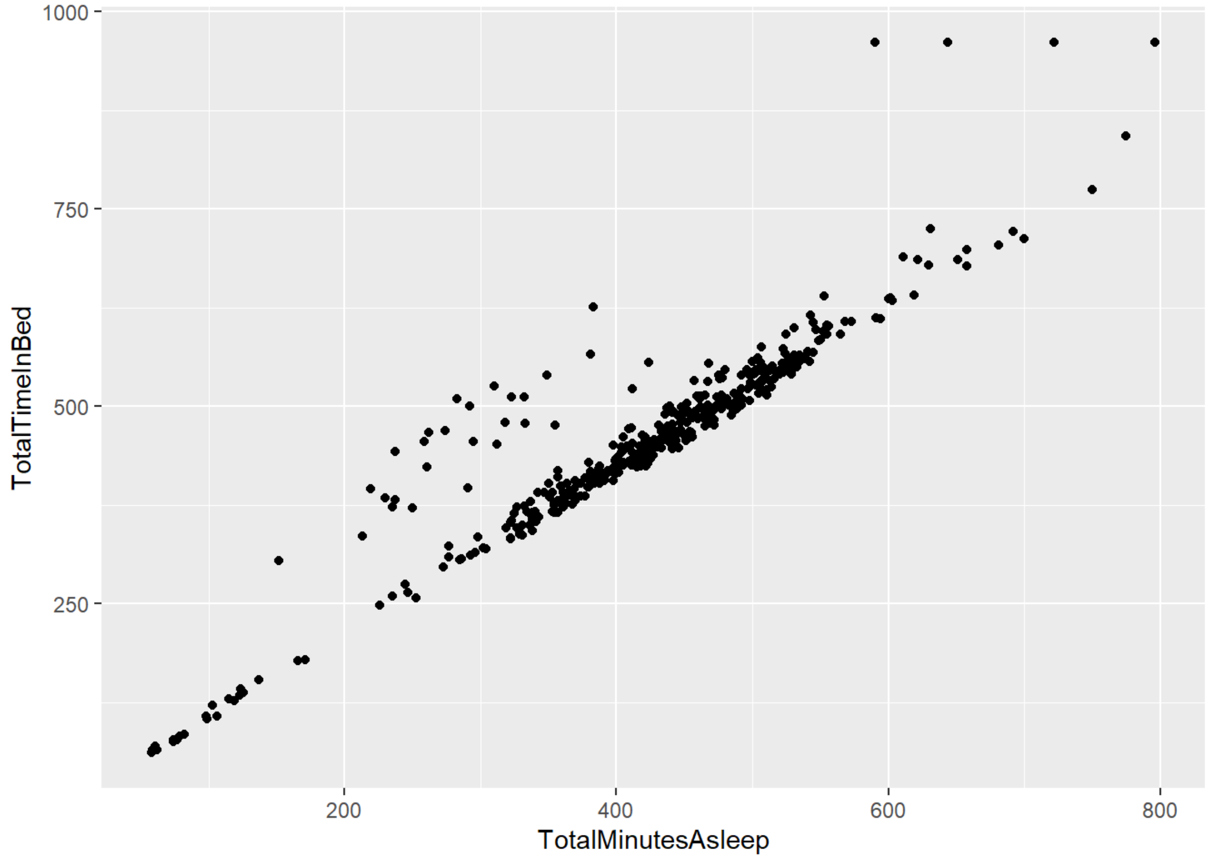
What does this tell us about how this sample of people’s activities?

Plotting a little for exploration

```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes)) + geom_point()
```



```
ggplot(data=sleep_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) + geom_point()
```



Found a problem with format of SleepDay and Activity Date, fixing it by extracting a part of the string to match. Good practice is to not modify the initial dataaset, so im creating v2.

Resources (https://stackoverflow.com/questions/17031002/get-weekdays-in-english-in-r (https://stackoverflow.com/questions/17031002/get-weekdays-in-english-in-r))

```
sleepday_v2 <- sleep_day %>% mutate(Date = substring(SleepDay,1,9))
daily_activity_v2 <- daily_activity %>% mutate(Date = daily_activity$ActivityDate)
```

```
sleepday_v2$Date <- as.Date(sleepday_v2$Date, "%m/%d/%Y") #The default format is yyyy-mm-dd
sleepday_v2$month <- format(as.Date(sleepday_v2$Date), "%m")
sleepday_v2$day <- format(as.Date(sleepday_v2$Date), "%d")
sleepday_v2$year <- format(as.Date(sleepday_v2$Date), "%Y")
sleepday_v2$day_of_week <- format(as.Date(sleepday_v2$Date), "%A")
daily_activity_v2$Date <- as.Date(daily_activity_v2$Date,"%m/%d/%Y") #The default format is yyyy-mm-dd
daily_activity_v2$month <- format(as.Date(daily_activity_v2$Date), "%m")
daily_activity_v2$day <- format(as.Date(daily_activity_v2$Date), "%d")
daily_activity_v2$year <- format(as.Date(daily_activity_v2$Date), "%Y")
daily_activity_v2$day_of_week <- format(as.Date(daily_activity_v2$Date), "%A")
```

I could add the time of day…but inspecting the data you can see that all entrys are taken on the same time of the day.

create breaks

#breaks <- hour(hm("00:00", "6:00", "12:00", "18:00", "23:59")) # labels for the breaks #labels <- c("Night", "Morning", "Afternoon", "Evening") ### Leavint the codes for learning purposes, using lubridate as library. Note: Data must be a date time column of course.

Merging, we have two data sets, IJ uses an inner join, keeping only rows matched in the two datasets. OJ sticks to the outer join concept, kepping all values and joining them if posible, I decided this approach as a complete view but having the leaking data of missing dates.

```
combined_data_ij <- merge(sleepday_v2, daily_activity_v2, by=c("Id","Date","month","day","year","day_of
_week"))
combined_data_oj <- merge(sleepday_v2, daily_activity_v2, by=c("Id","Date","month","day","year","day_of
_week"), all=TRUE)
```

Grouping data into a new data frame to analyze.

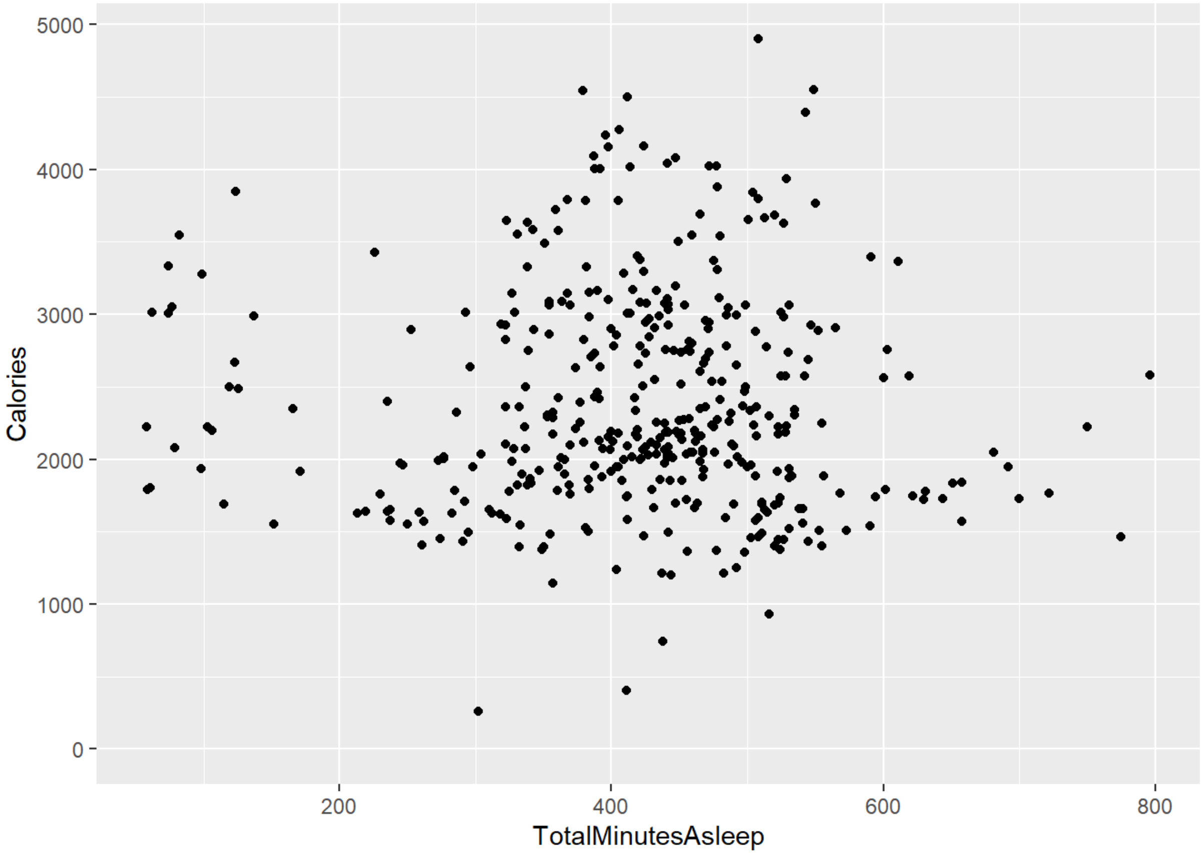
```
df <- combined_data_oj %>%
  mutate(weekday = wday(Date, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(weekday) %>%  #groups by usertype and weekday
  summarise(Observations = n()                          #calculates the number of rides and average dur
ation
  ,average_sleep_duration = mean(TotalTimeInBed), average_steps = mean(TotalSteps), average_distance =
mean(TotalDistance)) %>%         # calculates the average duration
  arrange(weekday)                              # sorts
df
```

weekday	Observations	average_sleep_duration	average_steps	average_distance
<ord>	<int>	<dbl>	<dbl>	<dbl>
Sun	121	NA	6933.231	5.027190
Mon	121	NA	7819.083	5.588347
Tue	152	NA	8125.007	5.832237
Wed	150	NA	7559.373	5.488333
Thu	148	NA	7420.682	5.326216
Fri	126	NA	7448.230	5.309921
Sat	125	NA	8202.712	5.901040

7 rows

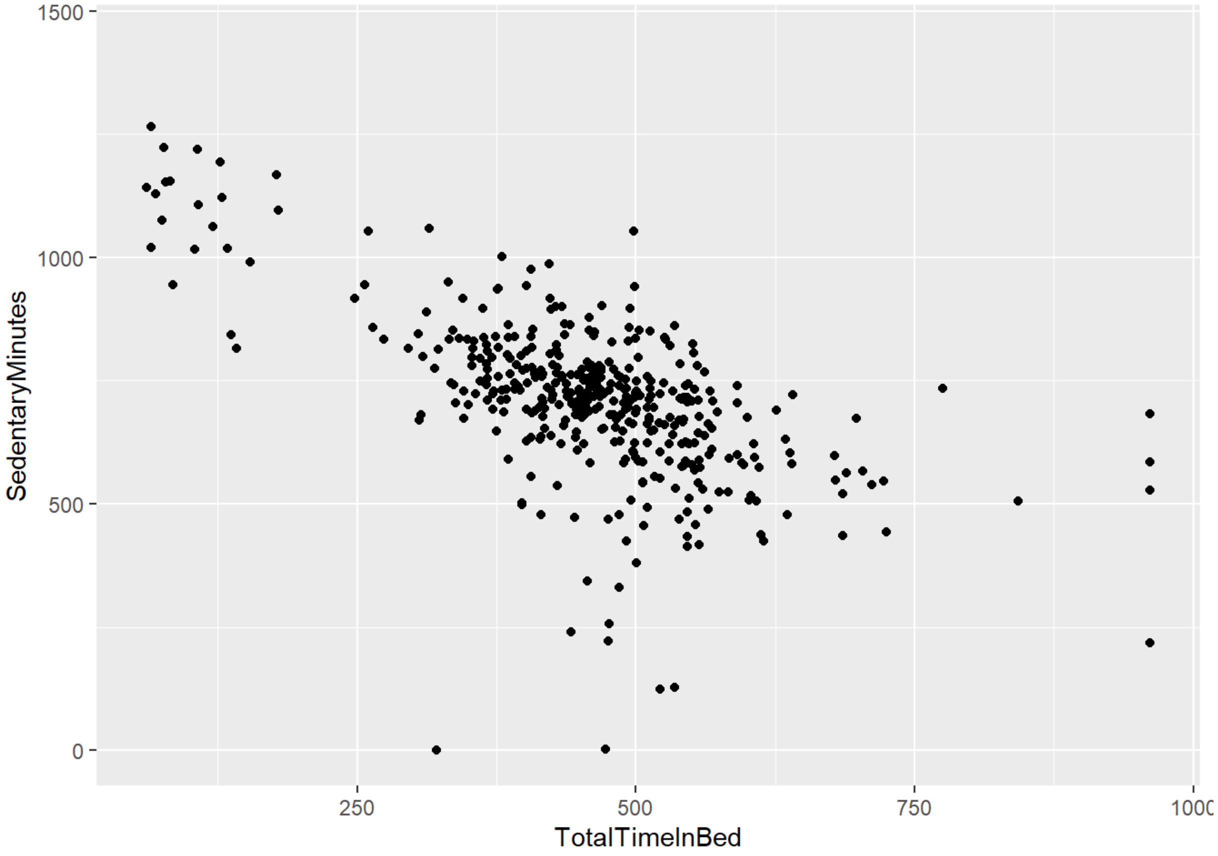
```
ggplot(data=combined_data_oj, aes(x=TotalMinutesAsleep, y=Calories)) + geom_point()
```

```
## Warning: Removed 530 rows containing missing values (`geom_point()`).
```



```
ggplot(data=combined_data_oj, aes(x=TotalTimeInBed, y=SedentaryMinutes)) + geom_point()
```

```
## Warning: Removed 530 rows containing missing values (`geom_point()`).
```


1500
1000
500
0
SedentaryMinutes
250
500
750
1000
TotalTimeInBed