Smart device usage insight

Terence Chau

2021-07-02

Introduction

The purpose of this report is to help the small company Bellabeat, a high-tech manufacturer of health-focused products for women, to become a larger player in the global smart device market. The management of the company believes that analyzing smart device fitness data could help unlock new growth opportunities for the company.

Data Overview

In this report, the public available dataset FitBit Fitness Tracker Data of 33 fitbit users with data between 2016-04-12 to 2016-05-12 have been used for the analysis.

Since data in the dataset is directly generated from fitbit smart devices, the data should be reliable. However, there are serveral limitation from the data:

- it only contains data of 33 users, which makes it a very small dataset for analysis
- no user information including sex information is available, where Bellabeat focus on women
- the data is from 2016 which is 5 years old, users behaviour may have changed
- no activity type is available, which makes it difficult to distinguish whether users are working out or having regular everyday life
- no device wearing time available, we assume all users wear their device all day

Given the above limitation, the insights from this report will have limited usefulness for Bellabeat which is target at women.

Data cleaning and processing

```
library(tidyverse)
```

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

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## 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(lubridate)
```

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

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

```
library(here)
```

```
## here() starts at D:/Projects/Courses/Google Data Analytics Professional Certificate/Course 8 - Googl
```

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

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

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

```
library(knitr)
```

6 csv files were imported into R for analysis.

```
daily_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()
## )
```

```r
hourly_calories <- read_csv("hourlyCalories_merged.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityHour = col_character(),
##   Calories = col_double()
## )
```

```r
hourly_intensities <- read_csv("hourlyIntensities_merged.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityHour = col_character(),
##   TotalIntensity = col_double(),
##   AverageIntensity = col_double()
## )
```

```r
hourly_steps <- read_csv("hourlySteps_merged.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityHour = col_character(),
##   StepTotal = col_double()
## )
```

```r
sleep_day <- read_csv("sleepDay_merged.csv")
```

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

```r
weight_log_info <- read_csv("weightLogInfo_merged.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   Date = col_character(),
##   WeightKg = col_double(),
##   WeightPounds = col_double(),
##   Fat = col_double(),
##   BMI = col_double(),
##   IsManualReport = col_logical(),
##   LogId = col_double()
## )
```

Let's take a look at the imported data frame.

```r
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```r
glimpse(hourly_calories)
```

```
## Rows: 22,099
## Columns: 3
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/20~
## $ Calories     <dbl> 81, 61, 59, 47, 48, 48, 48, 47, 68, 141, 99, 76, 73, 66, ~
```

```r
glimpse(hourly_intensities)
```

```
## Rows: 22,099
## Columns: 4
## $ Id               <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 15039~
## $ ActivityHour     <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/1~
## $ TotalIntensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 5~
## $ AverageIntensity <dbl> 0.333333, 0.133333, 0.116667, 0.000000, 0.000000, 0.0~
```

```r
glimpse(hourly_steps)
```

```
## Rows: 22,099
## Columns: 3
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityHour <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/12/20~
## $ StepTotal    <dbl> 373, 160, 151, 0, 0, 0, 0, 0, 250, 1864, 676, 360, 253, 2~
```

```r
glimpse(sleep_day)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```r
glimpse(weight_log_info)
```

```
## Rows: 67
## Columns: 8
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212~
## $ Date           <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, ~
## $ WeightPounds   <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6~
## $ Fat            <dbl> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,~
## $ IsManualReport <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, ~
## $ LogId          <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,~
```

Seems like there are little issues on data format on ID and date, it needed to be fixed.

```r
# fix data format for date and ID of daily_activity
daily_activity$ActivityDate <- as.Date(daily_activity$ActivityDate, "%m/%d/%y")
daily_activity$Id <- as.character(daily_activity$Id)

# fix data format for date and ID of hourly_calories
hourly_calories$ActivityHour <- mdy_hms(hourly_calories$ActivityHour)
hourly_calories$Id <- as.character(hourly_calories$Id)

# fix data format for date and ID of hourly_intensities
hourly_intensities$ActivityHour <- mdy_hms(hourly_intensities$ActivityHour)
hourly_intensities$Id <- as.character(hourly_intensities$Id)

# fix data format for date and ID of hourly_steps
hourly_steps$ActivityHour <- mdy_hms(hourly_steps$ActivityHour)
hourly_steps$Id <- as.character(hourly_steps$Id)

# fix data format for date and ID of sleep_day
sleep_day$SleepDay <- mdy_hms(sleep_day$SleepDay)
sleep_day$Id <- as.character(sleep_day$Id)

# fix data format for date and ID of weight_log_info
weight_log_info$Date <- mdy_hms(weight_log_info$Date)
weight_log_info$Id <- as.character(weight_log_info$Id)
```

Now check the number of users in each data frame.

```r
# check number of sample for daily_activity
n_distinct(daily_activity$Id)
```

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

```r
# check number of sample for hourly_calories
n_distinct(hourly_calories$Id)
```

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

```r
# check number of sample for hourly_intensities
n_distinct(hourly_intensities$Id)
```

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

```r
# check number of sample for hourly_steps
n_distinct(hourly_steps$Id)
```

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

```r
# check number of sample for sleep_day
n_distinct(sleep_day$Id)
```

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

```r
# check number of sample for weight_log_info
n_distinct(weight_log_info$Id)
```

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

Here we found that although 33 users in the dataset, only 24 recorded their sleep and 8 recorded their weight information. The number of sample for sleep and weight record are too small for generating a useful insights, but nevertheless we will still take a look on them.

As hourly__calories, hourly__intensities and hourly__steps have a very similar structure and a same number of observation, combine the 3 data frame into 1 for easier working.

```r
hourly <- merge(merge(hourly_calories, hourly_intensities, all=TRUE), hourly_steps, all=TRUE)
```

Separate the date and time in the hourly data frame.

```r
# separate time and date into their own column
hourly$Date <- format(hourly$ActivityHour, format = "%Y-%m-%d")
hourly$Time <- format(hourly$ActivityHour, format = "%H:%M:%S")
```

```r
# convert into date format
hourly$Date <- as_date(hourly$Date, "%Y-%m-%d", tz = NULL)
```

```r
# convert into time format
library(hms)
```

```
##
## Attaching package: 'hms'

## The following object is masked from 'package:lubridate':
##
##     hms
```

```r
hourly$Time <- as_hms(hourly$Time)
```

Let's check if the data frame are completed with data.

```r
skim_without_charts(daily_activity)
```

Table 1: Data summary

Name	daily_activity
Number of rows	940
Number of columns	15
Column type frequency:	
character	1
Date	1
numeric	13
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Id	0	1	10	10	0	33	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
ActivityDate	0	1	2020-04-12	2020-05-12	2020-04-26	31

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p10
TotalSteps	0	1	7637.91	5087.15	0	3789.75	7405.50	10727.00	36019.0
TotalDistance	0	1	5.49	3.92	0	2.62	5.24	7.71	28.0
TrackerDistance	0	1	5.48	3.91	0	2.62	5.24	7.71	28.0
LoggedActivitiesDistance	0	1	0.11	0.62	0	0.00	0.00	0.00	4.9
VeryActiveDistance	0	1	1.50	2.66	0	0.00	0.21	2.05	21.9
ModeratelyActiveDistance	0	1	0.57	0.88	0	0.00	0.24	0.80	6.4
LightActiveDistance	0	1	3.34	2.04	0	1.95	3.36	4.78	10.7
SedentaryActiveDistance	0	1	0.00	0.01	0	0.00	0.00	0.00	0.1
VeryActiveMinutes	0	1	21.16	32.84	0	0.00	4.00	32.00	210.0
FairlyActiveMinutes	0	1	13.56	19.99	0	0.00	6.00	19.00	143.0
LightlyActiveMinutes	0	1	192.81	109.17	0	127.00	199.00	264.00	518.0
SedentaryMinutes	0	1	991.21	301.27	0	729.75	1057.50	1229.50	1440.0
Calories	0	1	2303.61	718.17	0	1828.50	2134.00	2793.25	4900.0

```
skim_without_charts(hourly)
```

Table 5: Data summary

Name	hourly
Number of rows	22099
Number of columns	8
Column type frequency:	
character	1
Date	1
difftime	1
numeric	4
POSIXct	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Id	0	1	10	10	0	33	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Date	0	1	2016-04-12	2016-05-12	2016-04-26	31

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Time	0	1	0 secs	82800 secs	11:00:00	24

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Calories	0	1	97.39	60.70	42	63	83.00	108.00	948
TotalIntensity	0	1	12.04	21.13	0	0	3.00	16.00	180
AverageIntensity	0	1	0.20	0.35	0	0	0.05	0.27	3
StepTotal	0	1	320.17	690.38	0	0	40.00	357.00	10554

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
ActivityHour	0	1	2016-04-12	2016-05-12 15:00:00	2016-04-26 06:00:00	736

```r
skim_without_charts(sleep_day)
```

Table 11: Data summary

Name	sleep_day
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	3
POSIXct	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Id	0	1	10	10	0	24	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TotalSleepRecords	0	1	1.12	0.35	1	1	1	1	3
TotalMinutesAsleep	0	1	419.47	118.34	58	361	433	490	796
TotalTimeInBed	0	1	458.64	127.10	61	403	463	526	961

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
SleepDay	0	1	2016-04-12	2016-05-12	2016-04-27	31

```r
skim_without_charts(weight_log_info)
```

Table 15: Data summary

Name	weight_log_info
Number of rows	67
Number of columns	8
Column type frequency:	
character	1
logical	1

numeric	5
POSIXct	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Id	0	1	10	10	0	8	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
IsManualReport	0	1	0.61	TRU: 41, FAL: 26

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
WeightKg	0	1.00	7.204000e+01	13.92	5.260000e+01	6.140000e+01	6.250000e+01	8.505000e+01	1.335000e+02
WeightPounds	0	1.00	1.588100e+02	30.70	1.159600e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02
Fat	65	0.03	2.350000e+01	2.12	2.200000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+01
BMI	0	1.00	2.519000e+01	3.07	2.145000e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01
LogId	0	1.00	1.461772e+12	782994783.6	1.460444e+12	1.461079e+12	1.461802e+12	1.462375e+12	1.463098e+12

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Date	0	1	2016-04-12 06:47:11	2016-05-12 23:59:59	2016-04-27 23:59:59	56

Except weight_log_info data frame with missing data in the column FAT, all other columns in other data frames are filled with data.

Data analyze and insights

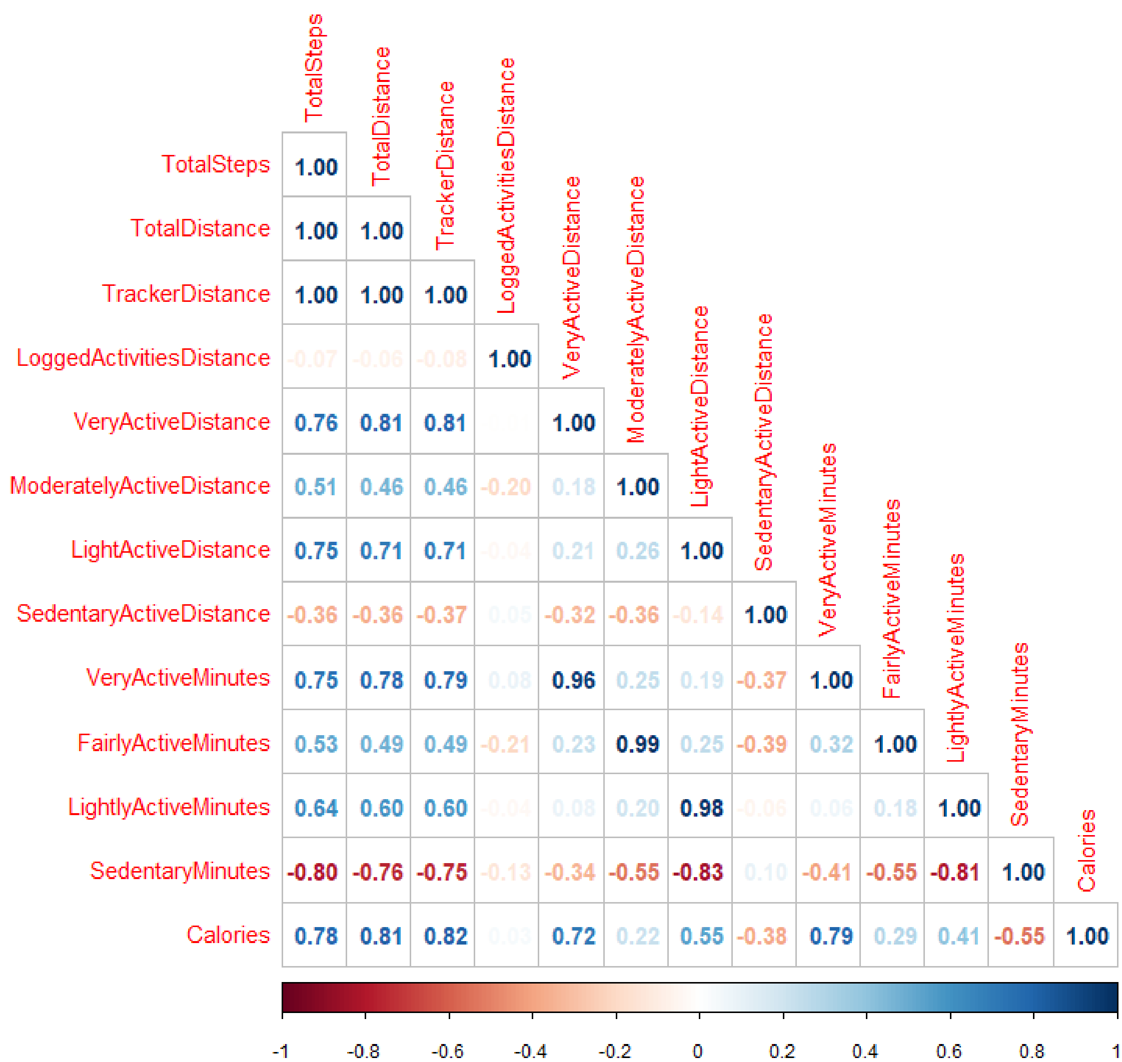
First let's focus on the daily_activity data frame.

Check how the values within daily_activity are correlated to each other.

```
daily_corr <- cor(daily_activity %>% select(-Id, -ActivityDate)) %>%  round(digits = 2)
```

Visualize the correlation for easier understanding.

```
library(corrplot)
png(height=800, width=800, pointsize=15, file="corrplot.png")
corrplot(cor(daily_corr), method = "number", type = "lower")
```



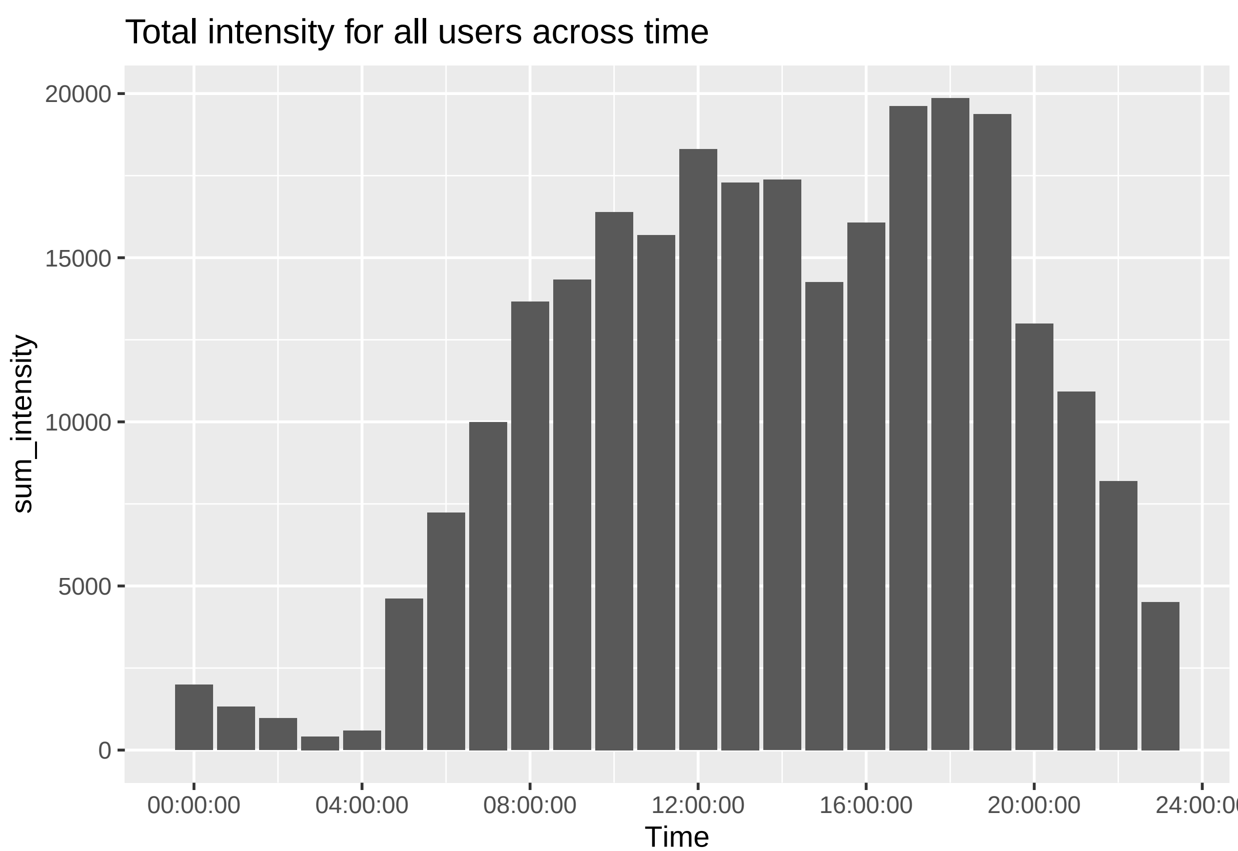
Some insights from the correlation heatmap:

- Total steps and total distance are directly related to each other

- The intensities of active minutes are highly related to the intensities of active distance
- Sedentary Minutes are negative correlated to total steps and total distance
- Calories is positively correlated to total steps and total distance
- Surprisingly, lightly active distance and lightly active minutes have a higher correlation to total steps and total steps then moderately active distance and fairly active minutes, perhaps those users who walk a lot covers more mileage but being categorize into lightly active distance / minutes, but the exact reason is not known, and here is the data limitation of no activity type of users available.

Now let's check the hourly data frame and see how user's activity spread across a day by grouping their total activity intensities into time.

```
hourly %>% group_by(Time) %>% summarise(sum_intensity = sum(TotalIntensity)) %>%
  ggplot(aes(x=Time, y=sum_intensity)) + geom_col() +
  labs(title = "Total intensity for all users across time")
```



From the graph we find that users tend to be active on day time and decrease activity at night, probably going to sleep at night.

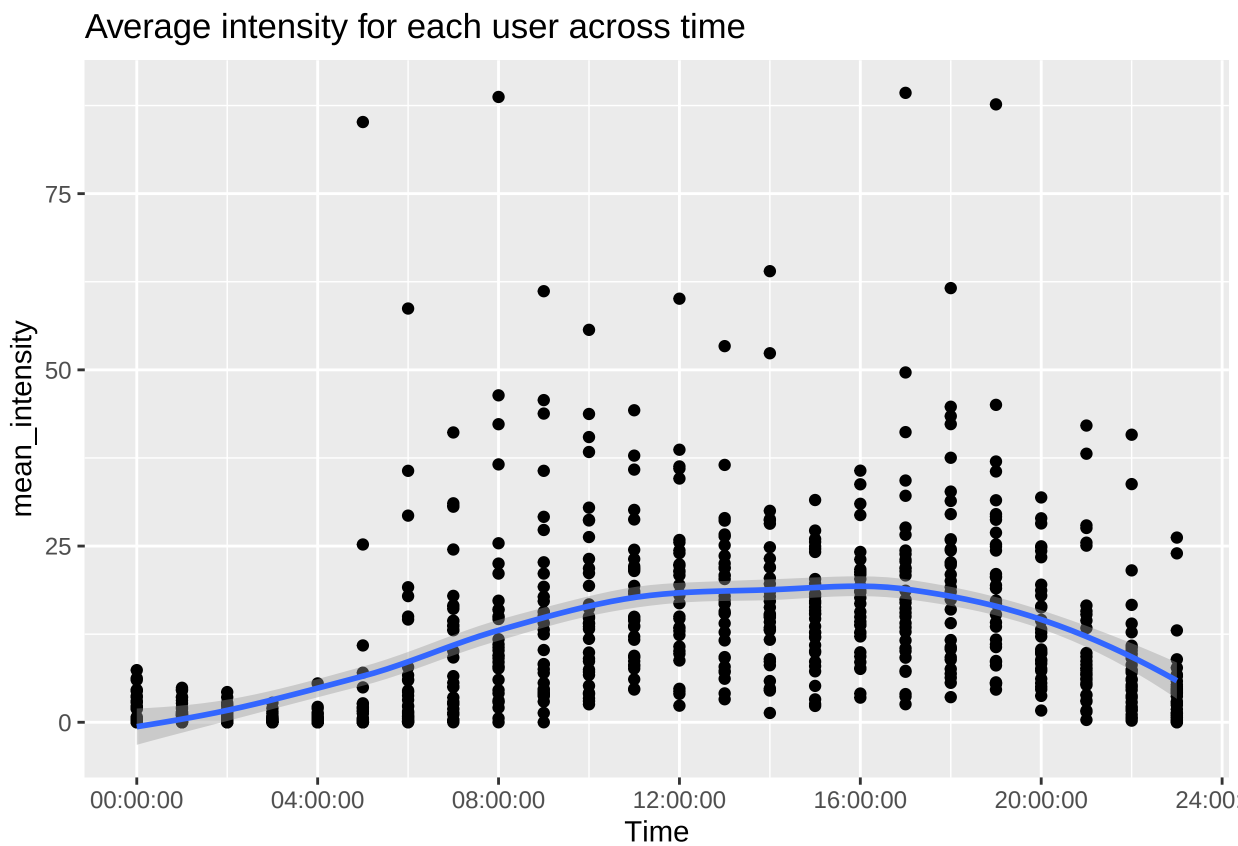
The highest intensity by time is at 5-7pm, probably users tends to workout after work.

Let's check the average intensity by users in hour

```
hourly %>% group_by(Id, Time) %>% summarise(mean_intensity = mean(TotalIntensity)) %>%
  ggplot(aes(x=Time, y=mean_intensity)) + geom_point() + geom_smooth() +
  labs(title = "Average intensity for each user across time")
```

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

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

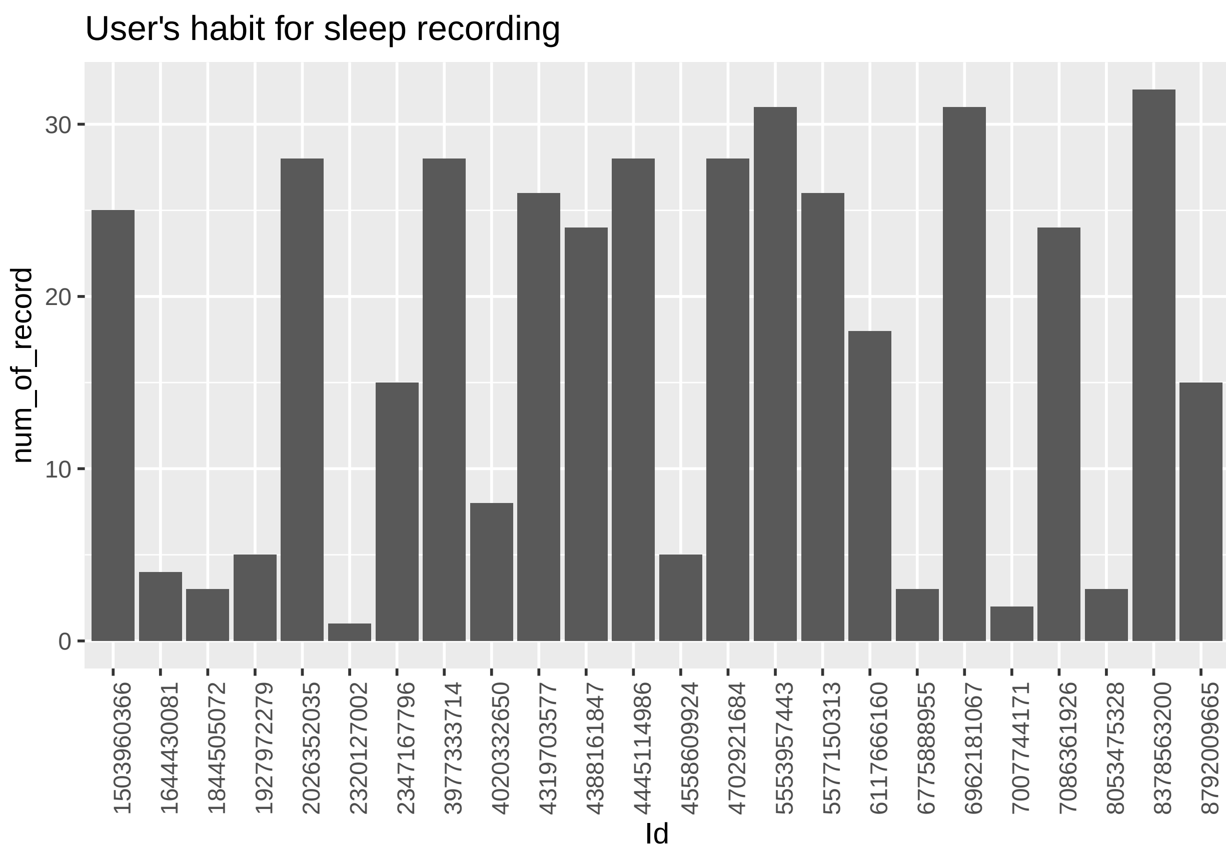


Here we can see although the overall trends agree with the graph above, intensities during active hour varies greatly among users, likely to be the different workout intensity among users.

Now let's focus on the sleeping data. As discovered above, only 24 users record their sleep. Let's check how often they record their sleep.

```
# habit to record sleep
sleep_record_habit <- sleep_day %>% count(Id)

# Visualize the sleep recording habit for different users
ggplot(data = sleep_record_habit, aes(x = Id, y = n)) + geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "User's habit for sleep recording") + ylab("num_of_record")
```



Here we found that most users didn't record all their sleep within a month, many just record a few.

Let's check the average sleep time of user, and add the asleep percentage and the number of sleep record from above.

```
# average sleep time
sleep_anaylyze <- sleep_day %>% group_by(Id) %>% summarise(mean_asleep = mean(TotalMinutesAsleep), mean_

# asleep percentage
sleep_anaylyze$asleep_percentage <- (sleep_anaylyze$mean_asleep / sleep_anaylyze$mean_bed) * 100

# add the number of sleep record for comparison
sleep_anaylyze$num_of_record <- sleep_record_habit$n

head(sleep_anaylyze)
```

```
## # A tibble: 6 x 5
##   Id         mean_asleep mean_bed asleep_percentage num_of_record
##   <chr>            <dbl>    <dbl>             <dbl>         <int>
## 1 1503960366        360.     383.              94.0            25
## 2 1644430081        294      346               85.0             4
## 3 1844505072        652      961               67.8             3
## 4 1927972279        417      438.              95.2             5
## 5 2026352035        506.     538.              94.1            28
## 6 2320127002         61       69               88.4             1
```

Let's convert the unit from minutes to hour and filter out those with too little record and only keep those who record at least 2/3 of their sleep in a month, assuming those with at least 21 record meet the requirement.

```
# filter for only users with 21 or more sleep records
sleep_analyze_trim <- sleep_anaylyze %>% filter(num_of_record >= 21)

# convert into hour
sleep_analyze_trim$mean_asleep_hour <- sleep_analyze_trim$mean_asleep / 60
sleep_analyze_trim$mean_bed_hour <- sleep_analyze_trim$mean_bed / 60

# shows only essential columns
sleep_analyze_output <- sleep_analyze_trim %>% select(Id, mean_asleep_hour, mean_bed_hour, asleep_percen

kable(sleep_analyze_output)
```

Id	mean_asleep_hour	mean_bed_hour	asleep_percentage
1503960366	6.004667	6.386667	94.01879
2026352035	8.436310	8.960714	94.14773
3977333714	4.894048	7.685714	63.67720
4319703577	7.944231	8.366026	94.95824
4388161847	6.718750	7.103472	94.58403
4445114986	6.419643	6.947024	92.40853
4702921684	7.019048	7.366071	95.28889
5553957443	7.724731	8.431183	91.62097
5577150313	7.200000	7.676923	93.78758
6962181067	7.466667	7.768817	96.11073
7086361926	7.552083	7.773611	97.15026
8378563200	7.389062	8.055208	91.73025

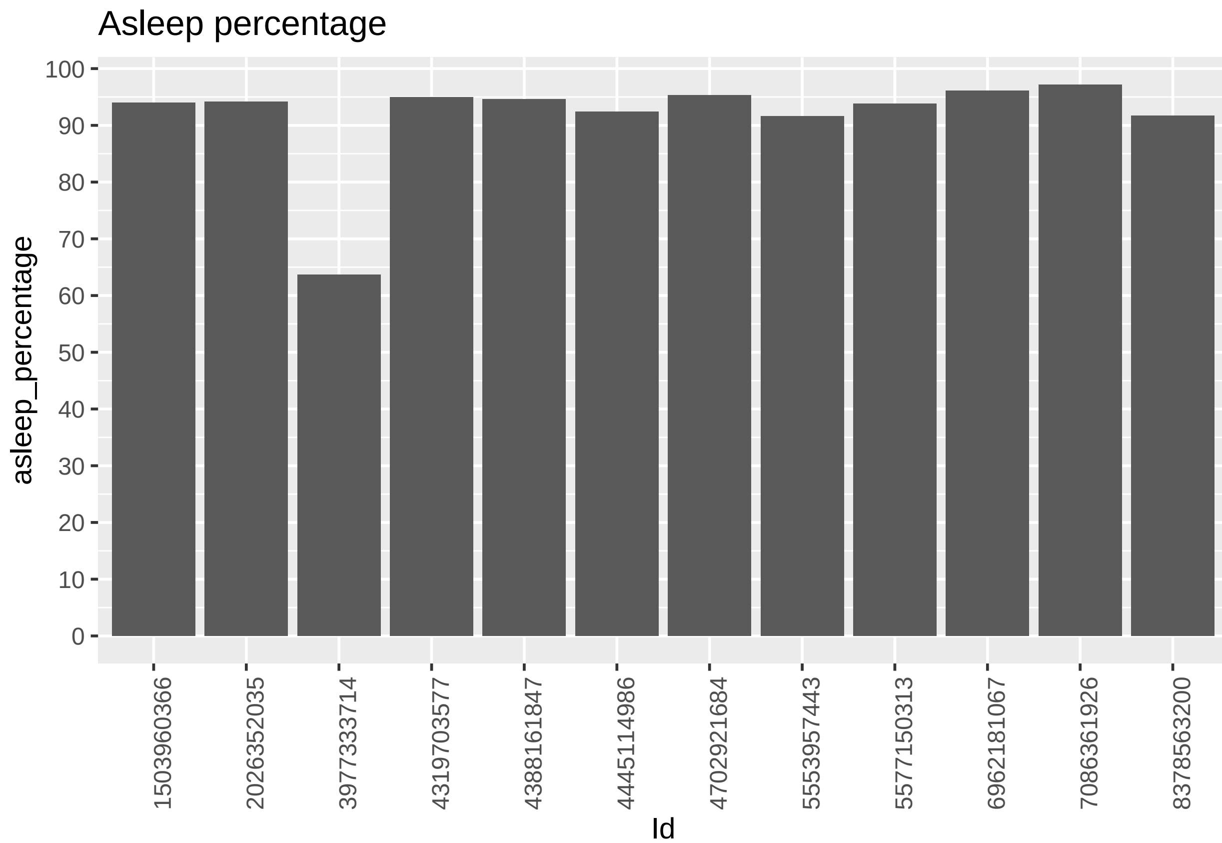
```
n_distinct(sleep_analyze_output$Id)
```

```
## [1] 12
```

Only 12 users record over 21 sleeping records.

Let's see how is the sleeping quality of this 12 users.

```
ggplot(data = sleep_analyze_output, aes(x = Id, y = asleep_percentage)) + geom_col() +
  labs(title = "Asleep percentage") + theme(axis.text.x = element_text(angle = 90)) +
  scale_y_continuous(breaks = seq(0,100,10))
```



Most user fall asleep for over 90% of time when they are on the bed.

Also check the average time on bed and average time of asleep.

```
# average time on bed
mean(sleep_analyze_output$mean_bed_hour)
```

```
## [1] 7.710119
```

```
# average time asleep
mean(sleep_analyze_output$mean_asleep_hour)
```

```
## [1] 7.064103
```

According to the Centers for Disease Control and Prevention, sleeping time for adult should have above 7 hours. From the data, the average on bed time is 7.71 hours, and the average asleep time is 7.06 hours. Yet with only 12 valid samples, the number of samples are too little to draw meaningful conclusion.

Lastly let's check the weight log although there are only 8 users recorded their weight.

The latest record of BMI for each users were used.

```r
# latest BMI for individuals
individuals_BMI <- weight_log_info %>% group_by(Id) %>% slice_tail(n=1) %>% select(Id, BMI)
```

Let's also see the how healthy among this small sample of users

```r
# number of sample for BMI over 25
nrow(filter(individuals_BMI, BMI > 25))
```

```
## [1] 5
```

With BMI higher than 25 classified as overweight, 5 out of 8 users are overweight!

Summary insight

As sleep and weight data contains too few samples, they are not able to provide conclusive insights.

From the above insights,

- Smart device users activity is higher at day time and lower at early morning and late night.
- Smart device users probably prefer to workout after work, between 5-7pm.
- Smart device users differs a lot in terms of activity intensity, and those work hard and have high intensity during workout is minority, which suggest that the majority of smart device users are casual users.
- Most smart device users didn't record all their sleep with a month, many just record sleep occasionally.
- Only small proportion of smart device users record their weight information.

Recommendation

1. Smart device users are not fully utilize the function of their device, we could encourage users to record their sleep and weight to monitor and improve their health by:
    - Develop new device which is comforatble to wear during sleep
    - Adopt push notification to remind users to record their weight information
    - Educate existing and potential users of the benifit of knowing their own body statistic including activity intensity, sleep and weight information.
2. Develop a greater range of device so that not only casual user, but enthusiastic or even professional users will find our smart device are useful and helpful for them.