Capstone Project: Analysing Smart Device Usage Data in R

Overview

This Case Study is a Capstone project for the Google Data Analytics Course by Coursera. For the purpose of this case study, I am a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

The goal of this project is to analyse smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. Then, using this information to provide the high-level recommendations for how these trends can inform Bellabeat marketing strategy.

Prepare data for exploration

The dataset being used is public data that explores smart device users' daily habits: 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 dataset for analysis is stored locally for the purpose of this project. The data is original, comprehensive and cited. However, it includes a small sample, a small period of data collection (between 03.12.2016-05.12.2016.) and it doesn't show any demographic information, which translates in highly possibly biased data. Another limitation is that the data is not current (2016).

Processing Data

Tools used: **R**

Installing and loading common packages and libraries

```
install.packages('tidyverse')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("tidyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyr)
library(ggplot2)
library(janitor)
```

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

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(lubridate)
```

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

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

```
library(tidyverse)
```

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

```
## v tibble  3.1.6     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

Loading your CSV files

Here we'll create a dataframe named 'daily_activity' and read in one of the CSV files from the dataset.

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

We'll create another dataframe for the sleep data.

```
sleep_day <- read.csv("sleepDay_merged.csv")
minute_sleep <- read.csv("minuteSleep_merged.csv")
```

We'll create another dataframe for the step and calories data.

```
daily_steps <- read.csv("dailySteps_merged.csv")
hourly_steps <- read.csv("hourlySteps_merged.csv")
daily_calories <- read.csv("dailyCalories_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
```

Exploring a few key tables

- Take a look at the daily_activity data.

```
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366      4/12/16      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
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 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
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 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
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes 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
```

- 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)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 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
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

- Identify all the columsn in the sleep_day data.

```
colnames(sleep_day)
```

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

- Take a look at the minute_sleep data.

```
head(minute_sleep)
```

```
##           Id                  date value       logId
## 1 1503960366 4/12/2016 2:47:30 AM     3 11380564589
## 2 1503960366 4/12/2016 2:48:30 AM     2 11380564589
## 3 1503960366 4/12/2016 2:49:30 AM     1 11380564589
## 4 1503960366 4/12/2016 2:50:30 AM     1 11380564589
## 5 1503960366 4/12/2016 2:51:30 AM     1 11380564589
## 6 1503960366 4/12/2016 2:52:30 AM     1 11380564589
```

- Identify all the columsn in the minute_sleep data.

```
colnames(minute_sleep)
```

```
## [1] "Id"    "date"  "value" "logId"
```

- Take a look at the daily_steeps data

```
head(daily_steps)
```

```
##           Id ActivityDay StepTotal
## 1 1503960366     4/12/16     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
```

- Identify all the columsn in the daily_steps data.

```
colnames(daily_steps)
```

```
## [1] "Id"          "ActivityDay" "StepTotal"
```

- Take a look at the daily_calories data

```
head(daily_calories)
```

```
##           Id ActivityDay Calories
## 1 1503960366     4/12/16     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
```

- Identify all the columsn in the daily_calories data

```
colnames(daily_calories)
```

```
## [1] "Id"          "ActivityDay" "Calories"
```

Note that both datasets have the 'Id' field - this can be used to merge the datasets.

We notice that the daily_activity dataframe already includes data in the daily_calories and daily_steps dataframe. Thus, we remove these two dataframes.

```
rm(daily_calories,daily_steps)
```

Structure of the dataframes

- First, we take a look ar the dataframes daily_activity and sleep_day

```
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/16", "4/13/2016", "4/14/2016", "4/15/20~
## $ TotalSteps               <int> 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        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
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  <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

- Check for duplicates

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

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

```
sum(duplicated(sleep_day))
```

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

- Remove duplicate values in sleep_day data

```
sleep_day <- sleep_day %>%
  distinct()
```

- Remove rows with empty fields

```
hourly_steps <- hourly_steps %>%
  drop_na(ActivityHour)

hourly_calories <- hourly_calories %>%
  drop_na(ActivityHour)
```

- Correct format of the date columns

```
daily_activity <- daily_activity %>%
  mutate(ActivityDate = as_date(ActivityDate, format = "%m/%d/%Y"))
```

```
sleep_day <- sleep_day %>%
  mutate(SleepDay = as_date(SleepDay , format = "%m/%d/%Y"))
```

```
hourly_steps$date <- as_date(mdy_hms(hourly_steps$ActivityHour))
hourly_steps$time<- format(as.POSIXct(mdy_hms(hourly_steps$ActivityHour)), format = "%H:%M")
hourly_steps$time <- hour(mdy_hms(hourly_steps$ActivityHour)) #%>% hour(hourly_steps$ActivityHour)
hourly_steps$day<- weekdays(hourly_steps$date)
head(hourly_steps)
```

```
##           Id          ActivityHour StepTotal       date time     day
## 1 1503960366 4/12/2016 12:00:00 AM       373 2016-04-12    0 Tuesday
## 2 1503960366  4/12/2016 1:00:00 AM       160 2016-04-12    1 Tuesday
## 3 1503960366  4/12/2016 2:00:00 AM       151 2016-04-12    2 Tuesday
## 4 1503960366  4/12/2016 3:00:00 AM         0 2016-04-12    3 Tuesday
## 5 1503960366  4/12/2016 4:00:00 AM         0 2016-04-12    4 Tuesday
## 6 1503960366  4/12/2016 5:00:00 AM         0 2016-04-12    5 Tuesday
```

```
hourly_calories$date <- as_date(mdy_hms(hourly_calories$ActivityHour))
hourly_calories$time <- format(as.POSIXct(mdy_hms(hourly_calories$ActivityHour)),format = "%H:%M")
hourly_calories$time <- hour(mdy_hms(hourly_calories$ActivityHour))
hourly_calories$day <- weekdays(hourly_calories$date)
```

```
head(hourly_calories)
```

```
##           Id          ActivityHour Calories       date time     day
## 1 1503960366 4/12/2016 12:00:00 AM       81 2016-04-12    0 Tuesday
## 2 1503960366  4/12/2016 1:00:00 AM       61 2016-04-12    1 Tuesday
## 3 1503960366  4/12/2016 2:00:00 AM       59 2016-04-12    2 Tuesday
## 4 1503960366  4/12/2016 3:00:00 AM       47 2016-04-12    3 Tuesday
## 5 1503960366  4/12/2016 4:00:00 AM       48 2016-04-12    4 Tuesday
## 6 1503960366  4/12/2016 5:00:00 AM       48 2016-04-12    5 Tuesday
```

- Fixing format

```
d <- unique(daily_activity$ActivityDate)
print(d)
```

```
##  [1] "16-04-12"   "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
##  [6] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [11] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [16] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "16-05-01"
## [21] "16-05-02"   "16-05-03"   "16-05-04"   "16-05-05"   "16-05-06"
## [26] "16-05-07"   "16-05-08"   "16-05-09"   "16-05-10"   "16-05-11"
## [31] "16-05-12"
```

```
class(daily_activity$ActivityDate)
```

```
## [1] "Date"
```

```
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366     16-04-12      13162          8.50            8.50
## 2 1503960366   2016-04-13      10735          6.97            6.97
## 3 1503960366   2016-04-14      10460          6.74            6.74
## 4 1503960366   2016-04-15       9762          6.28            6.28
## 5 1503960366   2016-04-16      12669          8.16            8.16
## 6 1503960366   2016-04-17       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 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
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 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
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes 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
```

- Check the unique participants are there in each dataframe

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

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

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

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

- Check number of observationn in each dataframe

```
nrow(daily_activity)
```

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

```
nrow(sleep_day)
```

```
## [1] 410
```

Merging these two datasets together

- Rename the columns with date to a same name

```
daily_activity <-  daily_activity %>%
  rename(date = ActivityDate)

sleep_day <-  sleep_day %>%
  rename(date = SleepDay)
```

***Note**: There were more participant Ids in the daily_activity dataset than in sleep_day dataset that lead to some Ids in daily_activity have been filtered out using merge.*

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

```
glimpse(combined_data)
```

```
## Rows: 251
## Columns: 18
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ date                     <date> 2016-04-13, 2016-04-15, 2016-04-16, 2016-04-~
## $ TotalSleepRecords        <int> 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep       <int> 384, 412, 340, 700, 304, 360, 325, 361, 430, ~
## $ TotalTimeInBed           <int> 407, 442, 367, 712, 320, 377, 364, 384, 449, ~
## $ TotalSteps               <int> 10735, 9762, 12669, 9705, 15506, 10544, 9819,~
## $ TotalDistance            <dbl> 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.34, 9.0~
## $ TrackerDistance          <dbl> 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.34, 9.0~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.34, 2.8~
## $ ModeratelyActiveDistance <dbl> 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.35, 0.8~
## $ LightActiveDistance      <dbl> 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.65, 5.3~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <int> 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 31, 4~
## $ FairlyActiveMinutes      <int> 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23, 28,~
## $ LightlyActiveMinutes     <int> 217, 209, 221, 164, 264, 205, 211, 262, 238, ~
## $ SedentaryMinutes         <int> 776, 726, 773, 539, 775, 818, 838, 732, 709, ~
## $ Calories                 <int> 1797, 1745, 1863, 1728, 2035, 1786, 1775, 194~
```

- Take a look at how many participants are in this data set

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

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

- Add weekday column to comnbined_data

```
final_daily <- combined_data
final_daily$weekday <- weekdays(final_daily$date)
glimpse(final_daily)
```

```
## Rows: 251
## Columns: 19
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ date                     <date> 2016-04-13, 2016-04-15, 2016-04-16, 2016-04-~
## $ TotalSleepRecords        <int> 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep       <int> 384, 412, 340, 700, 304, 360, 325, 361, 430, ~
## $ TotalTimeInBed           <int> 407, 442, 367, 712, 320, 377, 364, 384, 449, ~
## $ TotalSteps               <int> 10735, 9762, 12669, 9705, 15506, 10544, 9819,~
## $ TotalDistance            <dbl> 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.34, 9.0~
## $ TrackerDistance          <dbl> 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.34, 9.0~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.34, 2.8~
## $ ModeratelyActiveDistance <dbl> 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.35, 0.8~
## $ LightActiveDistance      <dbl> 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.65, 5.3~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <int> 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 31, 4~
## $ FairlyActiveMinutes      <int> 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23, 28,~
## $ LightlyActiveMinutes     <int> 217, 209, 221, 164, 264, 205, 211, 262, 238, ~
## $ SedentaryMinutes         <int> 776, 726, 773, 539, 775, 818, 838, 732, 709, ~
## $ Calories                 <int> 1797, 1745, 1863, 1728, 2035, 1786, 1775, 194~
## $ weekday                  <chr> "Wednesday", "Friday", "Saturday", "Sunday", ~
```

Analyse and Share Phases

Summary statistics

- **For the daily activity dataset:**

```
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 dataset:**

```
sleep_day %>%
  select(TotalSleepRecords,
```

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

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.00      Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.00      1st Qu.:361.0      1st Qu.:403.8
##  Median :1.00      Median :432.5      Median :463.0
##  Mean   :1.12      Mean   :419.2      Mean   :458.5
##  3rd Qu.:1.00      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.00      Max.   :796.0      Max.   :961.0
```

```
summary(final_daily)
```

```
##        Id                 date            TotalSleepRecords TotalMinutesAsleep
##  Min.   :1.504e+09   Min.   :2016-04-13   Min.   :1.000     Min.   : 59.0
##  1st Qu.:3.977e+09   1st Qu.:2016-04-17   1st Qu.:1.000     1st Qu.:361.0
##  Median :4.703e+09   Median :2016-04-22   Median :1.000     Median :430.0
##  Mean   :4.962e+09   Mean   :2016-04-21   Mean   :1.127     Mean   :418.5
##  3rd Qu.:6.776e+09   3rd Qu.:2016-04-26   3rd Qu.:1.000     3rd Qu.:487.0
##  Max.   :8.792e+09   Max.   :2016-04-30   Max.   :3.000     Max.   :775.0
##  TotalTimeInBed    TotalSteps    TotalDistance    TrackerDistance
##  Min.   : 65.0   Min.   :   42   Min.   : 0.030   Min.   : 0.030
##  1st Qu.:406.0   1st Qu.: 5204   1st Qu.: 3.620   1st Qu.: 3.620
##  Median :461.0   Median : 9105   Median : 6.280   Median : 6.280
##  Mean   :457.3   Mean   : 8583   Mean   : 6.077   Mean   : 6.069
##  3rd Qu.:522.0   3rd Qu.:11390   3rd Qu.: 8.065   3rd Qu.: 8.055
##  Max.   :961.0   Max.   :22359   Max.   :17.190   Max.   :17.190
##  LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
##  Min.   :0.00000          Min.   : 0.000     Min.   :0.0000
##  1st Qu.:0.00000          1st Qu.: 0.000     1st Qu.:0.0000
##  Median :0.00000          Median : 0.560     Median :0.4200
##  Mean   :0.09596          Mean   : 1.501     Mean   :0.7445
##  3rd Qu.:0.00000          3rd Qu.: 2.465     3rd Qu.:1.0350
##  Max.   :4.08169          Max.   :12.540     Max.   :5.1200
##  LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
##  Min.   :0.030       Min.   :0.0000000       Min.   :  0.00
##  1st Qu.:2.535       1st Qu.:0.0000000       1st Qu.:  0.00
##  Median :3.690       Median :0.0000000       Median :  9.00
##  Mean   :3.785       Mean   :0.0008366       Mean   : 26.08
##  3rd Qu.:4.910       3rd Qu.:0.0000000       3rd Qu.: 36.00
##  Max.   :9.480       Max.   :0.1100000       Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   : 0.00       Min.   :  4.0        Min.   :   2.0   Min.   : 403
##  1st Qu.: 0.00       1st Qu.:159.0        1st Qu.: 646.0   1st Qu.:1881
##  Median :12.00       Median :206.0        Median : 721.0   Median :2200
##  Mean   :18.23       Mean   :217.0        Mean   : 724.0   Mean   :2415
##  3rd Qu.:28.00       3rd Qu.:263.5        3rd Qu.: 781.5   3rd Qu.:2908
##  Max.   :98.00       Max.   :518.0        Max.   :1265.0   Max.   :4900
##    weekday
##  Length:251
##  Class :character
##  Mode  :character
##
```

##
##

- **The average:**

```
daily_average <- combined_data %>%
  group_by(Id) %>%
  summarise(average_steps = mean(TotalSteps), average_calories = mean(Calories), average_minutes_sleep =
head(daily_average)
```

```
## # A tibble: 6 x 5
##           Id average_steps average_calories average_minutes_sl~ average_time_in~
##        <dbl>         <dbl>            <dbl>               <dbl>            <dbl>
## 1 1503960366        12233.            1866.                375.             399.
## 2 1644430081        10694.            3172                 122.             134.
## 3 1844505072         3929             1744                 683              961
## 4 1927972279         1693             2340.                334.             354.
## 5 2026352035         4826.            1507.                511.             548.
## 6 2320127002         5079             1804                  61               69
```

```
summary(daily_average)
```

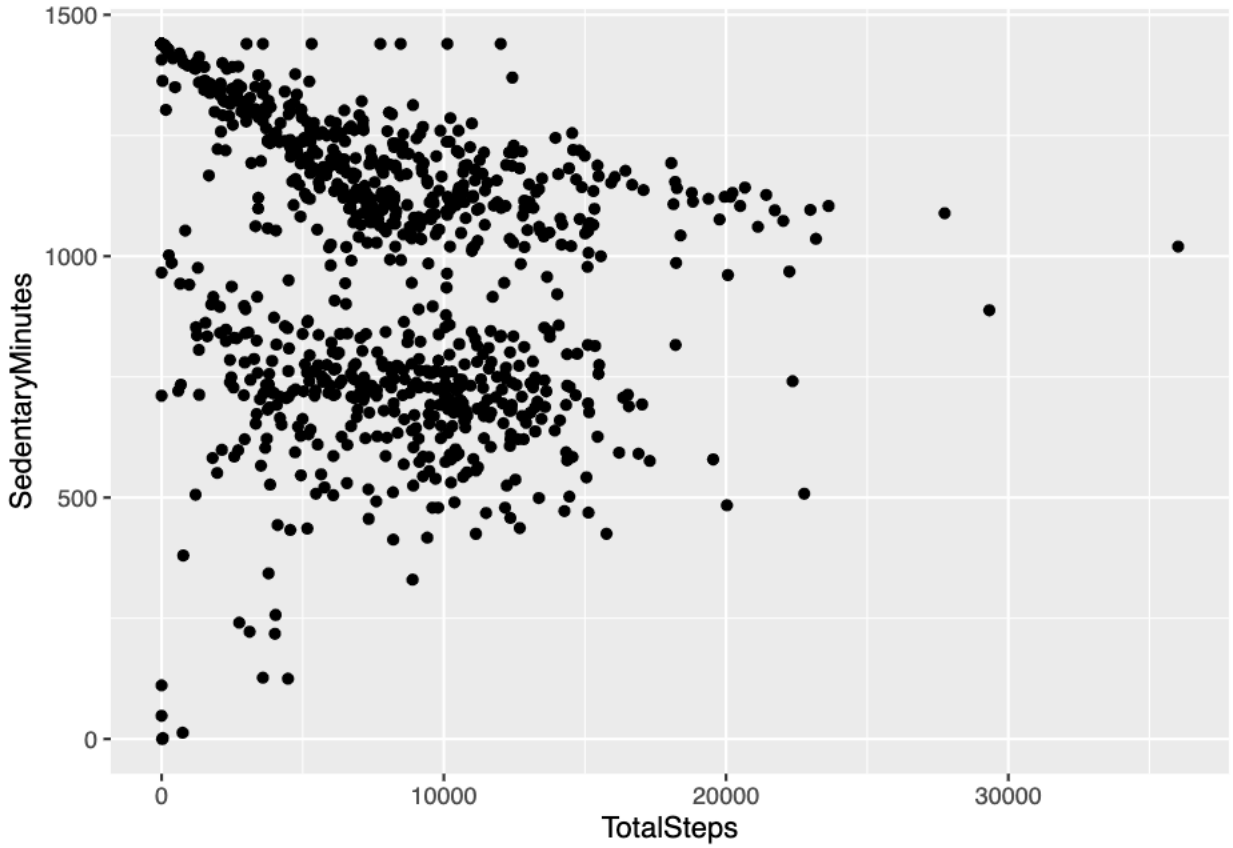
```
##        Id            average_steps   average_calories average_minutes_sleep
##  Min.   :1.504e+09   Min.   : 1693   Min.   :1507     Min.   : 61.0
##  1st Qu.:2.340e+09   1st Qu.: 4598   1st Qu.:1953     1st Qu.:325.0
##  Median :4.502e+09   Median : 8959   Median :2275     Median :414.4
##  Mean   :4.764e+09   Mean   : 7915   Mean   :2425     Mean   :361.9
##  3rd Qu.:6.822e+09   3rd Qu.:10096   3rd Qu.:3059     3rd Qu.:460.2
##  Max.   :8.792e+09   Max.   :18734   Max.   :3666     Max.   :683.0
##  average_time_in_bed
##  Min.   : 69.0
##  1st Qu.:365.1
##  Median :437.5
##  Mean   :399.7
##  3rd Qu.:489.3
##  Max.   :961.0
```

Note: - The average calories burn for this sample = 2397 while the maximum burned calories = 3539 - The average steps count for this sample = 7880 while the maximum steps count = 19079

Plotting a few explorations

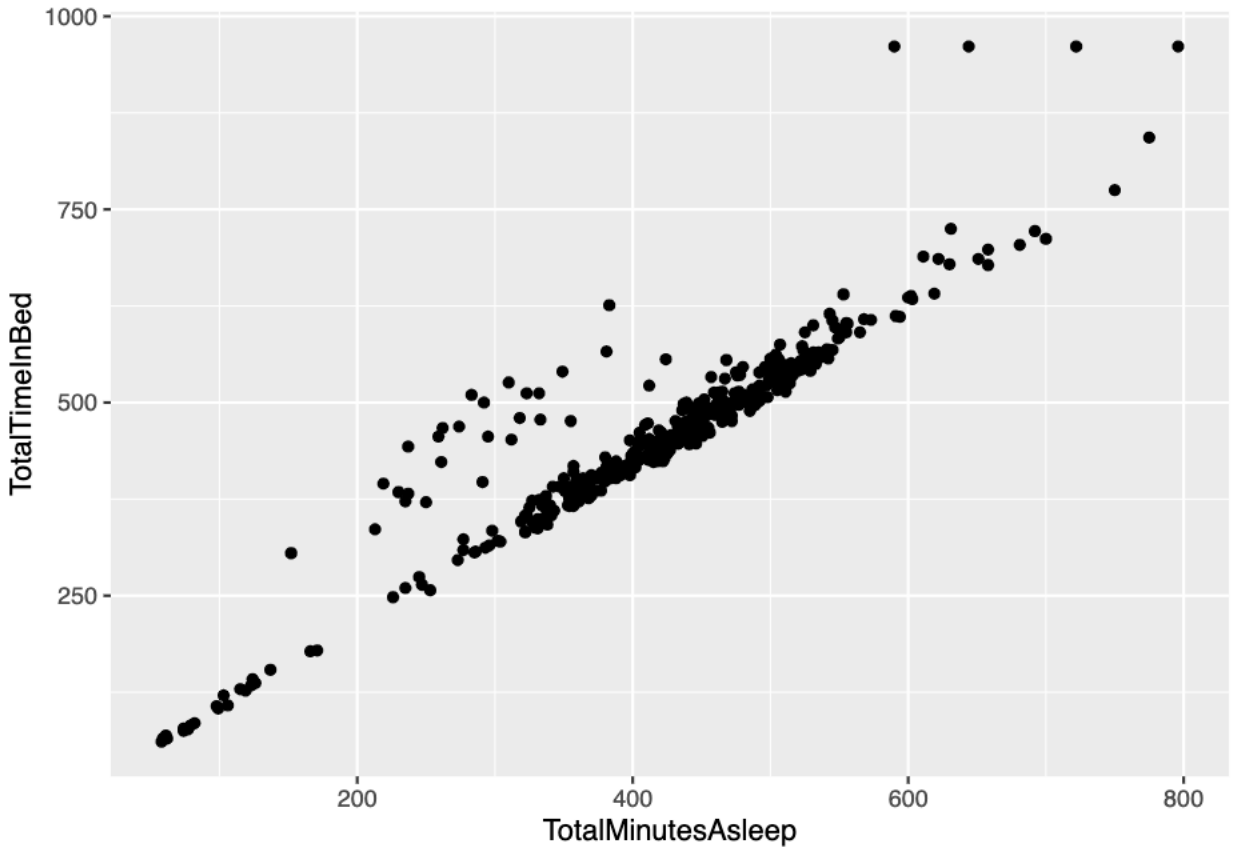
- **Correlations between Total Steps and Sedentary Minutes**

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



- **Correlations between Total time in bed and Total Minutes Asleep**

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

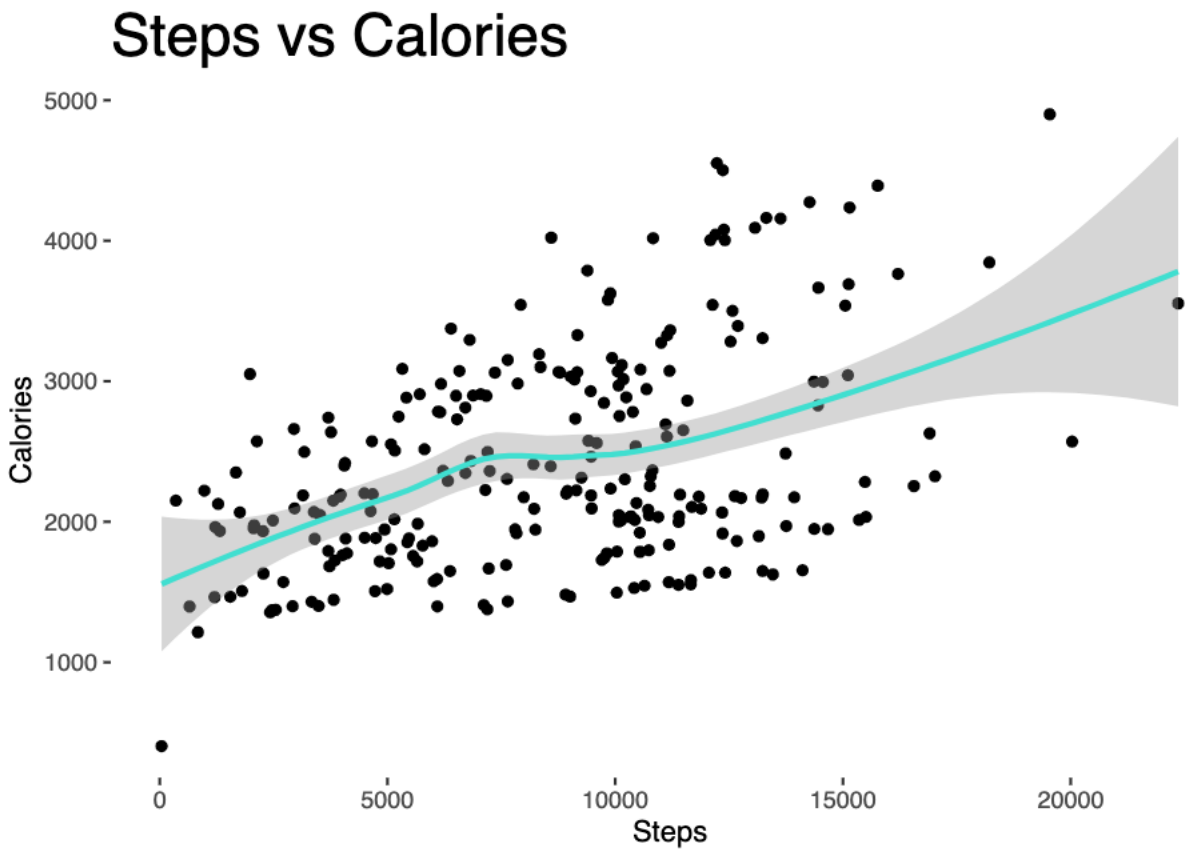


Insight: The correlation between minutes asleep and time in bed is almost linear

- **Correlations between Steps and Calories**

```
ggplot(final_daily, aes(x=TotalSteps, y=Calories))+
  geom_jitter()+
  geom_smooth(color = "turquoise")+
  labs(title = "Steps vs Calories", x = "Steps", y = "Calories")+
  theme(panel.background = element_blank(), plot.title = element_text( size=22))
```

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



Insight: There is a positive correlation between the total number of steps and the burned calories

Activity levels

- Use a guideline on steps and activity levels as the classification levels: Sedentary is less than 5,000 steps per day Low active is 5,000 to 7,499 steps per day Somewhat active is 7,500 to 9,999 steps per day Active is more than 10,000 steps per day Highly active is more than 12,500

```
classification_steps_day <- tibble(
  steps_day = c('<5000', '5000 - 7499', '7500 - 9999', '>10000'),
  activity_level = c('Sedentary', 'Low active', 'Somewhat active', 'Active')
)

print(classification_steps_day)
```

```
## # A tibble: 4 x 2
##   steps_day   activity_level
##   <chr>       <chr>
## 1 <5000       Sedentary
## 2 5000 - 7499 Low active
## 3 7500 - 9999 Somewhat active
## 4 >10000      Active
```

- Assign this classification to the data

```
daily_average_levels <- daily_average %>%
  mutate(activity_level = case_when(
    average_steps < 5000 ~ 'Sedentary',
    average_steps >= 5000 & average_steps < 7500 ~ 'Low active',
```

```
    average_steps >= 7500 & average_steps < 10000 ~ 'Somewhat active',
    average_steps >= 10000 ~ 'Active'
    ))
```

- Calculate the percentage of users for each activity level

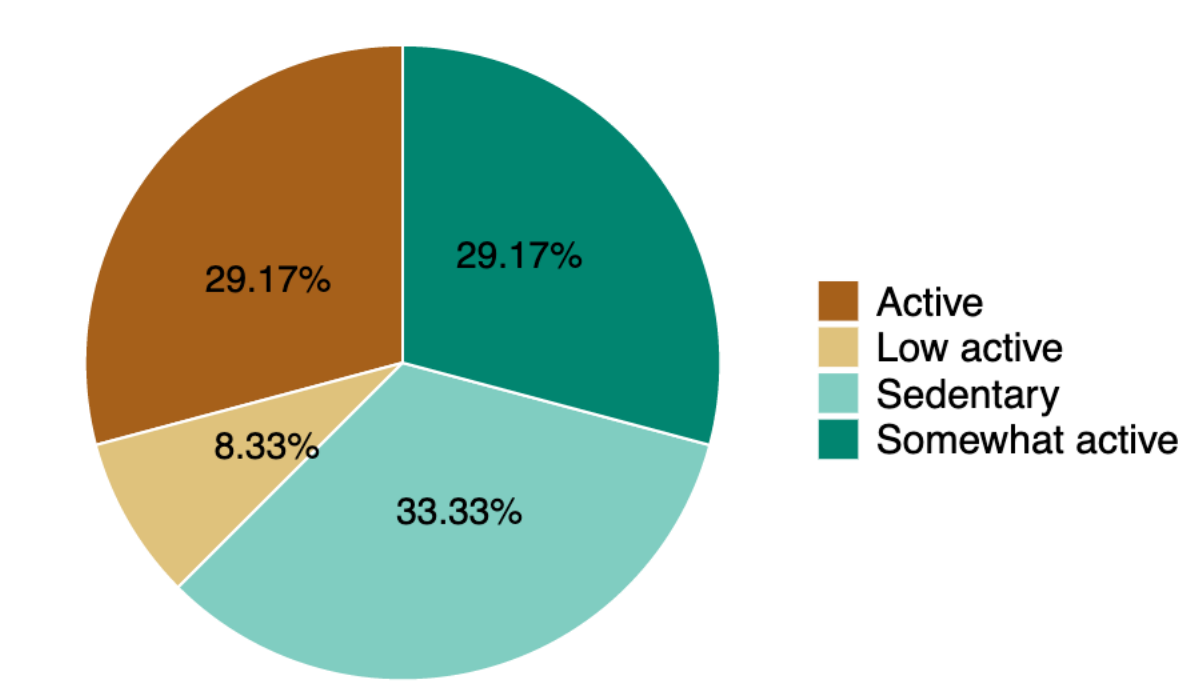
```
activity_level_percentage <- daily_average_levels %>%
  group_by(activity_level) %>%
  summarise(total_level=n()) %>%
  mutate(percentage = (total_level /sum(total_level))) %>%
  mutate(percentage = formattable::percent(percentage)) %>%
  arrange((activity_level))
```

```
glimpse(activity_level_percentage)
```

```
## Rows: 4
## Columns: 3
## $ activity_level <chr> "Active", "Low active", "Sedentary", "Somewhat active"
## $ total_level    <int> 7, 2, 8, 7
## $ percentage     <formttbl> 29.17%, 8.33%, 33.33%, 29.17%
```

- Create a visualization for activity levels

```
ggplot(activity_level_percentage, aes(x="", y=percentage, fill=activity_level))+
  geom_bar(width=1, stat="identity", color="white")+
  coord_polar("y", start = 0)+
  geom_text(aes(label=percentage), position = position_stack(vjust = 0.45), size = 5)+
  labs(title = "Activity level distribution")+
  scale_fill_brewer(palette ="BrBG")+
  guides(fill = guide_legend(title=NULL))+
  theme_void()+
theme(plot.title = element_text(size=22), legend.text = element_text(size=15))
```

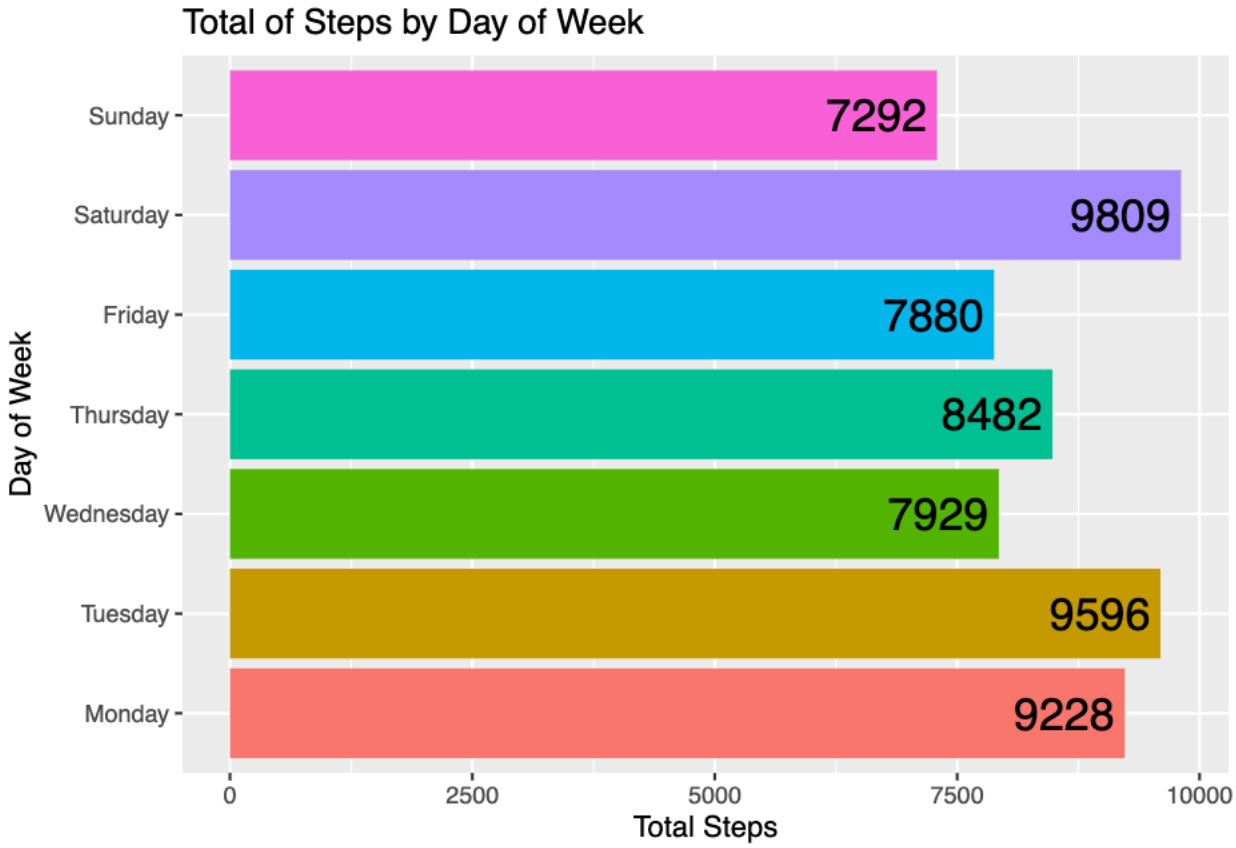


Insight: The biggest part of the users (37.50%) is somewhat active, with an average between 7500 and 9999 steps per day, meanwhile there are as many active users as low active and sedentary.

- **Calculate average total steps by day of week**

```
final_daily$weekday <- factor(final_daily$weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursd
final_daily %>% group_by(weekday) %>% summarise(Mean_total_steps= round(mean(TotalSteps, na.rm = TRUE),
  ggplot(aes(weekday, Mean_total_steps , fill= weekday ))+
  geom_bar(stat="identity", position=position_dodge())+
  coord_flip() +
  geom_text(aes(label= Mean_total_steps ), hjust=1.1, vjust=.5, color="black",position = position_dodge
  #scale_fill_viridis_d() +
  labs(title = "Total of Steps by Day of Week", x="Day of Week", y="Total Steps") +
  theme(plot.subtitle = element_text(color = "black" , face = "italic"), legend.position = "non" )
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



Insight: Saturday has the most active day by users.

- **Calculate daily active time by hour**
- Merge hourly_steps and hourly_calories

```
n_distinct(hourly_steps)
```

```
## [1] 22099
```

```
n_distinct(hourly_calories)
```

```
## [1] 22099
```

```
combined_hourly <- merge(hourly_calories, hourly_steps)
```

```
glimpse(combined_hourly)
```

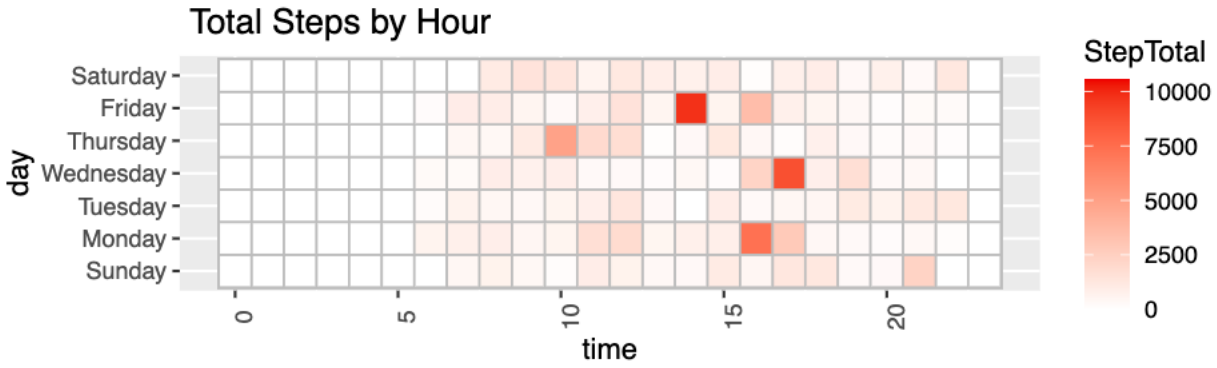
```
## Rows: 22,099
## Columns: 7
## $ Id           <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityHour <chr> "4/12/2016 1:00:00 AM", "4/12/2016 1:00:00 PM", "4/12/201~
## $ date         <date> 2016-04-12, 2016-04-12, 2016-04-12, 2016-04-12, 2016-04-~
## $ time         <int> 1, 13, 10, 22, 11, 23, 0, 12, 2, 14, 3, 15, 4, 16, 5, 17,~
## $ day          <chr> "Tuesday", "Tuesday", "Tuesday", "Tuesday", "Tuesday", "T~
## $ Calories     <int> 61, 66, 99, 65, 76, 81, 81, 73, 59, 110, 47, 151, 48, 76,~
## $ StepTotal    <int> 160, 221, 676, 89, 360, 338, 373, 253, 151, 1166, 0, 2063~
```

- Visualization

```
combined_hourly$day<- factor(combined_hourly$day, levels = c("Sunday", "Monday", "Tuesday", "Wednesday"
n_distinct(combined_hourly)
```

```
## [1] 22099
```

```
ggplot(combined_hourly, aes(time, day, fill= StepTotal)) +  geom_tile(color= "grey", lwd=.4 , linetype
```



Insight: Most of participants activity hours during the week days between 9:00AM and 4:00PM

Recommendations

Marketing Campaigns are recommended to be conducted on Saturday and during the daytime to attract more active users.

Most of the users belong to Somewhat active, with average steps between 7500 and 9999. Thus, Bellabeat smart devices with notification functions of total steps reminder and provide tips about how to gain more steps might encourage users to exceed 10000 steps per day.

Other functions such as bed-time reminders and total daily calories calculating would be necessary for users.

For further analysis, I would recommend Bellabeat store a bigger sample of data and include characteristics such as age, demographics, preferences, and lifestyle. The data could be obtained from periodic surveys done through the Bellabeat app.

–

Thank you!