Bellabeat Case Study with R

Quang Thien

2023-07-05

1.Summary

Bellabeat is a high-tech company that manufactures health-focused smart products.They offer different smart devices that collect data on activity, sleep, stress, and reproductive health to empower women with knowledge about their own health and habits.

The main focus of this case is to analyze smart devices fitness data and determine how it could help unlock new growth opportunities for Bellabeat. We will focus on one of Bellabeat's products: Bellabeat app.

The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products.

2.Ask phase

2.1 About company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women. By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website. The company has invested in traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates. Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy.

2.2 Business Task

Sršen asks you to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. She then wants you to select one Bellabeat product to apply these insights to in your presentation. These questions will guide your analysis:

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

3.Prepare phase

3.1 Data used

The data source used for our case study is FitBit Fitness Tracker Data. This dataset is stored in Kaggle and was made available through Mobius. Data

3.2 Accessibility and privacy of data

Verifying the metadata of our dataset we can confirm it is open-source. The owner has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law. You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

3.3 Information about our dataset

These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences.

3.4 Data Organization and verification

Available to us are 18 CSV documents. Each document represents different quantitative data tracked by Fitbit. The data is considered long since each row is one time point per subject, so each subject will have data in multiple rows.Every user has a unique ID and different rows since data is tracked by day and time. Due to the small size of sample I sorted and filtered tables creating Pivot Tables in Excel. I was able to verify attributes and observations of each table and relations between tables. Counted sample size (users) of each table and verified time length of analysis - 31 days.

3.5 Data Credibility and Integrity

Due to the limitation of size (30 users) and not having any demographic information we could encounter a sampling bias. We are not sure if the sample is representative of the population as a whole. Another problem we would encounter is that the dataset is not current and also the time limitation of the survey (2 months long). That is why we will give our case study an operational approach.

4.Process phase

I will focus my analysis in R due to the accessibility, amount of data and to be able to create data visualization to share my results with stakeholders.

4.1 Loading packages

```
library(ggpubr)
library(tidyverse)
library(here)
library(skimr)
library(janitor)
library(lubridate)
library(ggrepel)
```

4.2 Importing database

```
activity <- read_csv("~/Case study/Data source/dailyActivity_merged.csv")
sleep <- read_csv("~/Case study/Data source/sleepDay_merged.csv")
weight <- read_csv("~/Case study/Data source/weightLoginfo_merged.csv")
hourly_intensities <- read_csv("~/Case study/Data source/hourlyIntensities_merged.csv")
hourly_calories <- read_csv("~/Case study/Data source/hourlyCalories_merged.csv")
hourly_steps <- read_csv("~/Case study/Data source/hourlySteps_merged.csv")
hear_rate<-read_csv("~/Case study/Data source/heartrate_seconds_merged.csv")
```

4.3 Cleaning and formating

4.3.1 N/A

I use is.na to check whether there are any missing values.

```
colSums(is.na(activity))
```

```
##                       Id             ActivityDate                TotalSteps
##                        0                        0                         0
##            TotalDistance          TrackerDistance LoggedActivitiesDistance
##                        0                        0                         0
##       VeryActiveDistance ModeratelyActiveDistance      LightActiveDistance
##                        0                        0                         0
##  SedentaryActiveDistance        VeryActiveMinutes      FairlyActiveMinutes
##                        0                        0                         0
##     LightlyActiveMinutes         SedentaryMinutes                  Calories
##                        0                        0                         0
```

```
colSums(is.na(sleep))
```

```
##                  Id           SleepDay  TotalSleepRecords TotalMinutesAslee
p
##                   0                  0                  0
0
##     TotalTimeInBed
##                   0
```

```
colSums(is.na(weight))
```

```
##             Id           Date       WeightKg   WeightPounds            Fat
##              0              0              0              0             65
##            BMI IsManualReport          LogId
##              0              0              0
```

```
colSums(is.na(hourly_calories))
```

```
##           Id ActivityHour     Calories
##            0            0            0
```

```
colSums(is.na(hourly_intensities))
```

```
##                Id     ActivityHour   TotalIntensity AverageIntensity
##                 0                0                0                0
```

```
colSums(is.na(hourly_steps))
```

```
##           Id ActivityHour    StepTotal
##            0            0            0
```

I already check the data. The Fat column is almost missing so i decided to drop this column

```
weight <- select(weight,-Fat)
```

4.3.2 Check duplicate

```
sum(duplicated(activity))
## [1] 0
sum(duplicated(sleep))
## [1] 3
sum(duplicated(weight))
## [1] 0
sum(duplicated(hourly_calories))
## [1] 0
sum(duplicated(hourly_intensities))
## [1] 0
sum(duplicated(hourly_steps))
## [1] 0
```

4.3.3 Exploring data

```
n_distinct(activity$Id)
## [1] 33
n_distinct(sleep$Id)
## [1] 24
n_distinct(weight$Id)
## [1] 8
n_distinct(hourly_calories$Id)
## [1] 33
n_distinct(hourly_intensities$Id)
## [1] 33
n_distinct(hourly_steps$Id)
## [1] 33
n_distinct(hear_rate$Id)
## [1] 14
```

This information tells us about number participants in each data sets. There is 33 participants in the activity, calories and intensities data sets, 24 in the sleep and only 8 in the weight data set. 8 participants is not significant to make any recommendations and conclusions based on this data.

4.3.4 Remove N/A and duplicate

```
sleep <- sleep %>%
  drop_na() %>%
  distinct()
```

4.3.5 Clean and rename column

```
activity<- clean_names(activity)
sleep<- clean_names(sleep)
hourly_calories<- clean_names(hourly_calories)
hourly_intensities<- clean_names(hourly_intensities)
hourly_steps<- clean_names(hourly_steps)
```

4.3.6 Fixing formating of datasets

```
## hourly_intensities
hourly_intensities$ActivityHour=as.POSIXct(hourly_intensities$activity_hour,
format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
hourly_intensities$time <- format(hourly_intensities$ActivityHour, format = "
%H:%M:%S")
hourly_intensities$date <- format(hourly_intensities$ActivityHour, format = "
%d/%m/%y")
## hourly_calories
hourly_calories$ActivityHour=as.POSIXct(hourly_calories$activity_hour, format
="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
hourly_calories$time <- format(hourly_calories$ActivityHour, format = "%H:%M:
%S")
hourly_calories$date <- format(hourly_calories$ActivityHour, format = "%d/%m/
%y")
## hourly_steps
hourly_steps$activity_hour=as.POSIXct(hourly_steps$activity_hour, format="%m/
%d/%Y %I:%M:%S %p", tz=Sys.timezone())
## activity
```

```
activity$activity_date=as.POSIXct(activity$activity_date, format="%m/%d/%Y",
tz=Sys.timezone())

activity$date <- format(activity$activity_date, format = "%d/%m/%y")

## sleep

sleep$sleep_day=as.POSIXct(sleep$sleep_day, format="%m/%d/%Y %I:%M:%S %p", tz
=Sys.timezone())

sleep$date <- format(sleep$sleep_day, format = "%d/%m/%y")
```

Change all date format into dd/mm/yy. Popular format in VietNam

4.3.7 Summary data

```
## activity
activity %>%
  select(total_steps,
         total_distance,
         calories) %>%
  summary()
```

```
##    total_steps    total_distance      calories
##  Min.   :    0   Min.   : 0.000   Min.   :   0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:1828
##  Median : 7406   Median : 5.245   Median :2134
##  Mean   : 7638   Mean   : 5.490   Mean   :2304
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:2793
##  Max.   :36019   Max.   :28.030   Max.   :4900
```

```
activity %>%
  select(total_steps,
         total_distance,
         calories) %>%
  summarise(cv_total_steps = sd(total_steps)/mean(total_steps),
            cv_total_distance = sd(total_distance)/mean(total_distance),
            cv_calories = sd(calories)/mean(calories))
```

```
## # A tibble: 1 × 3
##   cv_total_steps cv_total_distance cv_calories
##            <dbl>             <dbl>       <dbl>
## 1          0.666             0.715       0.312
```

```
## explore num of active minutes per category
activity %>%
  select(very_active_minutes, fairly_active_minutes, lightly_active_minutes, sedentary_minutes) %>%
  summary()
```

```
##  very_active_minutes fairly_active_minutes lightly_active_minutes
##  Min.   :  0.00      Min.   :  0.00        Min.   :  0.0
##  1st Qu.:  0.00      1st Qu.:  0.00        1st Qu.:127.0
##  Median :  4.00      Median :  6.00        Median :199.0
##  Mean   : 21.16      Mean   : 13.56        Mean   :192.8
##  3rd Qu.: 32.00      3rd Qu.: 19.00        3rd Qu.:264.0
##  Max.   :210.00      Max.   :143.00        Max.   :518.0
##  sedentary_minutes
##  Min.   :   0.0
##  1st Qu.: 729.8
##  Median :1057.5
##  Mean   : 991.2
##  3rd Qu.:1229.5
##  Max.   :1440.0
```

```
activity %>%
  select(very_active_minutes,
         fairly_active_minutes,
         lightly_active_minutes,
         sedentary_minutes) %>%
  summarise(cv_very_active_minutes = sd(very_active_minutes)/mean(very_active_minutes),
            cv_fairly_active_minutes = sd(fairly_active_minutes)/mean(fairly_active_minutes),
            cv_lightly_active_minutes = sd(lightly_active_minutes)/mean(lightly_active_minutes),
            cv_sedentary_minutes = sd(sedentary_minutes)/mean(sedentary_minutes))
```

```
## # A tibble: 1 × 4
##   cv_very_active_minutes cv_fairly_active_minutes cv_lightly_active_minutes
##                    <dbl>                    <dbl>                      <dbl>
```

```
## 1                   1.55                      1.47                      0.56
6

## # ℹ 1 more variable: cv_sedentary_minutes <dbl>
```

```
## intensities dataset is similar to activity dataset

## hourly_intensities

hourly_intensities %>%

  select(total_intensity) %>%

  summary()
```

```
##  total_intensity

##  Min.   :  0.00

##  1st Qu.:  0.00

##  Median :  3.00

##  Mean   : 12.04

##  3rd Qu.: 16.00

##  Max.   :180.00
```

```
sd(hourly_intensities$TotalIntensity)/mean(hourly_intensities$TotalIntensity)
```

```
## Warning: Unknown or uninitialised column: `TotalIntensity`.

## Unknown or uninitialised column: `TotalIntensity`.

## Warning in mean.default(hourly_intensities$TotalIntensity): argument is no
t

## numeric or logical: returning NA

## [1] NA
```

```
## hourly_calories

hourly_calories %>%

  select(calories) %>%

  summary()
```

```
##     calories

##  Min.   : 42.00

##  1st Qu.: 63.00

##  Median : 83.00

##  Mean   : 97.39

##  3rd Qu.:108.00

##  Max.   :948.00
```

```
sd(hourly_calories$Calories)/mean(hourly_calories$Calories)
```

```
## Warning: Unknown or uninitialised column: `Calories`.
```

```
## Warning: Unknown or uninitialised column: `Calories`.
## Warning in mean.default(hourly_calories$Calories): argument is not numeric
or
## logical: returning NA
## [1] NA
```

```
## hourly_steps
hourly_steps %>%
  select(step_total) %>%
  summary()
```

```
##    step_total
##  Min.   :    0.0
##  1st Qu.:    0.0
##  Median :   40.0
##  Mean   :  320.2
##  3rd Qu.:  357.0
##  Max.   :10554.0
```

```
sd(hourly_steps$step_total)/mean(hourly_steps$step_total)
```

```
## [1] 2.15633
```

```
## sleep
sleep %>%
  select(total_sleep_records, total_minutes_asleep, total_time_in_bed) %>%
  summary()
```

```
##  total_sleep_records total_minutes_asleep total_time_in_bed
##  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
```

```
sleep %>%
  select(total_sleep_records, total_minutes_asleep, total_time_in_bed) %>%
  summarise(cv_total_sleep_records=sd(total_sleep_records)/mean(total_sleep_r
ecords),
            cv_total_minutes_asleep=sd(total_minutes_asleep)/mean(total_minut
es_asleep),
```

```
            cv_TotalTimeInBed=sd(total_time_in_bed)/mean(total_time_in_bed))
## # A tibble: 1 × 3
##   cv_total_sleep_records cv_total_minutes_asleep cv_TotalTimeInBed
##                    <dbl>                   <dbl>             <dbl>
## 1                  0.310                   0.283             0.278
```

4.4 Merging data

```
# Join df activity and sleep
sleep_activity_merged <- merge(activity, sleep, by = c("id", "date"))
#Change data type of id column
sleep_activity_merged$id <- as.character(sleep_activity_merged$id)
head(sleep_activity_merged)
```

```
##           id     date activity_date total_steps total_distance tracker_dis
tance
## 1 1503960366 01/05/16    2016-05-01       10602           6.81
6.81
## 2 1503960366 02/05/16    2016-05-02       14727           9.71
9.71
## 3 1503960366 03/05/16    2016-05-03       15103           9.66
9.66
## 4 1503960366 05/05/16    2016-05-05       14070           8.90
8.90
## 5 1503960366 06/05/16    2016-05-06       12159           8.03
8.03
## 6 1503960366 07/05/16    2016-05-07       11992           7.71
7.71
##   logged_activities_distance very_active_distance moderately_active_distan
ce
## 1                          0                 2.29                       1.
60
## 2                          0                 3.21                       0.
57
## 3                          0                 3.73                       1.
05
## 4                          0                 2.92                       1.
08
## 5                          0                 1.97                       0.
25
```

```
## 6                          0                2.46                       2.
12

##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  2.92                         0                  33
## 2                  5.92                         0                  41
## 3                  4.88                         0                  50
## 4                  4.88                         0                  45
## 5                  5.81                         0                  24
## 6                  3.13                         0                  37
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    35                    246               730     1820
## 2                    15                    277               798     2004
## 3                    24                    254               816     1990
## 4                    24                    250               857     1959
## 5                     6                    289               754     1896
## 6                    46                    175               833     1821
##    sleep_day total_sleep_records total_minutes_asleep total_time_in_bed
## 1 2016-05-01                   1                  369               396
## 2 2016-05-02                   1                  277               309
## 3 2016-05-03                   1                  273               296
## 4 2016-05-05                   1                  247               264
## 5 2016-05-06                   1                  334               367
## 6 2016-05-07                   1                  331               349
```

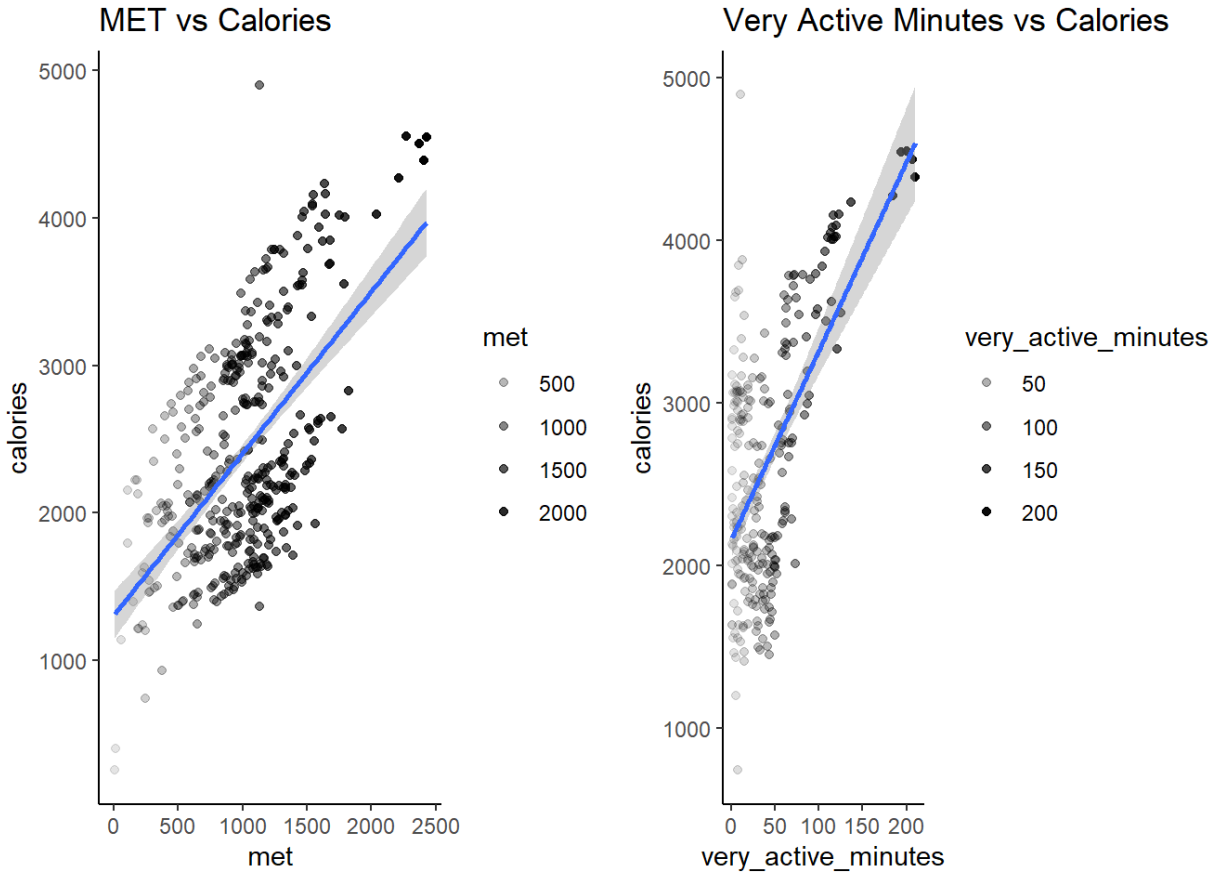
5 Analzye and share phase

5.1 Calculate MET

```
##Create Column MET

sleep_activity_merged <- sleep_activity_merged %>%

  mutate(met = 3.3 * lightly_active_minutes + 4 * fairly_active_minutes + 8 *
very_active_minutes)

#Correlation between activities and calories

ggarrange(
```

```
  ggplot(data=sleep_activity_merged, aes(y=calories, x=met)) +

    geom_point(aes(alpha=met)) + geom_smooth(method = lm) + theme_classic() +
labs(title = "MET vs Calories"),

  ggplot(data=subset(sleep_activity_merged, very_active_minutes != 0), aes(x=
very_active_minutes, y=calories)) +

    geom_point(aes(alpha=very_active_minutes)) + geom_smooth(method = lm) + t
heme_classic() + labs(title = "Very Active Minutes vs Calories")

)
```

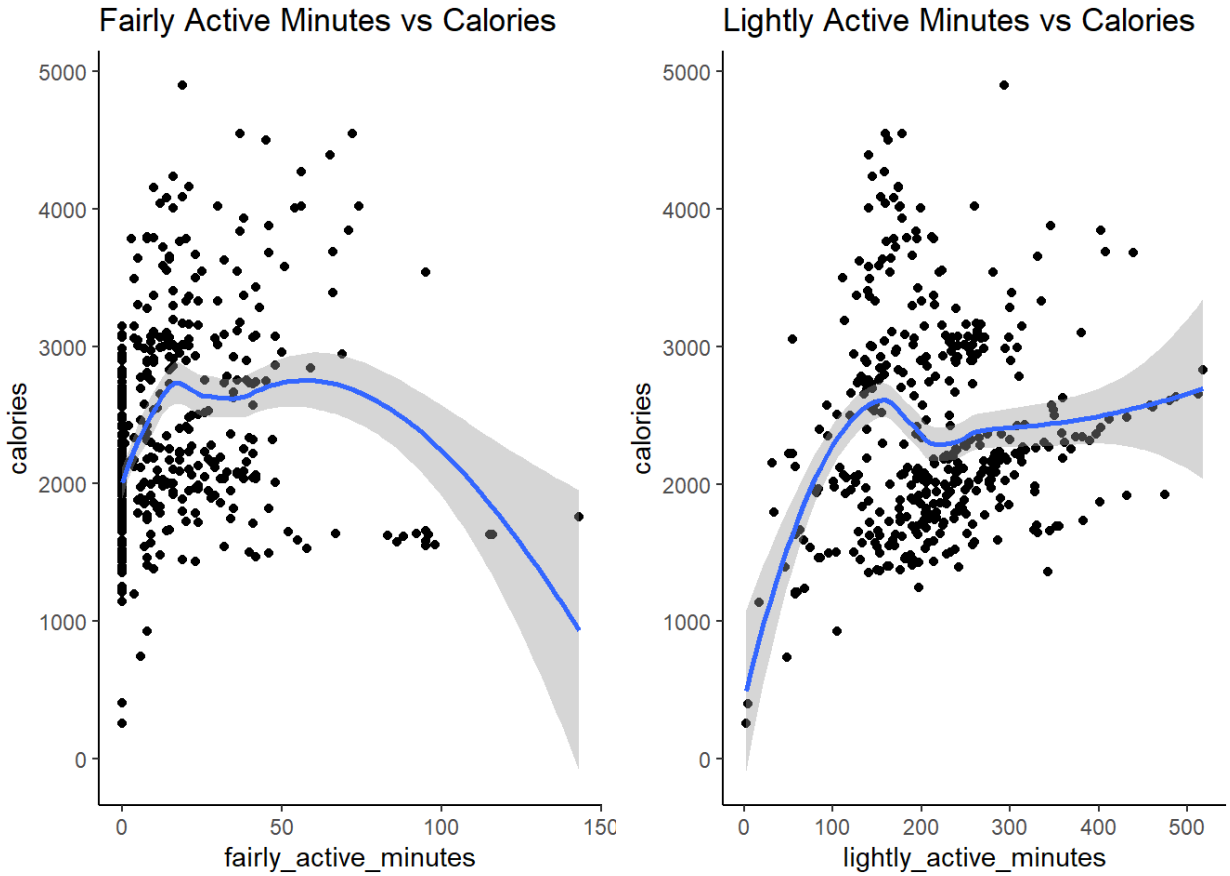
```
## `geom_smooth()` using formula = 'y ~ x'

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




```
##Positive relation

ggarrange(

  ggplot(data=sleep_activity_merged, aes(x=fairly_active_minutes, y=calories)
) +

    geom_point() + geom_smooth() + theme_classic() + labs(title = "Fairly Act
ive Minutes vs Calories"),
```

```
  ggplot(data=sleep_activity_merged, aes(x=lightly_active_minutes, y=calories
)) +

    geom_point() + geom_smooth()+ theme_classic() + labs(title = "Lightly Act
ive Minutes vs Calories")

)
```

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

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




```
## Weak relation
```

MET and Very Active Minutes have a positive relation with Calories Fairly Active Minutes and Lightly Active Minutes may have no relation with Calories –> It mean to burn more calories user should focus on vigorous minutes.

5.2 Sleep Quality

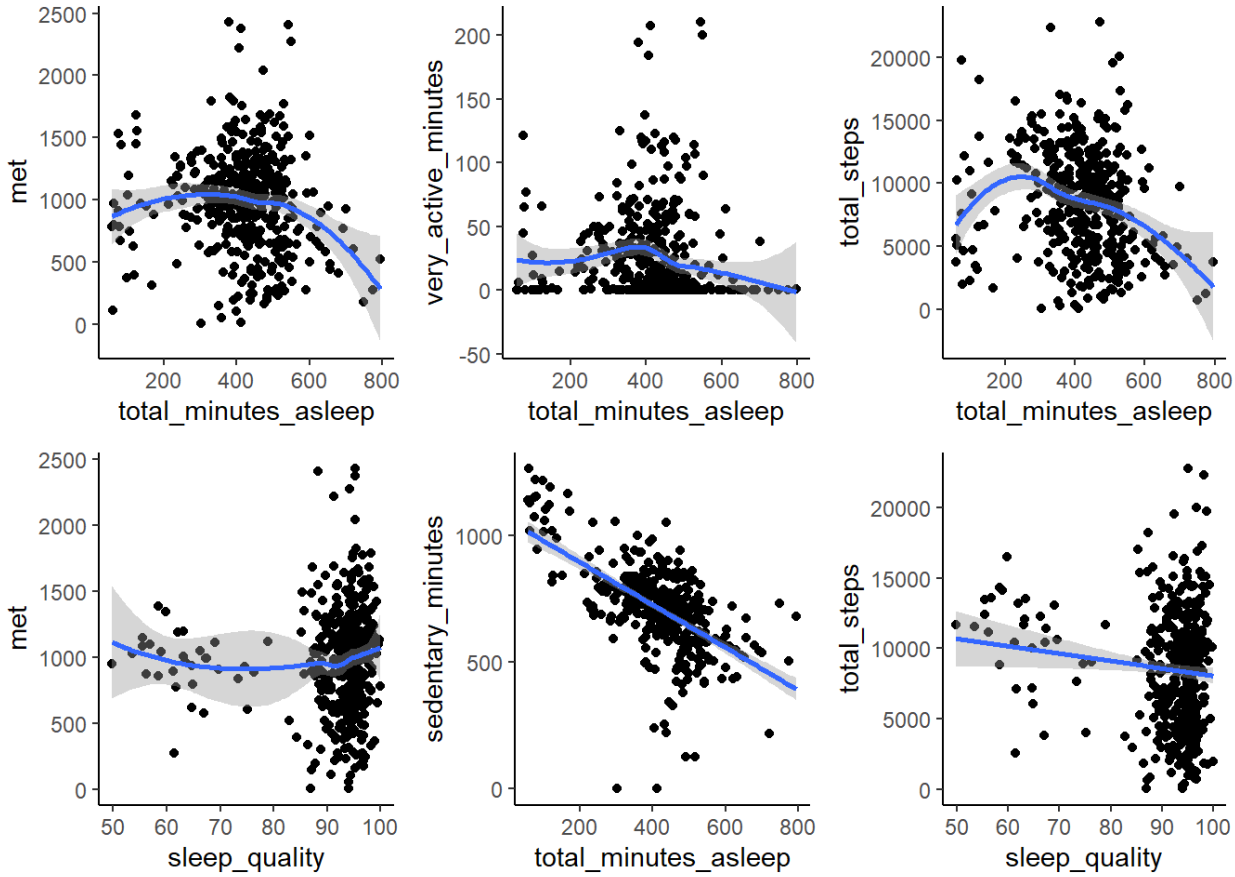
Sleep quality = time asleep/time in bed:

- Sleep quality >=85% is Very good
- Sleep quality <85% and >=75% is Good

- Sleep quality <75% and >=65% is Fair
- Sleep quality <65% is Poor

```
sleep_activity_merged <- sleep_activity_merged %>%
  mutate(sleep_quality = total_minutes_asleep / total_time_in_bed * 100)
##Create Column Time to fall asleep
sleep_activity_merged <- sleep_activity_merged %>%
  mutate(time_fasl = total_time_in_bed - total_minutes_asleep)
ggarrange(
ggplot(data=sleep_activity_merged, aes(x=total_minutes_asleep, y=met)) +
  geom_point() + geom_smooth() + theme_classic(),
ggplot(data=sleep_activity_merged, aes(x=total_minutes_asleep, y=very_active_
minutes)) +
  geom_point() + geom_smooth() + theme_classic(),
ggplot(data=sleep_activity_merged, aes(x=total_minutes_asleep, y=total_steps)
) +
  geom_point() + geom_smooth()+theme_classic() ,
ggplot(data=sleep_activity_merged, aes(x=sleep_quality, y=met)) +
  geom_point() + geom_smooth()+theme_classic() ,
ggplot(data=sleep_activity_merged, aes(y=sedentary_minutes, x=total_minutes_a
sleep)) +
  geom_point() + geom_smooth(method = lm) + theme_classic(),
ggplot(data=sleep_activity_merged, aes(y=total_steps, x=sleep_quality)) +
  geom_point() + geom_smooth(method = lm) + theme_classic()
)
```

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



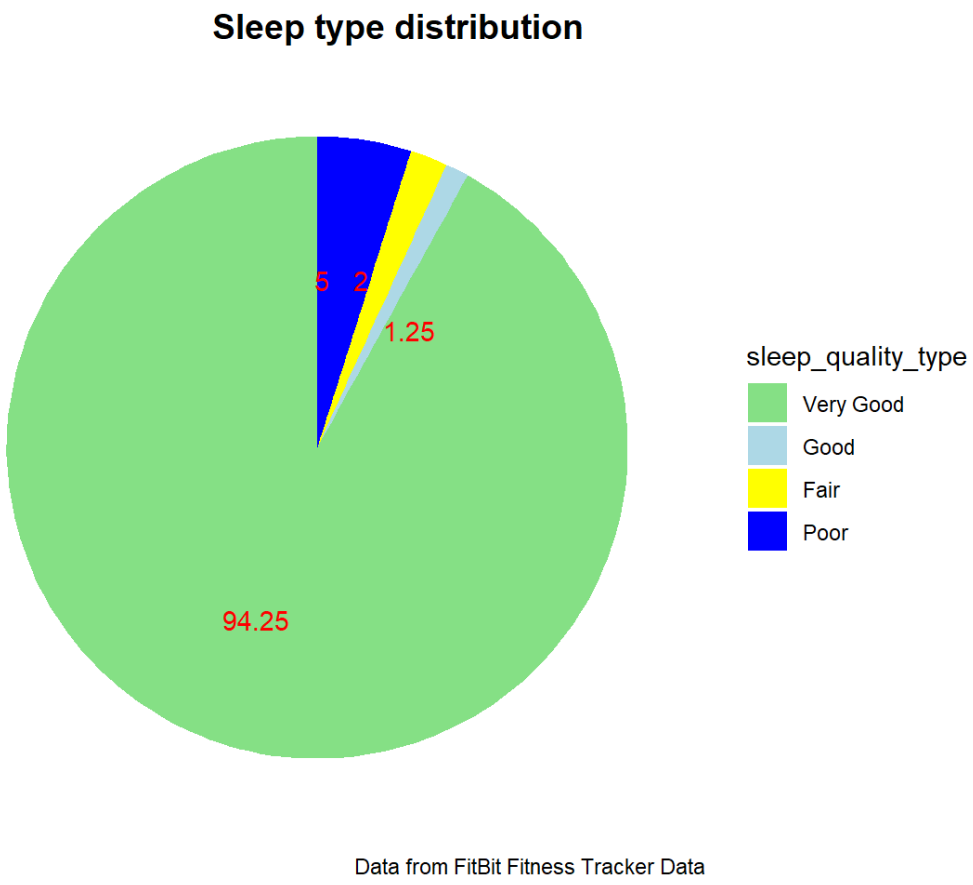
Physical Activities have no significant influence on sleep quality.However people with sedentary lifestyle tend to have fewer sleep time. Type of sleep quality:

```
sleep_activity_merged <- sleep_activity_merged %>%
  mutate(sleep_quality_type= case_when(
    sleep_quality >=85 ~ "Very Good",
    sleep_quality >=75 & sleep_quality <85 ~ "Good",
    sleep_quality >=65 & sleep_quality <75 ~ "Fair",
    sleep_quality <65 ~ "Poor"
  ))
sleep_quality_percent<- sleep_activity_merged %>%
  group_by(sleep_quality_type) %>%
  summarise(total=n())%>%
  mutate(total_percent=total/sum(n()))
sleep_quality_percent$sleep_quality_type<-factor(sleep_quality_percent$sleep_
quality_type,levels = c("Very Good", "Good", "Fair", "Poor"))
head(sleep_quality_percent)
```

```
## # A tibble: 4 × 3
##   sleep_quality_type total total_percent
##   <fct>              <int>         <dbl>
## 1 Fair                   8          2
## 2 Good                   5          1.25
## 3 Poor                  20          5
## 4 Very Good            377         94.2
```

I use pie chart to visualize sleep quality percent

```
sleep_quality_percent %>%
  ggplot(aes(x="",y=total_percent, fill=sleep_quality_type)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0) +
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.7, size=14, face = "bold"))+
  scale_fill_manual(values = c("#85e085","lightblue", "yellow", "blue")) +
  geom_text_repel(aes(label = total_percent),
            position = position_stack(vjust = 0.5),color="red")+
  labs(title="Sleep type distribution",caption="Data from FitBit Fitness Trac
ker Data")
```



Almost all users have a very good sleep.

5.3 User type

since we don't have any demographic variables from our sample we want to determine the type of users with the data we have. We can classify the users by activity considering the daily amount of steps. We can categorize users as follows:

- Sedentary - Less than 5000 steps a day.
- Lightly active - Between 5000 and 7499 steps a day.
- Fairly active - Between 7500 and 9999 steps a day.
- Very active - More than 10000 steps a day.

Classification has been made per the following article Click here

First we will calculate the daily steps average by user.

```
daily_average <- sleep_activity_merged %>%
  group_by(id) %>%
  summarise (mean_daily_steps = mean(total_steps), mean_daily_calories = mean
(calories), mean_daily_sleep = mean(total_minutes_asleep))
head(daily_average)
```

```
## # A tibble: 6 × 4
##   id         mean_daily_steps mean_daily_calories mean_daily_sleep
##   <chr>                 <dbl>               <dbl>            <dbl>
## 1 1503960366           12406.               1872.             360.
## 2 1644430081            7968.               2978.             294
## 3 1844505072            3477                1676.             652
## 4 1927972279            1490                2316.             417
## 5 2026352035            5619.               1541.             506.
## 6 2320127002            5079                1804               61
```

We will now classify our users by the daily average steps.

```
daily_average <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7500 ~ "lightly active",
    mean_daily_steps >=7500 & mean_daily_steps < 10000 ~ "fairly active",
    mean_daily_steps >= 10000 ~ "very active"
  ))
head(daily_average)
```

```
## # A tibble: 6 × 5
##   id         mean_daily_steps mean_daily_calories mean_daily_sleep user_ty
pe
##   <chr>                 <dbl>               <dbl>            <dbl> <chr>
## 1 1503960366           12406.               1872.             360. very ac
tive
## 2 1644430081            7968.               2978.             294  fairly
active
## 3 1844505072            3477                1676.             652  sedenta
ry
## 4 1927972279            1490                2316.             417  sedenta
ry
## 5 2026352035            5619.               1541.             506. lightly
acti…
## 6 2320127002            5079                1804               61  lightly
acti…
```

Now that we have a new column with the user type we will create a data frame with the percentage of each user type to better visualize them on a graph.

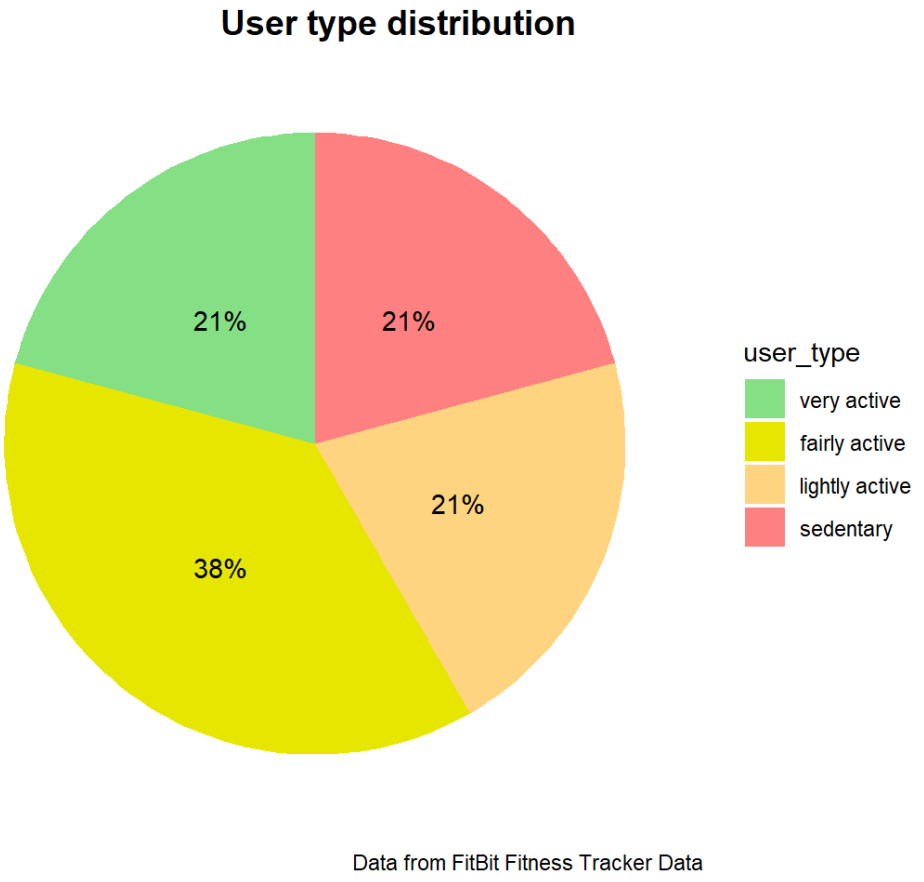
```
user_type_percent <- daily_average %>%
  group_by(user_type) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  mutate(total_percent = total/totals) %>%
  mutate(percent = scales::percent(total_percent))
user_type_percent$user_type <- factor(user_type_percent$user_type ,
levels = c("very active", "fairly active", "lightly active", "sedentary"))
head(user_type_percent)
```

```
## # A tibble: 4 × 5
##   user_type      total totals total_percent percent
##   <fct>          <int>  <int>         <dbl> <chr>
## 1 fairly active      9     24         0.375 38%
## 2 lightly active     5     24         0.208 21%
## 3 sedentary          5     24         0.208 21%
## 4 very active        5     24         0.208 21%
```

Below we can see that users are fairly distributed by their activity considering the daily amount of steps. **We can determine that based on users activity all kind of users wear smart-devices.**

```
user_type_percent %>%
  ggplot(aes(x="",y=total_percent, fill=user_type)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0) +
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.7, size=14, face = "bold"))+
  scale_fill_manual(values = c("#85e085","#e6e600", "#ffd480", "#ff8080")) +
  geom_text(aes(label = percent),
            position = position_stack(vjust = 0.5))+
```

```
  labs(title="User type distribution",caption="Data from FitBit Fitness Track
er Data")
```



5.4 Steps and minutes asleep per week

We want to know now what days of the week are the users more active and also what days of the week users sleep more. We will also verify if the users walk the recommended amount of steps and have the recommended amount of sleep.

Below we are calculating the weekdays based on our column date. We are also calculating the average steps walked and minutes sleeped by weekday.

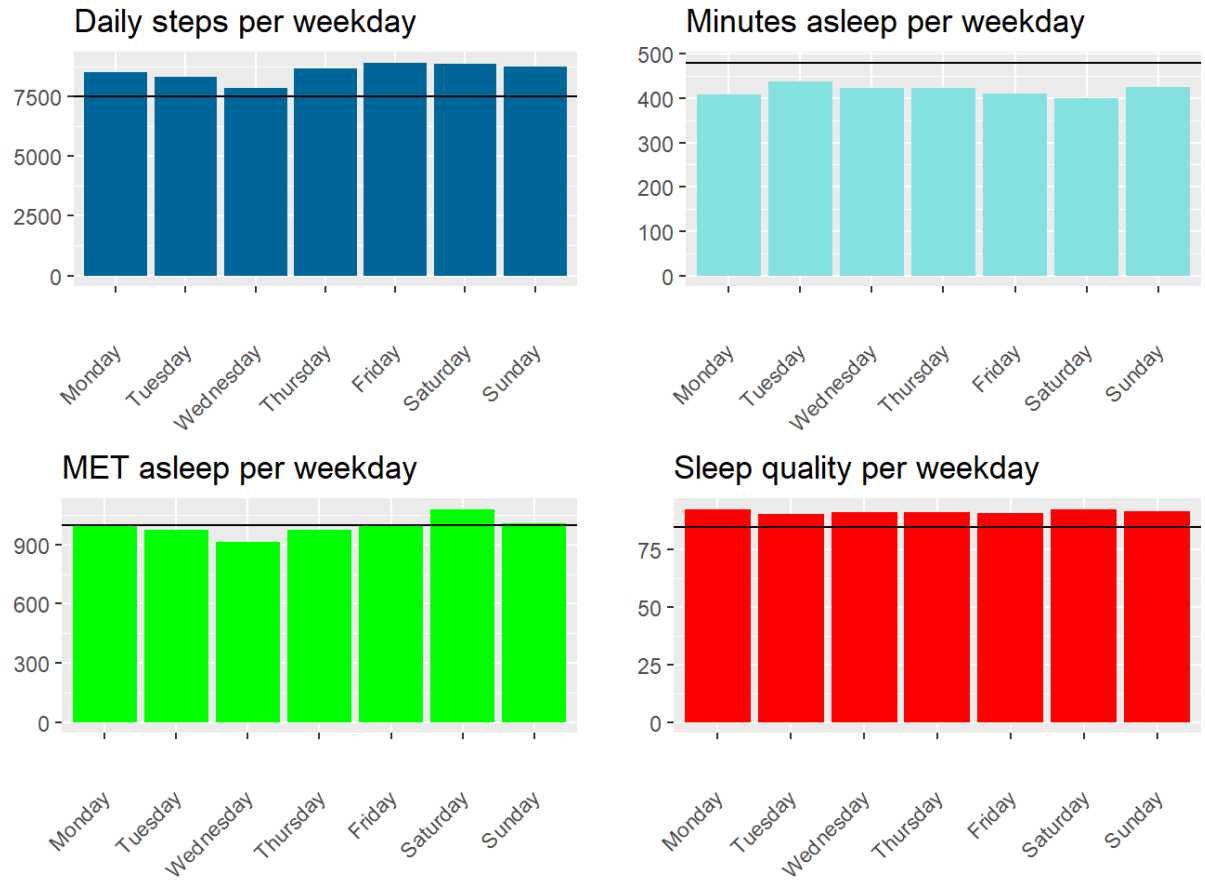
```
sleep_activity_merged<-sleep_activity_merged %>%
  mutate(weekday = weekdays(as.Date(date)))
sleep_activity_merged$weekday <-ordered(sleep_activity_merged$weekday, levels
=c("Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
weekday_steps_sleep <-sleep_activity_merged%>%
  group_by(weekday) %>%
```

```
  summarize (daily_steps = mean(total_steps), daily_sleep = mean(total_minute
s_asleep),
              daily_met = mean(met), daily_sleep_quality = mean(sleep_quality)
)
head(weekday_steps_sleep)
```

```
## # A tibble: 6 × 5
##   weekday   daily_steps daily_sleep daily_met daily_sleep_quality
##   <ord>           <dbl>       <dbl>     <dbl>               <dbl>
## 1 Monday          8515.        408.      997.                92.7
## 2 Tuesday         8296.        437.      976.                90.8
## 3 Wednesday       7837.        422.      912.                91.4
## 4 Thursday        8655.        422.      972.                91.5
## 5 Friday          8900.        411.      997.                91.1
## 6 Saturday        8861.        399.     1079.                92.6
```

```
ggarrange(
  ggplot(weekday_steps_sleep) +
    geom_col(aes(weekday, daily_steps), fill = "#006699") +
    geom_hline(yintercept = 7500) +
    labs(title = "Daily steps per weekday", x= "", y = "") +
    theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1)),
  ggplot(weekday_steps_sleep, aes(weekday, daily_sleep)) +
    geom_col(fill = "#85e0e0") +
    geom_hline(yintercept = 480) +
    labs(title = "Minutes asleep per weekday", x= "", y = "") +
    theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1)),
  ggplot(weekday_steps_sleep, aes(weekday, daily_met))+
    geom_col(fill = "green") +
    geom_hline(yintercept = 1000) +
    labs(title = "MET asleep per weekday", x= "", y = "") +
    theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1)),
  ggplot(weekday_steps_sleep, aes(weekday, daily_sleep_quality))+
    geom_col(fill = "red") +
    geom_hline(yintercept = 85) +
    labs(title = "Sleep quality per weekday", x= "", y = "") +
    theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 1))
```

```
)
```



In the graphs above we can determine the following:

- Users active level are good
- Although users didn't sleep enough 8 hours a day, their quality are high.

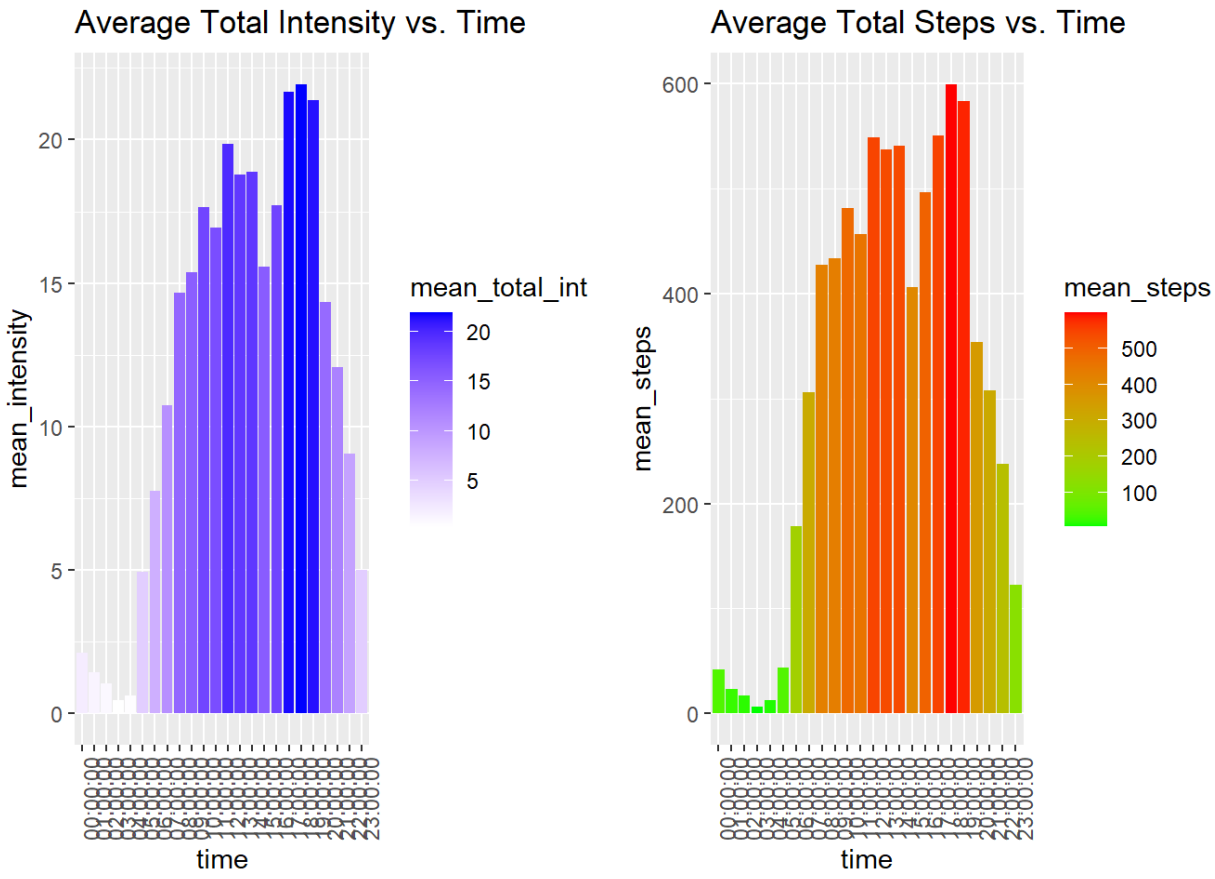
5.5 Mean intensity and steps hourly

Getting deeper into our analysis we want to know when exactly are users more active in a day.

We will use the hourly_steps and hourly_intensities data frame and separate date_time column.

```
int_new <- hourly_intensities %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_int = mean(total_intensity))
step_new <- hourly_steps %>%
  separate( activity_hour, into=c("date","time"), sep=" ") %>%
  group_by(time) %>%
```

```
  summarise(mean_steps= mean(step_total))
head(int_new)
## # A tibble: 6 × 2
##   time     mean_total_int
##   <chr>             <dbl>
## 1 00:00:00          2.13
## 2 01:00:00          1.42
## 3 02:00:00          1.04
## 4 03:00:00          0.444
## 5 04:00:00          0.633
## 6 05:00:00          4.95
head(step_new)
## # A tibble: 6 × 2
##   time     mean_steps
##   <chr>         <dbl>
## 1 00:00:00      42.2
## 2 01:00:00      23.1
## 3 02:00:00      17.1
## 4 03:00:00       6.43
## 5 04:00:00      12.7
## 6 05:00:00      43.9
ggarrange(
ggplot(data=int_new) + geom_col(aes(x=time, y=mean_total_int,fill=mean_total_
int)) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Average Total Intensity vs. Time")+
  scale_fill_gradient(low="white",high="blue")+ labs(y="mean_intensity"),
ggplot(step_new) + geom_col(aes(time, mean_steps,fill=mean_steps)) +
  theme(axis.text.x = element_text(angle = 90) ) + labs(title = "Average Tota
l Steps vs. Time")+
  scale_fill_gradient(low = "green", high = "red")
)
```



We can see that users are more active between 8am and 7pm. Walking more steps during lunch time from 12pm to 2pm and evenings from 5pm and 7pm.

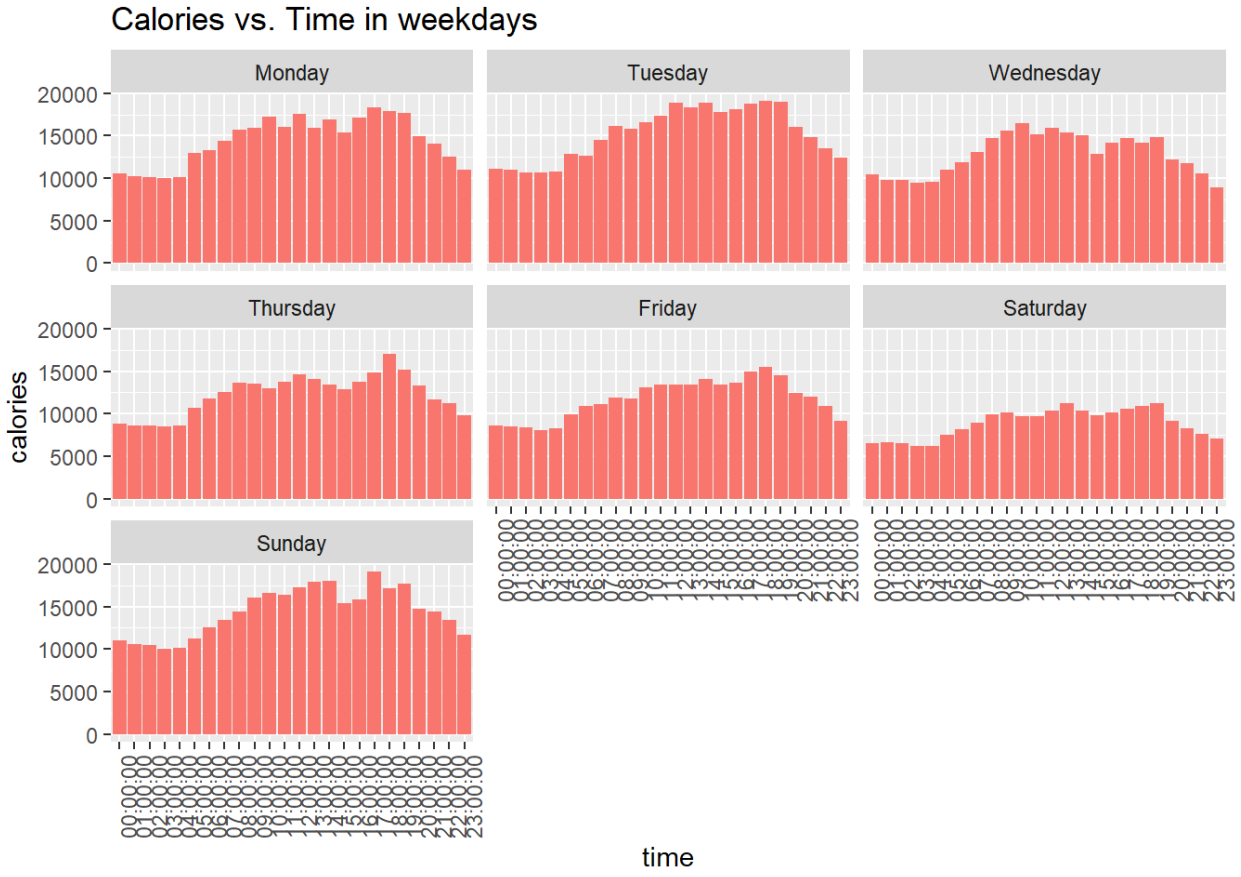
5.6 Hourly calories

```
hourly_calories <- hourly_calories %>%
  mutate(weekday=weekdays(as.Date(date)))
hourly_calories$weekday<-factor(hourly_calories$weekday,levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"))
head(hourly_calories)
```

```
## # A tibble: 6 × 7
##           id activity_hour      calories ActivityHour        time  date  weekday
##        <dbl> <chr>                <dbl> <dttm>              <chr> <chr> <fct>
## 1 1503960366 4/12/2016 12:00:0…       81 2016-04-12 00:00:00 00:0… 12/0… Monday
## 2 1503960366 4/12/2016 1:00:00…       61 2016-04-12 01:00:00 01:0… 12/0… Monday
```

```
## 3 1503960366 4/12/2016 2:00:00…        59 2016-04-12 02:00:00 02:0… 12/0… M
onday
## 4 1503960366 4/12/2016 3:00:00…        47 2016-04-12 03:00:00 03:0… 12/0… M
onday
## 5 1503960366 4/12/2016 4:00:00…        48 2016-04-12 04:00:00 04:0… 12/0… M
onday
## 6 1503960366 4/12/2016 5:00:00…        48 2016-04-12 05:00:00 05:0… 12/0… M
onday
```

```
ggplot(hourly_calories) + geom_col(aes(time, calories,fill="#00B7FF")) +
  theme(axis.text.x = element_text(angle = 90) ) + labs(title = "Calories vs.
Time in weekdays")+
  facet_wrap(~weekday)+theme(legend.position = "none")
```



Users are the most active on Monday, Tuesday and Sunday between 9am to 7pm.

5.7 Use of smart devices.

5.7.1 Number of day using smart devices

Now that we have seen some trends in activity, sleep and calories burned, we want to see how often do the users in our sample use their device. That way we can plan our marketing strategy and see what features would benefit the use of smart devices.

We will calculate the number of users that use their smart device on a daily basis, classifying our sample into three categories knowing that the date interval is 31 days:

high use - users who use their device between 21 and 31 days. moderate use - users who use their device between 10 and 20 days. low use - users who use their device between 1 and 10 days. First we will create a new data frame grouping by Id, calculating number of days used and creating a new column with the classification explained above.

```
day_use <- sleep_activity_merged %>%
  group_by(id) %>%
  summarise(days_used=sum(n())) %>%
  mutate(usage=case_when(
    days_used >= 1 & days_used <=10 ~ "low use",
    days_used >=11 & days_used <=20 ~ "moderate use",
    days_used >=21 & days_used <=31 ~ "high use"
   ))
head(day_use)
```

```
## # A tibble: 6 × 3
##    id         days_used usage
##    <chr>          <int> <chr>
## 1 1503960366         25 high use
## 2 1644430081          4 low use
## 3 1844505072          3 low use
## 4 1927972279          5 low use
## 5 2026352035         28 high use
## 6 2320127002          1 low use
```

We will now create a percentage data frame to better visualize the results in the graph. We are also ordering our usage levels.

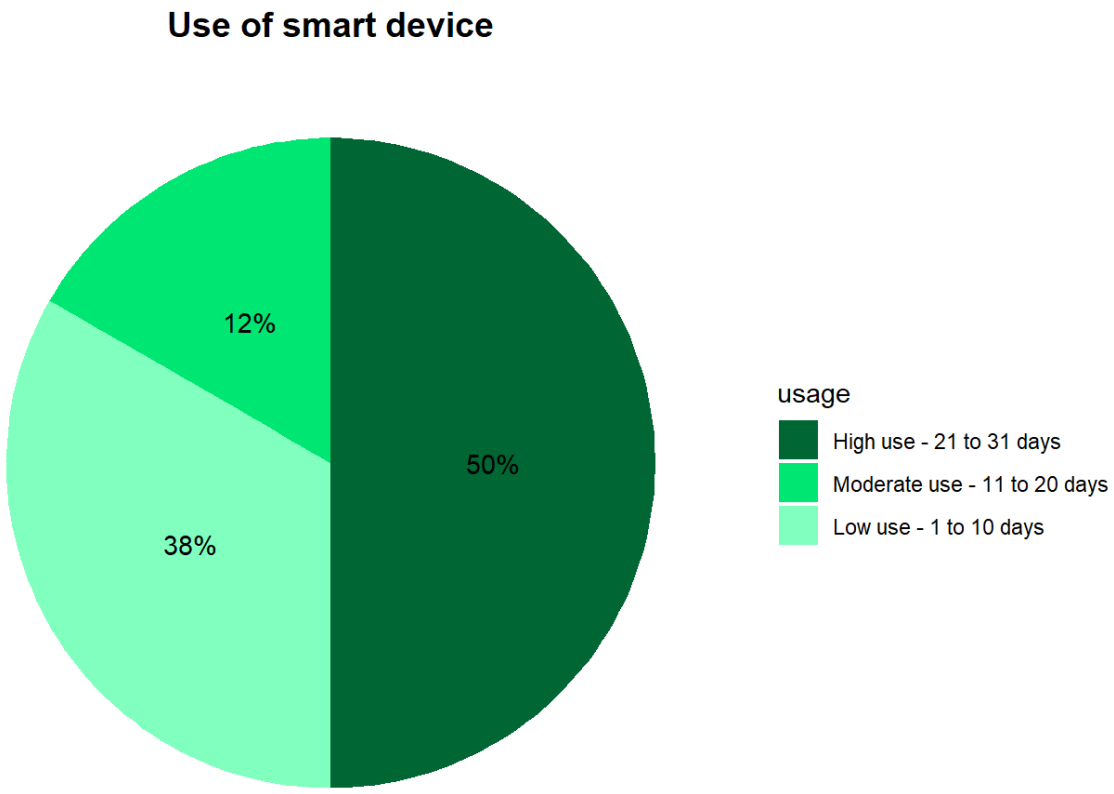
```
usage_percent <- day_use %>%
  group_by(usage) %>%
  summarise(total=n())%>%
  mutate(total_percent=total/sum(total)) %>%
  mutate(use_type_percent=scales::percent(total_percent))
usage_percent$usage <- factor(usage_percent$usage, levels = c("high use","mod
erate use", "low use"))
```

```
head(usage_percent)
```

```
## # A tibble: 3 × 4
##   usage         total total_percent use_type_percent
##   <fct>         <int>         <dbl> <chr>
## 1 high use         12         0.5   50%
## 2 low use           9         0.375 38%
## 3 moderate use      3         0.125 12%
```

Now that we have our new table we can create our plot:

```
usage_percent %>%
  ggplot(aes(x="",y=use_type_percent, fill=usage)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  geom_text(aes(label = use_type_percent),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#006633","#00e673","#80ffbf"),
                    label = c("High use - 21 to 31 days",
                              "Moderate use - 11 to 20 days",
                              "Low use - 1 to 10 days"))+
  labs(title="Use of smart device")
```



Analyzing our results we can see that:

- 50% of the users of our sample use their device frequently - between 21 to 31 days.
- 12% use their device 11 to 20 days.
- 38% of our sample use really rarely their device.

5.7.2 Time used smart devices

Being more precise we want to see how many minutes do users wear their device per day. For that we will merge the created day_use data frame and activity to be able to filter results by daily use of device as well.

```
daily_use_merged <- merge(activity, day_use, by=c ("id"))

head(daily_use_merged)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    2016-05-07       11992           7.71             7.71
## 2 1503960366    2016-05-06       12159           8.03             8.03
## 3 1503960366    2016-05-01       10602           6.81             6.81
## 4 1503960366    2016-04-30       14673           9.25             9.25
## 5 1503960366    2016-04-12       13162           8.50             8.50
```

```
## 6 1503960366    2016-04-13       10735           6.97             6.97
##   logged_activities_distance very_active_distance moderately_active_distan
ce
## 1                          0                2.46                       2.
12
## 2                          0                1.97                       0.
25
## 3                          0                2.29                       1.
60
## 4                          0                3.56                       1.
42
## 5                          0                1.88                       0.
55
## 6                          0                1.57                       0.
69
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  3.13                         0                  37
## 2                  5.81                         0                  24
## 3                  2.92                         0                  33
## 4                  4.27                         0                  52
## 5                  6.06                         0                  25
## 6                  4.71                         0                  21
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    46                    175               833     1821
## 2                     6                    289               754     1896
## 3                    35                    246               730     1820
## 4                    34                    217               712     1947
## 5                    13                    328               728     1985
## 6                    19                    217               776     1797
##       date days_used    usage
## 1 07/05/16        25 high use
## 2 06/05/16        25 high use
## 3 01/05/16        25 high use
## 4 30/04/16        25 high use
## 5 12/04/16        25 high use
## 6 13/04/16        25 high use
```

We need to create a new data frame calculating the total amount of minutes users wore the device every day and creating three different categories:

- All day - device was worn all day.
- More than half day - device was worn more than half of the day.
- Less than half day - device was worn less than half of the day.

```
minutes_use <- daily_use_merged %>%
  mutate(total_minutes_used = very_active_minutes+fairly_active_minutes+light
ly_active_minutes+sedentary_minutes)%>%
  mutate (percent_minutes_used = (total_minutes_used/1440)*100) %>%
  mutate (worn = case_when(
    percent_minutes_used >= 100 ~ "All day",
    percent_minutes_used < 100 & percent_minutes_used >= 50~ "More than half
day",
    percent_minutes_used < 50 & percent_minutes_used > 0 ~ "Less than half da
y"
  ))
head(minutes_use)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    2016-05-07       11992           7.71             7.71
## 2 1503960366    2016-05-06       12159           8.03             8.03
## 3 1503960366    2016-05-01       10602           6.81             6.81
## 4 1503960366    2016-04-30       14673           9.25             9.25
## 5 1503960366    2016-04-12       13162           8.50             8.50
## 6 1503960366    2016-04-13       10735           6.97             6.97
##   logged_activities_distance very_active_distance moderately_active_distan
ce
## 1                          0                2.46                       2.
12
## 2                          0                1.97                       0.
25
## 3                          0                2.29                       1.
60
## 4                          0                3.56                       1.
42
## 5                          0                1.88                       0.
55
## 6                          0                1.57                       0.
69
```

```
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  3.13                         0                  37
## 2                  5.81                         0                  24
## 3                  2.92                         0                  33
## 4                  4.27                         0                  52
## 5                  6.06                         0                  25
## 6                  4.71                         0                  21
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    46                    175               833     1821
## 2                     6                    289               754     1896
## 3                    35                    246               730     1820
## 4                    34                    217               712     1947
## 5                    13                    328               728     1985
## 6                    19                    217               776     1797
##       date days_used    usage total_minutes_used percent_minutes_used
## 1 07/05/16        25 high use               1091             75.76389
## 2 06/05/16        25 high use               1073             74.51389
## 3 01/05/16        25 high use               1044             72.50000
## 4 30/04/16        25 high use               1015             70.48611
## 5 12/04/16        25 high use               1094             75.97222
## 6 13/04/16        25 high use               1033             71.73611
##                 worn
## 1 More than half day
## 2 More than half day
## 3 More than half day
## 4 More than half day
## 5 More than half day
## 6 More than half day
```

As we have done before, to better visualize our results we will create new data frames. In this case we will create four different data frames to arrange them later on on a same visualization.

- First data frame will show the total of users and will calculate percentage of minutes worn the device taking into consideration the three categories created.
- The three other data frames are filtered by category of daily users so that we can see also the difference of daily use and time use.

```
minutes_use_percent<- minutes_use%>%
  group_by(worn) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(worn) %>%
  summarise(total_percent = total / totals) %>%
  mutate(percent = scales::percent(total_percent))
##Minutes high use
minutes_high_use_percent<- minutes_use%>%
  group_by(worn) %>%
  filter(usage == "high use") %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(worn) %>%
  summarise(total_percent = total / totals) %>%
  mutate(percent = scales::percent(total_percent))
##Minutes moderate use
minutes_moderate_use_percent<- minutes_use%>%
  group_by(worn) %>%
  filter(usage == "moderate use") %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(worn) %>%
  summarise(total_percent = total / totals) %>%
  mutate(percent = scales::percent(total_percent))
##Minutes low use
minutes_low_use_percent<- minutes_use%>%
  group_by(worn) %>%
  filter(usage == "low use") %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(worn) %>%
  summarise(total_percent = total / totals) %>%
  mutate(percent = scales::percent(total_percent))
```

```
head(minutes_use_percent)
```

```
## # A tibble: 3 × 3
##   worn               total_percent percent
##   <chr>                      <dbl> <chr>
## 1 All day                   0.365  36%
## 2 Less than half day        0.0351 4%
## 3 More than half day        0.600  60%
```

```
head(minutes_high_use_percent)
```

```
## # A tibble: 3 × 3
##   worn               total_percent percent
##   <chr>                      <dbl> <chr>
## 1 All day                   0.0676 6.8%
## 2 Less than half day        0.0432 4.3%
## 3 More than half day        0.889  88.9%
```

```
head(minutes_moderate_use_percent)
```

```
## # A tibble: 3 × 3
##   worn               total_percent percent
##   <chr>                      <dbl> <chr>
## 1 All day                   0.267 27%
## 2 Less than half day        0.04  4%
## 3 More than half day        0.693 69%
```

```
head(minutes_low_use_percent)
```

```
## # A tibble: 3 × 3
##   worn               total_percent percent
##   <chr>                      <dbl> <chr>
## 1 All day                   0.802  80%
## 2 Less than half day        0.0224 2%
## 3 More than half day        0.175  18%
```

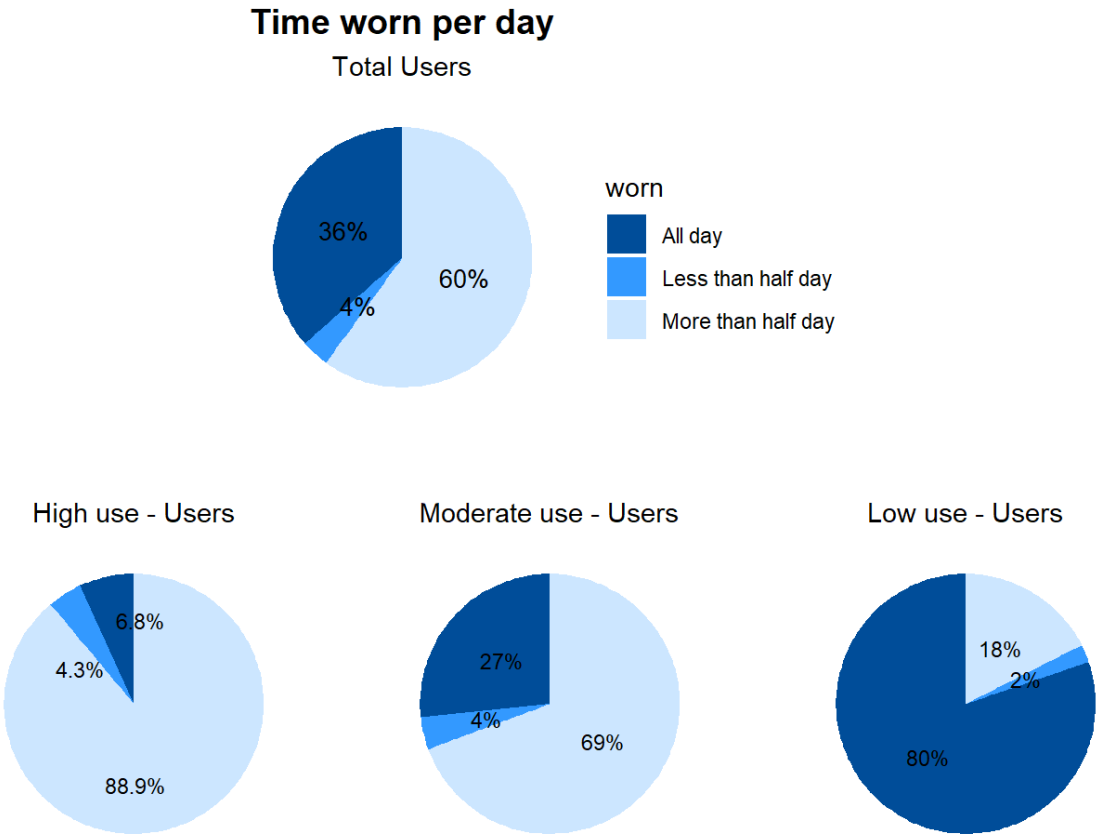
Now that we have created the four data frames and also ordered worn level categories, we can visualize our results in the following plots. All the plots have been arranged together for a better visualization.

```
ggarrange(
  ggplot(minutes_use_percent, aes(x="",y=total_percent, fill=worn)) +
    geom_bar(stat = "identity", width = 1)+
```

```
    coord_polar("y", start=0)+
    theme_minimal()+
    theme(axis.title.x= element_blank(),
          axis.title.y = element_blank(),
          panel.border = element_blank(),
          panel.grid = element_blank(),
          axis.ticks = element_blank(),
          axis.text.x = element_blank(),
          plot.title = element_text(hjust = 0.5, size=14, face = "bold"),
          plot.subtitle = element_text(hjust = 0.5)) +
    scale_fill_manual(values = c("#004d99", "#3399ff", "#cce6ff"))+
    geom_text(aes(label = percent),
              position = position_stack(vjust = 0.5), size = 3.5)+
    labs(title="Time worn per day", subtitle = "Total Users"),
  ggarrange(
    ggplot(minutes_high_use_percent, aes(x="",y=total_percent, fill=worn)) +
      geom_bar(stat = "identity", width = 1)+
      coord_polar("y", start=0)+
      theme_minimal()+
      theme(axis.title.x= element_blank(),
            axis.title.y = element_blank(),
            panel.border = element_blank(),
            panel.grid = element_blank(),
            axis.ticks = element_blank(),
            axis.text.x = element_blank(),
            plot.title = element_text(hjust = 0.5, size=14, face = "bold"),
            plot.subtitle = element_text(hjust = 0.5),
            legend.position = "none")+
      scale_fill_manual(values = c("#004d99", "#3399ff", "#cce6ff"))+
      geom_text_repel(aes(label = percent),
                      position = position_stack(vjust = 0.5), size = 3)+
      labs(title="", subtitle = "High use - Users"),
    ggplot(minutes_moderate_use_percent, aes(x="",y=total_percent, fill=worn)
) +
```

```
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5),
        legend.position = "none") +
  scale_fill_manual(values = c("#004d99", "#3399ff", "#cce6ff"))+
  geom_text(aes(label = percent),
            position = position_stack(vjust = 0.5), size = 3)+
  labs(title="", subtitle = "Moderate use - Users"),
ggplot(minutes_low_use_percent, aes(x="",y=total_percent, fill=worn)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5),
        legend.position = "none") +
  scale_fill_manual(values = c("#004d99", "#3399ff", "#cce6ff"))+
  geom_text(aes(label = percent),
            position = position_stack(vjust = 0.5), size = 3)+
  labs(title="", subtitle = "Low use - Users"),
ncol = 3),
```

```
  nrow = 2)
```



Per our plots we can see that 36% of the total of users wear the device all day long, 60% more than half day long and just 4% less than half day.

If we filter the total users considering the days they have used the device and also check each day how long they have worn the device, we have the following results:

Just a reminder:

- high use - users who use their device between 21 and 31 days.
- moderate use - users who use their device between 10 and 20 days.
- low use - users who use their device between 1 and 10 days.

High users - Just 6.8% of the users that have used their device between 21 and 31 days wear it all day. 88.9% wear the device more than half day but not all day. Moderate users are the ones who wear the device less on a daily basis. Being low users who wear more time their device the day they use it.

5.7.3 Another exploration

```
minutes_sleep_use <- inner_join(
  minutes_use,
```

```
  sleep,
  by = NULL,
  copy = FALSE,
  suffix = c("id","date"),
  keep = NULL
)
## Joining with `by = join_by(id, date)`
head(minutes_sleep_use)
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    2016-05-07       11992           7.71             7.71
## 2 1503960366    2016-05-06       12159           8.03             8.03
## 3 1503960366    2016-05-01       10602           6.81             6.81
## 4 1503960366    2016-04-30       14673           9.25             9.25
## 5 1503960366    2016-04-12       13162           8.50             8.50
## 6 1503960366    2016-04-13       10735           6.97             6.97
##   logged_activities_distance very_active_distance moderately_active_distan
ce
## 1                          0                2.46                       2.
12
## 2                          0                1.97                       0.
25
## 3                          0                2.29                       1.
60
## 4                          0                3.56                       1.
42
## 5                          0                1.88                       0.
55
## 6                          0                1.57                       0.
69
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  3.13                         0                  37
## 2                  5.81                         0                  24
## 3                  2.92                         0                  33
## 4                  4.27                         0                  52
## 5                  6.06                         0                  25
## 6                  4.71                         0                  21
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
```

```
## 1                    46                    175               833     1821
## 2                     6                    289               754     1896
## 3                    35                    246               730     1820
## 4                    34                    217               712     1947
## 5                    13                    328               728     1985
## 6                    19                    217               776     1797
##       date days_used    usage total_minutes_used percent_minutes_used
## 1 07/05/16        25 high use               1091             75.76389
## 2 06/05/16        25 high use               1073             74.51389
## 3 01/05/16        25 high use               1044             72.50000
## 4 30/04/16        25 high use               1015             70.48611
## 5 12/04/16        25 high use               1094             75.97222
## 6 13/04/16        25 high use               1033             71.73611
##                 worn  sleep_day total_sleep_records total_minutes_asleep
## 1 More than half day 2016-05-07                   1                  331
## 2 More than half day 2016-05-06                   1                  334
## 3 More than half day 2016-05-01                   1                  369
## 4 More than half day 2016-04-30                   1                  404
## 5 More than half day 2016-04-12                   1                  327
## 6 More than half day 2016-04-13                   2                  384
##   total_time_in_bed
## 1               349
## 2               367
## 3               396
## 4               425
## 5               346
## 6               407
all_day_use <- minutes_sleep_use %>%
  filter(worn == "All day") %>%
  mutate(actual_sedentary_minutes = sedentary_minutes - total_time_in_bed)
head(all_day_use)
##  [1] id                          activity_date
##  [3] total_steps                 total_distance
##  [5] tracker_distance            logged_activities_distance
```

```
##  [7] very_active_distance       moderately_active_distance
##  [9] light_active_distance      sedentary_active_distance
## [11] very_active_minutes        fairly_active_minutes
## [13] lightly_active_minutes     sedentary_minutes
## [15] calories                   date
## [17] days_used                  usage
## [19] total_minutes_used         percent_minutes_used
## [21] worn                       sleep_day
## [23] total_sleep_records        total_minutes_asleep
## [25] total_time_in_bed          actual_sedentary_minutes
## <0 rows> (or 0-length row.names)
```

```
#
minutes_use_o_sleep <- minutes_sleep_use %>%
  filter(worn != "All day") %>%
  mutate(real_total_minutes_use = total_minutes_used + total_time_in_bed)
head(minutes_use_o_sleep)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    2016-05-07       11992           7.71             7.71
## 2 1503960366    2016-05-06       12159           8.03             8.03
## 3 1503960366    2016-05-01       10602           6.81             6.81
## 4 1503960366    2016-04-30       14673           9.25             9.25
## 5 1503960366    2016-04-12       13162           8.50             8.50
## 6 1503960366    2016-04-13       10735           6.97             6.97
##   logged_activities_distance very_active_distance moderately_active_distan
ce
## 1                          0                2.46                       2.
12
## 2                          0                1.97                       0.
25
## 3                          0                2.29                       1.
60
## 4                          0                3.56                       1.
42
## 5                          0                1.88                       0.
55
## 6                          0                1.57                       0.
69
```

```
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  3.13                         0                  37
## 2                  5.81                         0                  24
## 3                  2.92                         0                  33
## 4                  4.27                         0                  52
## 5                  6.06                         0                  25
## 6                  4.71                         0                  21
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    46                   175               833     1821
## 2                     6                   289               754     1896
## 3                    35                   246               730     1820
## 4                    34                   217               712     1947
## 5                    13                   328               728     1985
## 6                    19                   217               776     1797
##       date days_used    usage total_minutes_used percent_minutes_used
## 1 07/05/16        25 high use               1091             75.76389
## 2 06/05/16        25 high use               1073             74.51389
## 3 01/05/16        25 high use               1044             72.50000
## 4 30/04/16        25 high use               1015             70.48611
## 5 12/04/16        25 high use               1094             75.97222
## 6 13/04/16        25 high use               1033             71.73611
##                 worn  sleep_day total_sleep_records total_minutes_asleep
## 1 More than half day 2016-05-07                   1                  331
## 2 More than half day 2016-05-06                   1                  334
## 3 More than half day 2016-05-01                   1                  369
## 4 More than half day 2016-04-30                   1                  404
## 5 More than half day 2016-04-12                   1                  327
## 6 More than half day 2016-04-13                   2                  384
##   total_time_in_bed real_total_minutes_use
## 1               349                  1440
## 2               367                  1440
## 3               396                  1440
## 4               425                  1440
## 5               346                  1440
```

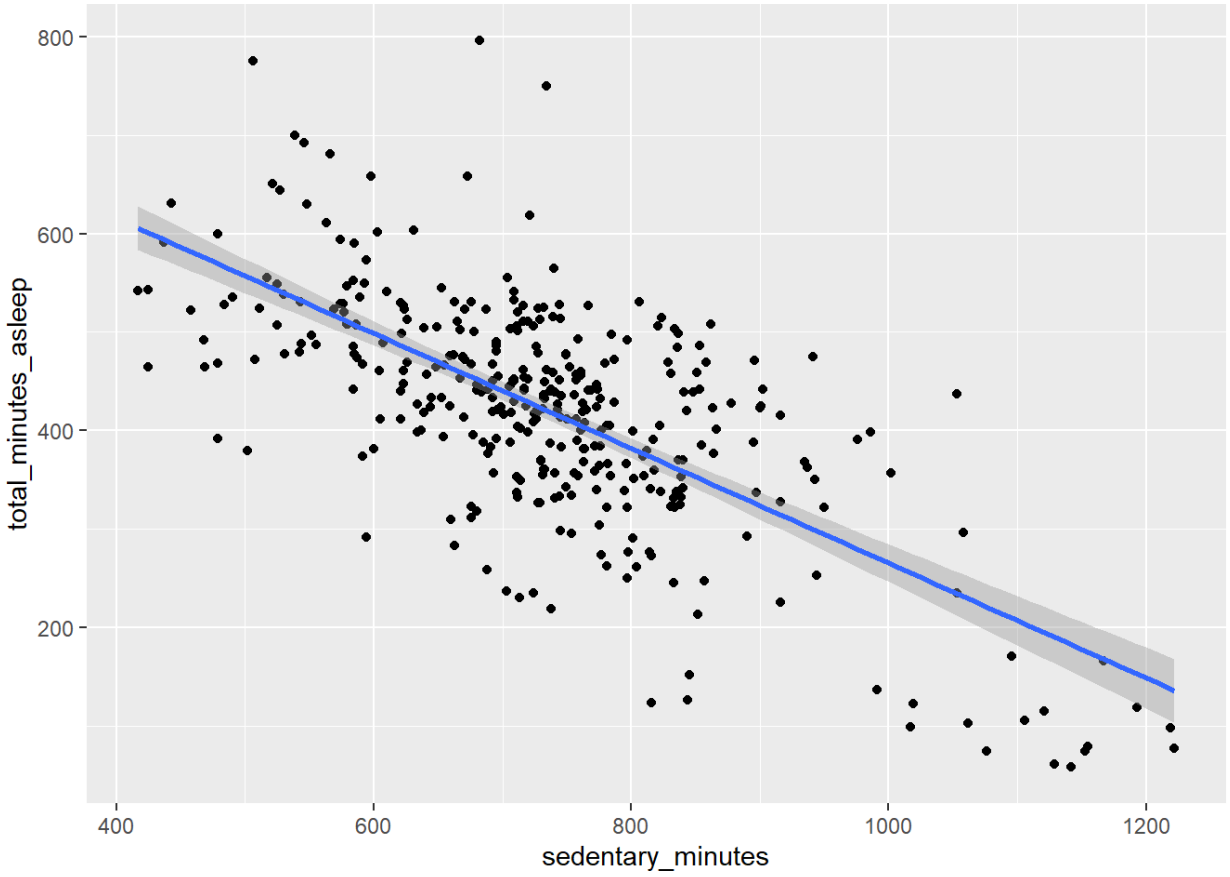
```
## 6                407                    1440
```

```
minutes_use_all_day <- minutes_use_o_sleep %>%
  filter(real_total_minutes_use >= 1380)
nrow(minutes_use_all_day)/nrow(minutes_use_o_sleep)*100
```

```
## [1] 87.56098
```

```
ggplot(minutes_use_all_day,aes(sedentary_minutes, total_minutes_asleep))+ geo
m_point()+geom_smooth(method = lm)
```

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



- Participants who use smart devices all day: They do not use sleep tracker function of their smart devices but they wear smart device when they sleep. So the sedentary minutes in their tracking data includes the time they are in bed.
- 87.5% of participants who do not use devices all day take their smart devices off when they sleep and wear it when they awake. So the sedentary minutes of their tracking data are the actual sedentary minutes.

```
minutes_use %>%
  group_by(usage) %>%
```

```
  summarise(mean(sedentary_minutes))
```

```
## # A tibble: 3 × 2
##   usage        `mean(sedentary_minutes)`
##   <chr>                            <dbl>
## 1 high use                          756.
## 2 low use                          1194.
## 3 moderate use                      872.
```

Participants with low use are physically-inactive people.

6.Conclusion (Act Phase)

Bellabeat's mission is to empower women by providing them with the data to discover themselves.

In order for us to respond to our business task and help Bellabeat on their mission, based on our results, I would advice to use own tracking data for further analysis. Datasets used have a small sample and can be biased since we didn't have any demographic details of users. Knowing that our main target are young and adult women I would encourage to continue finding trends to be able to create a marketing stragety focused on them.

That being said, after our analysis we have found different trends that may help our online campaign and improve **Bellabeat app**:

- There are total 33 participants in this data tracker. All of them are using smart devices to track their calories, intensity anh steps. However, just 24 participants (~72.72%), 14 participants (~42.42%) and 8 participants (~24.24%) use their smart devices to track their sleep, heart rate and weight respectively. User of Bellabeat smart devices are people who utilize smart devices to track when walking, going jogging rather than track their health. Therefore, Bellabeat marketing campaigns can focus on the set of customers **loving walking and jogging**. Moreover, Bellabeat may need an another research to analyze about why users are less likely to use smart devices to track their health in oder to improve their products.
- According to plot **5.2** and **5.7.3** users who have high sedentary minutes tend to have low sleep minutes. So company should have a messages to warn users when they are over 800 sedentary minutes in daytime. As an idea: if users want to improve their sleep, the Bellabeat app can recommend reducing sedentary time.
- We classified users into 4 categories and saw that the average of users walk more than 7,500 steps daily . We can encourage customers to reach at least daily recommended steps by CDC - 8.000 sending them alarms if they haven't reached the steps and creating also posts on our app explaining the benefits of reaching that goal. As CDC explains the more steps you walk the lower is the mortality rate. We also saw a positive correlation between steps and calories.
- Based on our results we can see that users sleep less than 8 hours a day. They could set up a desired time to go to sleep and receive a notification minutes before to prepare to sleep. Also offer helpfull resources to help customers sleep - ex. breathing advises, podcasts with relaxing music, sleep techniques.
- We are aware that some people don't get motivated by notifications so we could create a kind of game on our app for a limited period of time. Game would consist in reaching

different levels based on amount of steps walked every day. You need to maintain activity level for a period of time (maybe a month) to pass to the next level. For each level you would win certain amount of stars that would be redeemable for merchandise or discount on other Bellabeat products. Because the main object of Bellabeat marketing campaign is people interested in walking and jogging.

- Most activity happens between 5 pm and 7 pm - I suppose, that people go to a gym or for a walk after finishing work. Bellabeat can use this time to remind and motivate users to go for a run or walk.

Thank you for reading my analysis. **Bellabeat Case Study** This is my first project using R. I would appreciate any comments and recommendations for improvement!