FitBit Data Analysis For Bellabeat

Jyothi

2024-08-13

1. Ask

Business task

Analyze smart device usage data of **FitBit Fitness Tracker Data** in order to gain insight into how consumers use non-Bellabeat smart devices and apply these insights on one Bellabeat product.

Key stakeholders

- Urška Sršen: Bellabeat's co-founder and Chief Creative Officer.
- Sando Mur: Mathematician and Bellabeat's co-founder.
- Bellabeat marketing analytics team

2. Prepare

FitBit Fitness Tracker Data (https://www.kaggle.com/datasets/arashnic/fitbit) (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users.As it is an open source data and provided by well known entity.

The data set has 2 directories directory mturkfitbit_export_12.03.16-11.04.16 with 11 files directory mturkfitbit_export_12.04.16-12.05.16 with 18 files where 3 files are in both wide and long format

we need to merge 11 files common in both directories with different time period.

dailyActivity file = dailycalories + dailysteps + daily intensities

so merge only dailyActivity files from both directories. In Remaining files the data is in minutes,hours and seconds, if you want to go into minute analysis you can use all those files.

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(lubridate)
```

In this Analysis , we are taking only the dailyActivity data (calories,steps,activity details), sleepday , weightlog data and heart_rate data

```
daily_Activity1 <- read.csv("C:/Users/HI/Downloads/Bella_data/dailyActivity1.csv")
daily_Activity2 <- read.csv("C:/Users/HI/Downloads/Bella_data/dailyActivity2.csv")
daily_sleep <- read.csv("C:/Users/HI/Downloads/Bella_data/sleepDay_merged.csv")
weightinfo2 <- read.csv("C:/Users/HI/Downloads/Bella_data/weightLogInfo2.csv")
weightinfo1 <- read.csv("C:/Users/HI/Downloads/Bella_data/weightLogInfo1.csv")
heart_rate1 <- read.csv("C:/Users/HI/Downloads/Bella_data/heartrate_seconds1.csv")
heart_rate2 <- read.csv("C:/Users/HI/Downloads/Bella_data/heartrate_seconds2.csv")
```

3 Process

3.1 merging rows in common tables and removing duplicates

As the columns in both daily_Activity1 and daily_activity2 are same and the columns in both weightinfo1 and weightinfo2 are same we are binding the data of different months together.

We are removing duplicate rows by using Id, Date as primary key

```
df <- bind_rows(daily_Activity1,daily_Activity2) ## merging the rows of two tables
daily_Activity <- df %>% distinct(Id,ActivityDate, .keep_all = TRUE) ## using Id,Activity Date a
s primary key and removing duplicates

df1 <- bind_rows(weightinfo1,weightinfo2)
weight_info <- df1 %>% distinct(Id,Date, .keep_all = TRUE)

df2 <- bind_rows(heart_rate1,heart_rate2)
heart_rate <- df2 %>% distinct(Id,Time, .keep_all = TRUE)
```

We are not merging the three files because in the

- **daily_Activity** there are 35 respondents with 1373 rows
- **daily_sleep** we have 24 respondents with 413 rows
- **weight_info** we have 13 respondents with 98 rows

So if we merge all the 3 files with common data we will loose lot of data. so trying to analyse them individually

3.2 Viewing the datasets

```
view(weight_info)
view(daily_sleep)
view(daily_Activity)
view(heart_rate)
```

3.3 changing date to consistent format

Date is present in m/d/y format changing to d/m/y format in all the three files

```
daily_Activity <- daily_Activity %>%
  rename(Date = ActivityDate) %>%
  mutate(Date = as.Date(Date,"%m/%d/%Y"))
daily_sleep <- daily_sleep %>%
  rename(Date = SleepDay) %>%
  mutate(Date = as.Date(Date,"%m/%d/%Y"))
weight_info <- weight_info %>%
  mutate(Date = as.Date(Date,"%m/%d/%Y"))
heart_rate <- heart_rate %>%
  rename(Date = Time) %>%
  mutate(Date = as.Date(Date,"%m/%d/%Y"))
```

3.4 Removing some variables

```
weight_info <- weight_info %>% select(-c(Fat, LogId, IsManualReport))
daily_Activity <- daily_Activity %>% select(-c(TrackerDistance, LoggedActivitiesDistance))
daily_sleep <- daily_sleep %>% select(-c(TotalSleepRecords))
```

3.5 Adding new variables

We are adding column months which contains name of months

```
daily_Activity$months <- format(daily_Activity$Date, "%b")
daily_sleep$months <- format(daily_sleep$Date, "%b")
weight_info$months <- format(weight_info$Date, "%b")
```

we are adding column Weekday which contains week name

```
daily_Activity$weekday <- weekdays(as.Date(daily_Activity$Date))
daily_sleep$weekday <- weekdays(as.Date(daily_sleep$Date))
weight_info$weekday <- weekdays(as.Date(weight_info$Date))
```

3.6 Statistical summary of the variables

checking the data type and the statistical summary of each variable

```
str(heart_rate)
```

```
## 'data.frame':    3614915 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Date : Date, format: "2016-04-01" "2016-04-01" ...
##  $ Value: int  93 91 96 98 100 101 104 105 102 106 ...
```

```
str(weight_info)
```

```
## 'data.frame':    98 obs. of  7 variables:
##  $ Id          : num  1.50e+09 1.93e+09 2.35e+09 2.87e+09 2.87e+09 ...
##  $ Date        : Date, format: "2016-04-05" "2016-04-10" ...
##  $ WeightKg    : num  53.3 129.6 63.4 56.7 57.2 ...
##  $ WeightPounds: num  118 286 140 125 126 ...
##  $ BMI         : num  23 46.2 24.8 21.5 21.6 ...
##  $ months      : chr  "Apr" "Apr" "Apr" "Apr" ...
##  $ weekday     : chr  "Tuesday" "Sunday" "Sunday" "Wednesday" ...
```

```
str(daily_Activity)
```

```
## 'data.frame':    1373 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ Date                    : Date, format: "2016-03-25" "2016-03-26" ...
##  $ TotalSteps              : int  11004 17609 12736 13231 12041 10970 12256 12262 11248 10016
...
##  $ TotalDistance           : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ VeryActiveDistance      : num  2.57 6.92 4.66 3.19 2.16 ...
##  $ ModeratelyActiveDistance: num  0.46 0.73 0.16 0.79 1.09 ...
##  $ LightActiveDistance     : num  4.07 3.91 3.71 4.95 4.61 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  33 89 56 39 28 30 33 47 40 15 ...
##  $ FairlyActiveMinutes     : int  12 17 5 20 28 13 12 21 11 30 ...
##  $ LightlyActiveMinutes    : int  205 274 268 224 243 223 239 200 244 314 ...
##  $ SedentaryMinutes        : int  804 588 605 1080 763 1174 820 866 636 655 ...
##  $ Calories                : int  1819 2154 1944 1932 1886 1820 1889 1868 1843 1850 ...
##  $ months                  : chr  "Mar" "Mar" "Mar" "Mar" ...
##  $ weekday                 : chr  "Friday" "Saturday" "Sunday" "Monday" ...
```

```
str(daily_sleep)
```

```
## 'data.frame':    413 obs. of  6 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ Date              : Date, format: "2016-04-12" "2016-04-13" ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
##  $ months            : chr  "Apr" "Apr" "Apr" "Apr" ...
##  $ weekday           : chr  "Tuesday" "Wednesday" "Friday" "Saturday" ...
```

```
summary(daily_Activity)
```

```
##        Id                 Date              TotalSteps    TotalDistance
##  Min.   :1.504e+09   Min.   :2016-03-12   Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   1st Qu.:2016-04-08   1st Qu.: 3108   1st Qu.: 2.140
##  Median :4.445e+09   Median :2016-04-19   Median : 6910   Median : 4.900
##  Mean   :4.782e+09   Mean   :2016-04-19   Mean   : 7247   Mean   : 5.194
##  3rd Qu.:6.962e+09   3rd Qu.:2016-04-30   3rd Qu.:10538   3rd Qu.: 7.480
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :36019   Max.   :28.030
##  VeryActiveDistance ModeratelyActiveDistance LightActiveDistance
##  Min.   : 0.000     Min.   :0.0000           Min.   : 0.00
##  1st Qu.: 0.000     1st Qu.:0.0000           1st Qu.: 1.58
##  Median : 0.080     Median :0.1900           Median : 3.23
##  Mean   : 1.381     Mean   :0.5417           Mean   : 3.18
##  3rd Qu.: 1.810     3rd Qu.:0.7800           3rd Qu.: 4.68
##  Max.   :21.920     Max.   :6.4800           Max.   :12.51
##  SedentaryActiveDistance VeryActiveMinutes FairlyActiveMinutes
##  Min.   :0.000000        Min.   :  0.00    Min.   :  0.0
##  1st Qu.:0.000000        1st Qu.:  0.00    1st Qu.:  0.0
##  Median :0.000000        Median :  2.00    Median :  5.0
##  Mean   :0.001733        Mean   : 19.55    Mean   : 13.5
##  3rd Qu.:0.000000        3rd Qu.: 29.00    3rd Qu.: 18.0
##  Max.   :0.110000        Max.   :210.00    Max.   :660.0
##  LightlyActiveMinutes SedentaryMinutes    Calories       months
##  Min.   :  0.0        Min.   :   0.0   Min.   :   0   Length:1373
##  1st Qu.:108.0        1st Qu.: 729.0   1st Qu.:1793   Class :character
##  Median :195.0        Median :1058.0   Median :2114   Mode  :character
##  Mean   :184.6        Mean   : 993.4   Mean   :2264
##  3rd Qu.:260.0        3rd Qu.:1247.0   3rd Qu.:2766
##  Max.   :720.0        Max.   :1440.0   Max.   :4900
##    weekday
##  Length:1373
##  Class :character
##  Mode  :character
##
##
##
```

```
summary(daily_sleep)
```

```
##        Id                  Date                TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.504e+09   Min.   :2016-04-12   Min.   : 58.0      Min.   : 61.0
##  1st Qu.:3.977e+09   1st Qu.:2016-04-19   1st Qu.:361.0      1st Qu.:403.0
##  Median :4.703e+09   Median :2016-04-27   Median :433.0      Median :463.0
##  Mean   :5.001e+09   Mean   :2016-04-26   Mean   :419.5      Mean   :458.6
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04   3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :8.792e+09   Max.   :2016-05-12   Max.   :796.0      Max.   :961.0
##     months             weekday
##  Length:413         Length:413
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
summary(weight_info)
```

```
##        Id                  Date                  WeightKg       WeightPounds
##  Min.   :1.504e+09   Min.   :2016-03-30   Min.   : 52.60   Min.   :116.0
##  1st Qu.:6.962e+09   1st Qu.:2016-04-08   1st Qu.: 61.50   1st Qu.:135.6
##  Median :6.962e+09   Median :2016-04-19   Median : 62.50   Median :137.8
##  Mean   :6.812e+09   Mean   :2016-04-20   Mean   : 72.47   Mean   :159.8
##  3rd Qu.:8.878e+09   3rd Qu.:2016-05-01   3rd Qu.: 85.25   3rd Qu.:187.9
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :133.50   Max.   :294.3
##       BMI             months             weekday
##  Min.   :21.45   Length:98          Length:98
##  1st Qu.:24.00   Class :character   Class :character
##  Median :24.39   Mode  :character   Mode  :character
##  Mean   :25.37
##  3rd Qu.:25.59
##  Max.   :47.54
```

4 Analyze

Dataset generated by respondents to a distributed survey via Amazon Mechanical Turk between 12.03.2016-12.05.2016 i.e., (62 days)

- March - 20 days - 11 respondents
- April - 30 days - 24(new) + 11 = 35 respondents
- May - 12 days - 35-3(11)-2(24) = 30 respondents

Totally there are 35 respondents

8 respondents data we have for 3 months

- 1503960366
- 1624580081
- 4020332650
- 4319703577
- 4388161847
- 4445114986

- 4702921684
- 6962181067

3 repondents who are in March are not there in May

- 2347167796 – Active for 32 days
- 2891001357 – Active for 8 days
- 4057192912 – Active for 35 days

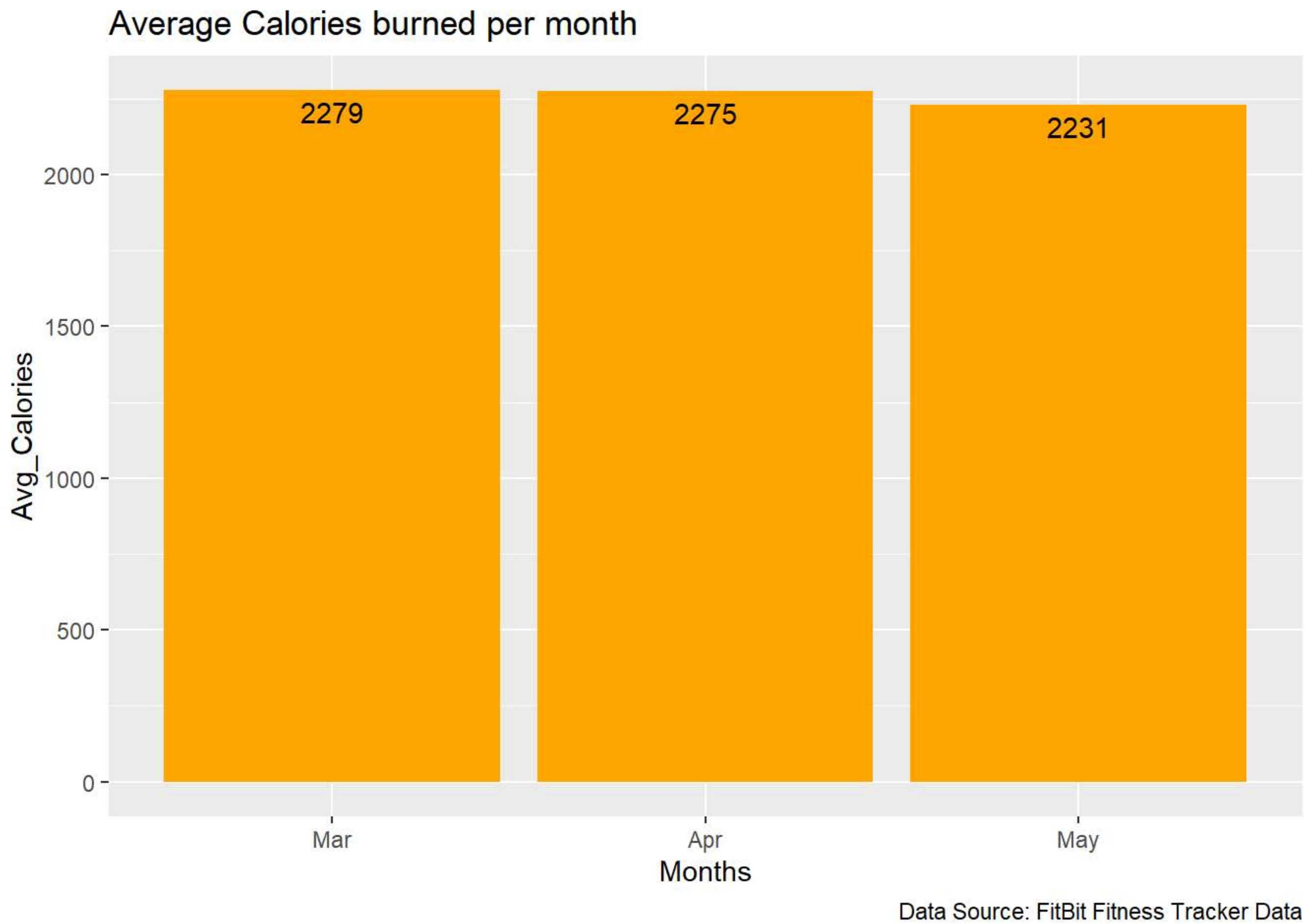
2 repondents who got added in April are not there in May

- 6391747486 – Active for 9 days
- 8253242879 – Active for 30 days

For Analysis I have used Average, as the sum will be misleading as the no. of respondents changed each month and the no. of days data we have vary per month

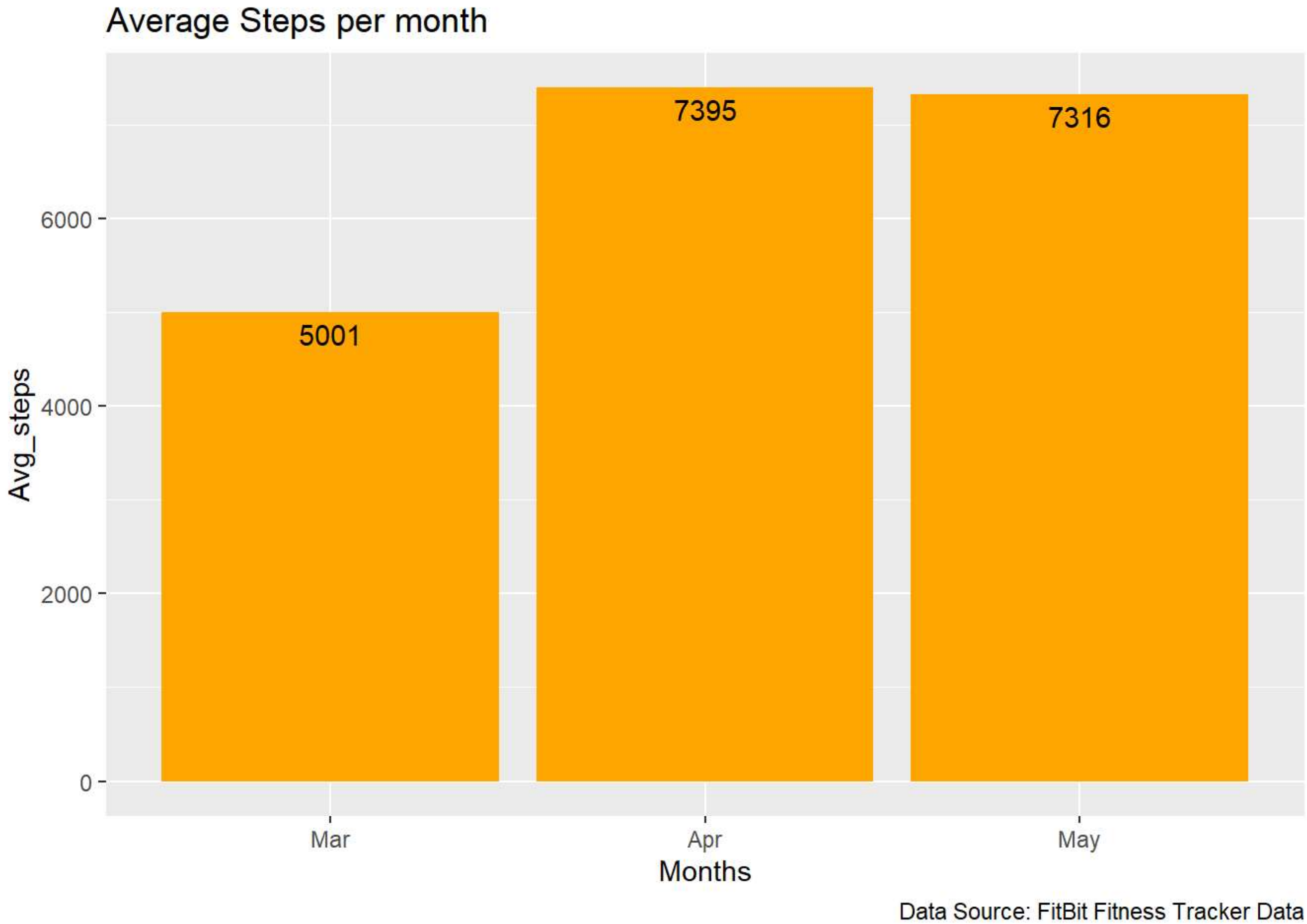
4.1 Average calories burnt per month

```
daily_Activity %>%  group_by(months) %>%
  summarise(avg_cal = mean(Calories)) %>%
  ggplot(aes(x = months, y = avg_cal)) +
  geom_col(fill = "orange") +
  scale_x_discrete(limits = c("Mar","Apr","May"))+
  geom_text(aes(label = round(avg_cal),vjust = 1.5)) +
  labs(title = 'Average Calories burned per month',
       x = 'Months',
       y = 'Avg_Calories',
       caption = 'Data Source: FitBit Fitness Tracker Data')
```



4.2 Average Steps per month

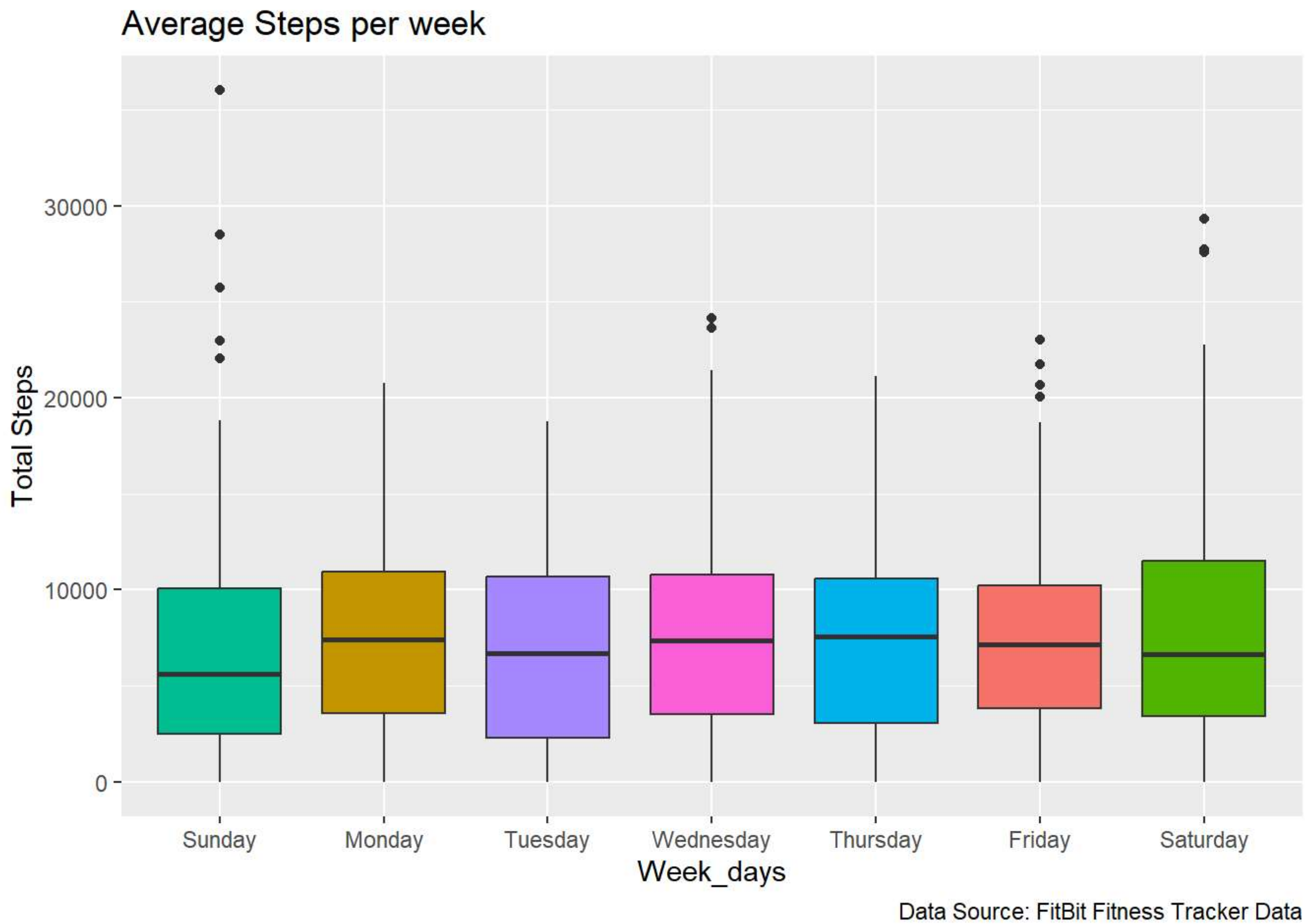
```
daily_Activity %>%
  group_by(months) %>%
  summarise(avg_steps = mean(TotalSteps)) %>%
ggplot(aes(x = months, y = avg_steps)) +
  geom_col(fill = "orange") +
  scale_x_discrete(limits = c("Mar","Apr","May"))+
  geom_text(aes(label = round(avg_steps),vjust = 1.5)) +
  labs(title = 'Average Steps per month',
       x = 'Months',
       y = 'Avg_steps',
       caption = 'Data Source: FitBit Fitness Tracker Data')
```



- As there are less respondents in March we can see the huge difference in steps

4.3 Average Steps per week

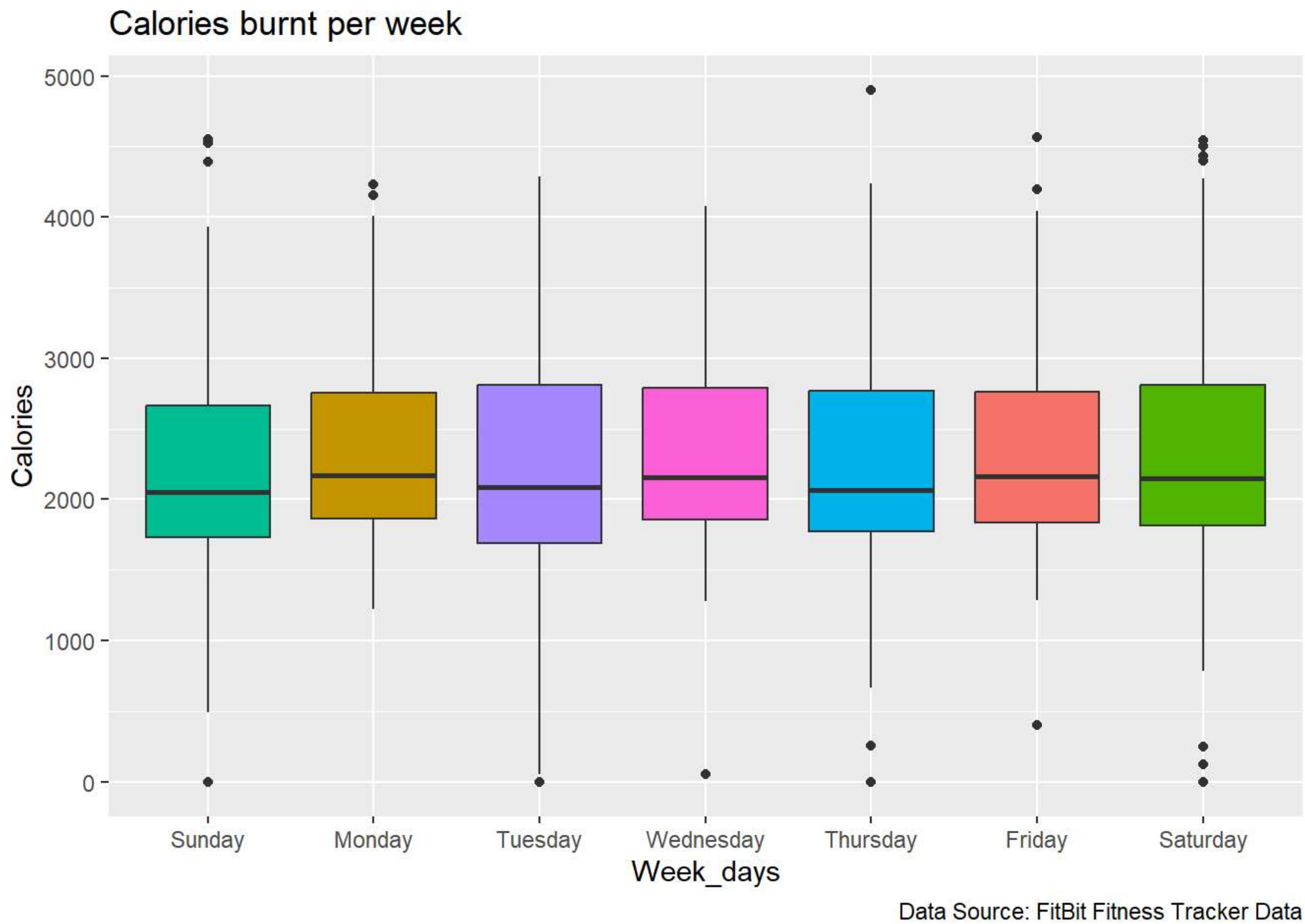
```
daily_Activity %>%
  ggplot(aes(weekday, TotalSteps,fill = weekday)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Satur
day"))+
  labs(title = 'Average Steps per week',
       x = 'Week_days',
       y = 'Total Steps',
       caption = 'Data Source: FitBit Fitness Tracker Data') + theme(legend.position = "none")
```



- Approximately, Average 6600 - 7800 steps are taken weekly. Black dots indicates Steps which are extremely high than normal steps

4.4 Average Calories burnt per week

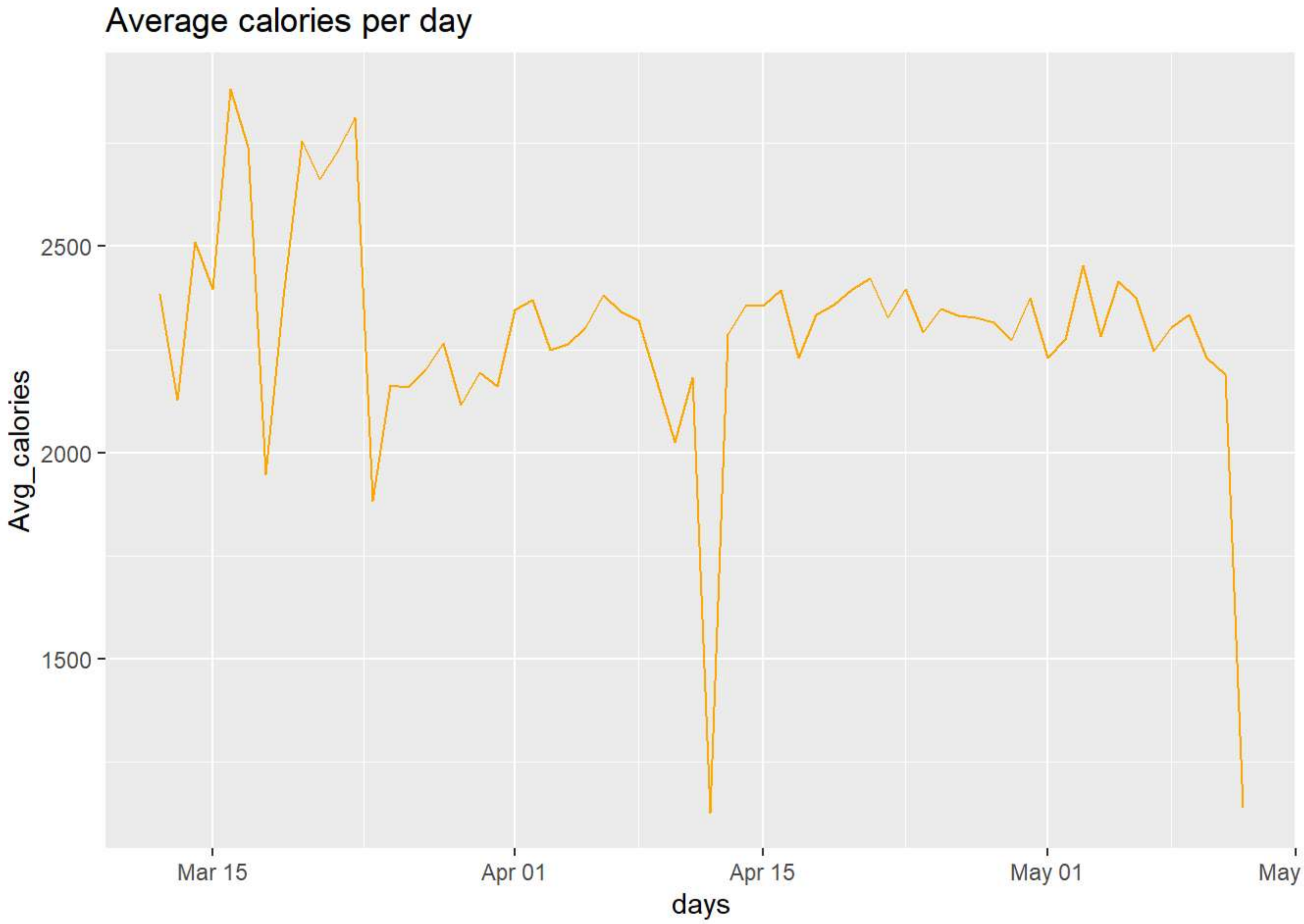
```
daily_Activity %>%
  ggplot(aes(weekday, Calories, fill = weekday)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Satur
day"))+
  labs(title = 'Calories burnt per week',
       x = 'Week_days',
       y = 'Calories',
       caption = 'Data Source: FitBit Fitness Tracker Data') + theme(legend.position = "none")
```



- 2200-2400 calories are averagely burnt weekly

4.4 Average calories burnt per day

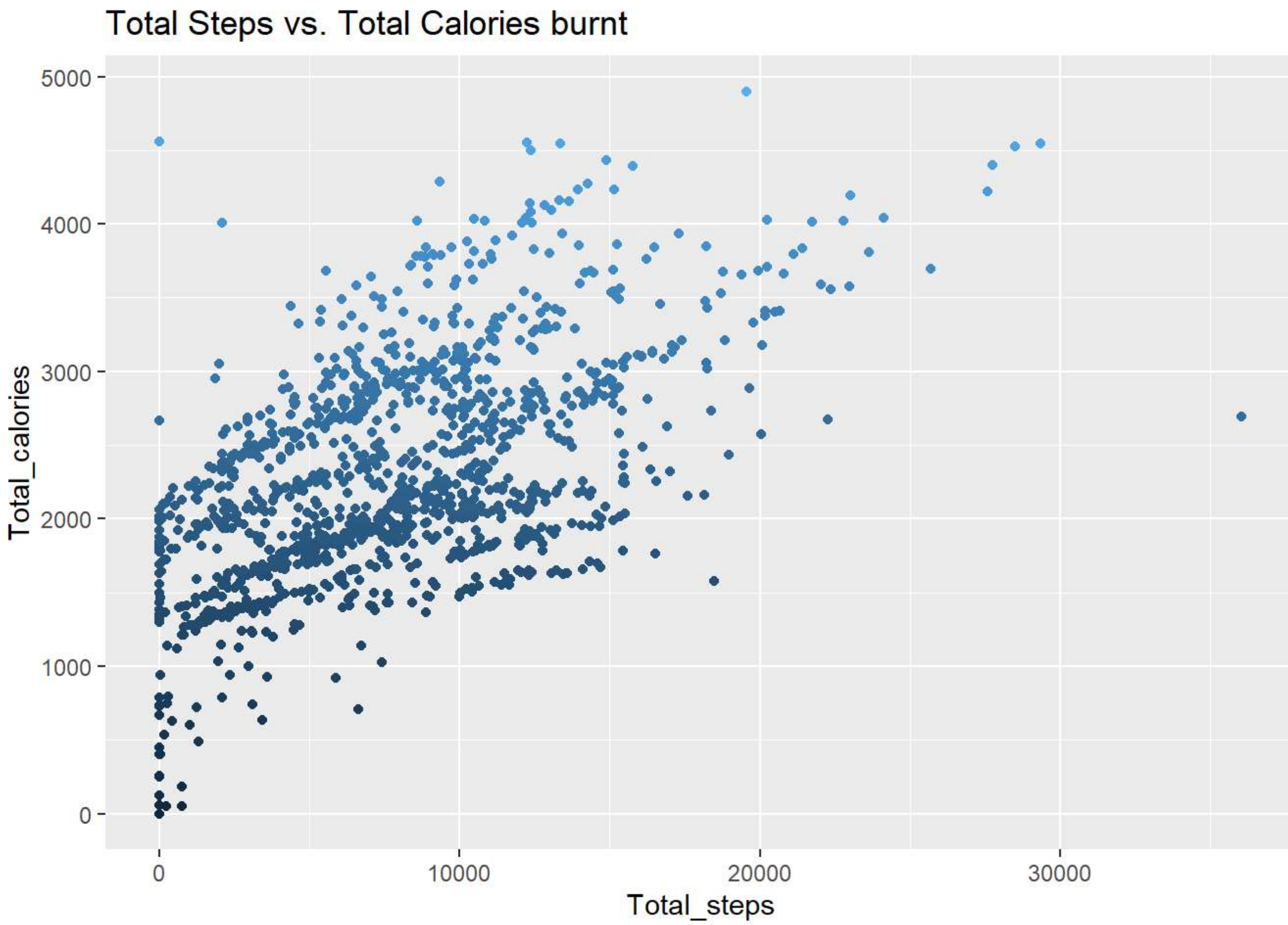
```
daily_Activity %>%
  group_by(Date) %>%
  summarise(avg_calories = mean(Calories)) %>%
  ggplot(aes(x = Date, y = avg_calories)) + geom_line(color = "orange") +
  labs(title = 'Average calories per day',
       x = 'days',
       y = 'Avg_calories')
```



- Average calories burnt in march is highly fluctuation as there are less respondents

4.4 Total Calories burnt vs. Total Steps

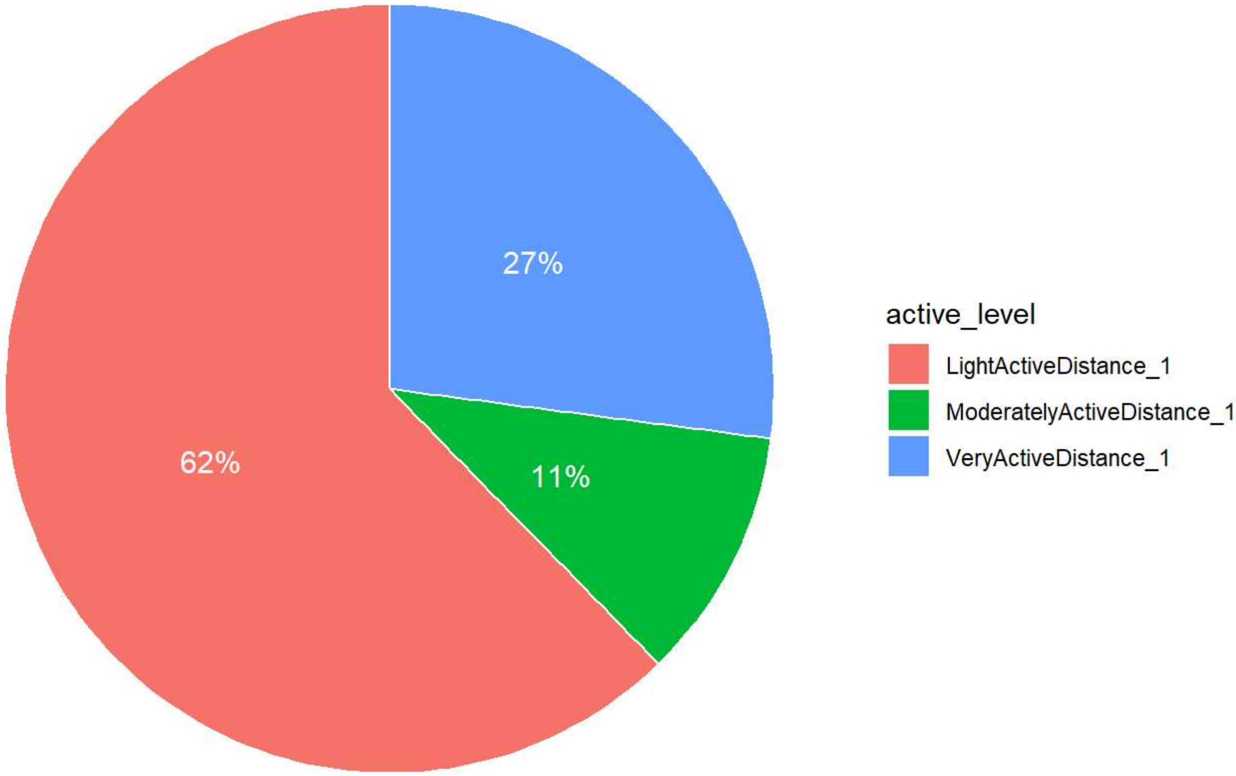
```
daily_Activity %>%  group_by(TotalSteps,Calories) %>%
  ggplot(aes(x = TotalSteps, y = Calories,color = Calories)) +
  geom_point() +
  labs(title = 'Total Steps vs. Total Calories burnt',
       x = 'Total_steps',
       y = 'Total_calories') +
  theme(legend.position = "none")
```



- We can say that total steps is directly proportional to calories burnt. As the no. of steps increases the Calories burnt increases

4.5 Active level by distance

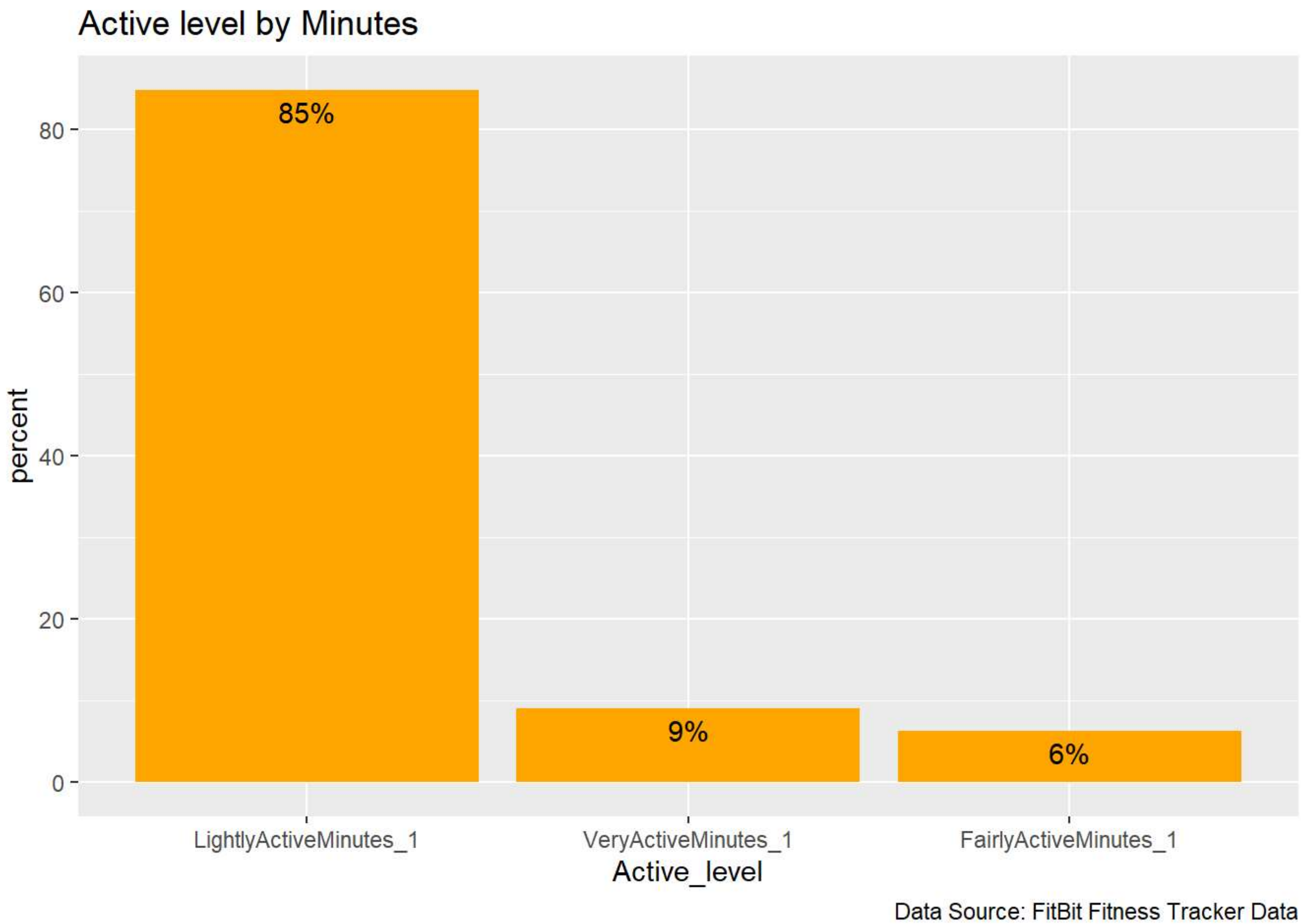
```
daily_Activity %>%
  select(VeryActiveDistance,ModeratelyActiveDistance,LightActiveDistance) %>%
  summarise(across(everything(),list(sum))) %>%
  gather(active_level,distance) %>%
  mutate(percent = (distance/sum(distance))*100) %>%
  mutate(ypos = cumsum(percent)- 0.5*percent ) %>%
  ggplot(aes(x="", y=percent, fill=active_level)) +
  geom_bar(stat="identity", width=1,color = "white") +
  coord_polar("y",start = 0)+theme_void() +
  geom_text(aes(y=ypos,label = paste0(round(percent),'%')),color = "white",size =4)
```



- 62% of distance covered by respondents is Light Active Distance, 27% is Moderately Active Distance, 11% is very active distance

4.6 Active level by Minutes

```
daily_Activity %>%
  select(VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes) %>%
  summarise(across(everything(), list(sum))) %>%
  gather(active_level, minutes) %>%
  mutate(perct = (minutes/sum(minutes))*100) %>%
  ggplot(aes(x = reorder(active_level,-perct), y = perct)) + geom_col(fill = "orange") +
  geom_text(aes(label = paste0(round(perct),'%')),vjust = 1.5) +
  labs(title = 'Active level by Minutes',
       x = 'Active_level',
       y = 'percent',
       caption = 'Data Source: FitBit Fitness Tracker Data')
```



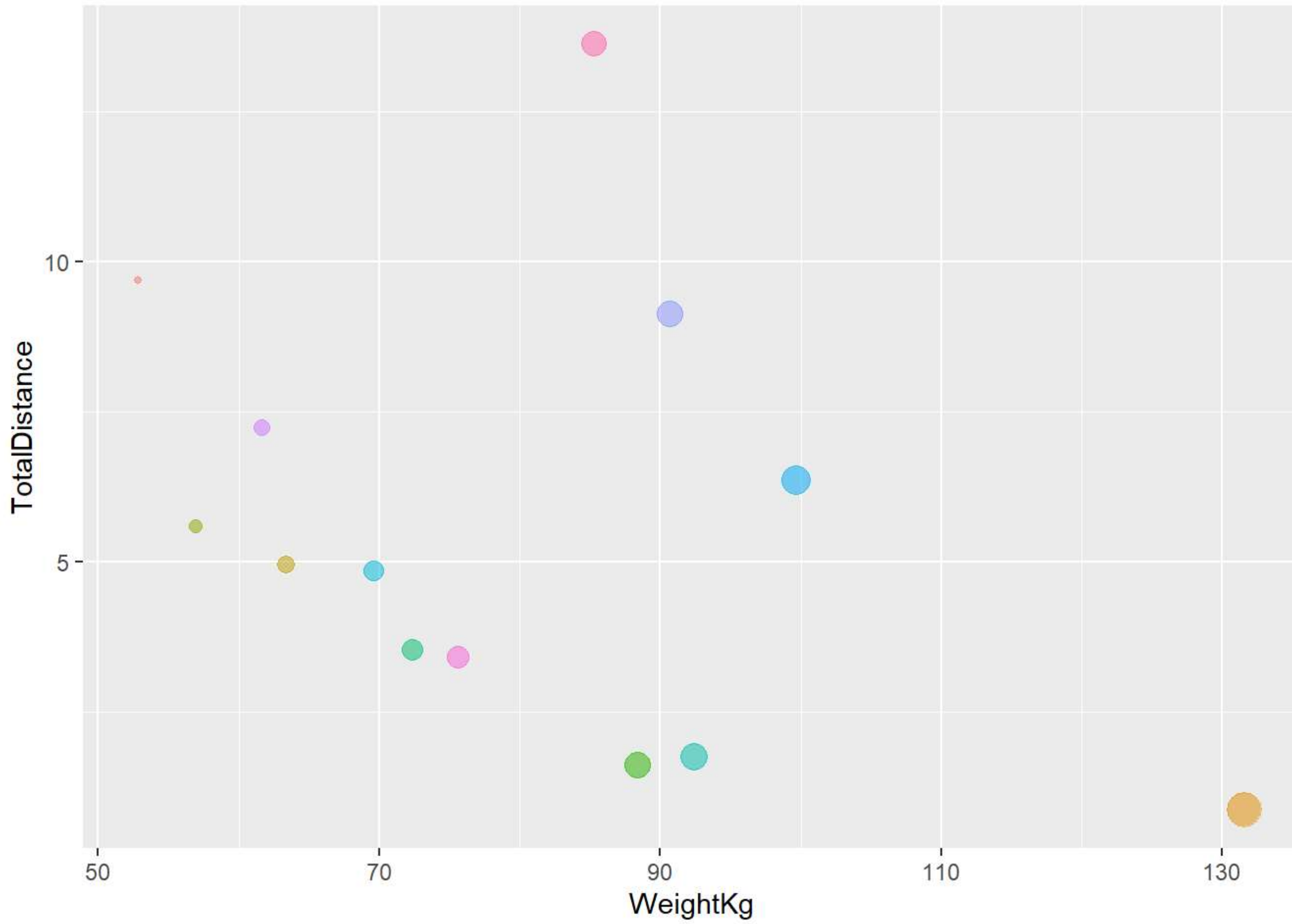
- Lightly Active minutes is the highest

Merging daily_Activity and weight info to see the correlation between distance and weight of respondents

```
dist_weight <- merge(x = daily_Activity,y = weight_info[,c("Id","Date","WeightKg")], by = c('I
d', 'Date'))
```

4.7 Total distance vs. Weight in kgs

```
dist_weight %>%
 select(Id, WeightKg, TotalDistance) %>%
 group_by(Id) %>%
 summarise_all(list(~mean(.))) %>%
 drop_na() %>%
 mutate(Id = factor(Id)) %>%
 ggplot(aes(WeightKg, TotalDistance, fill = Id)) +
 geom_point(aes(color = Id, size = WeightKg), alpha = 0.5) +
 theme(legend.position = "none")
```



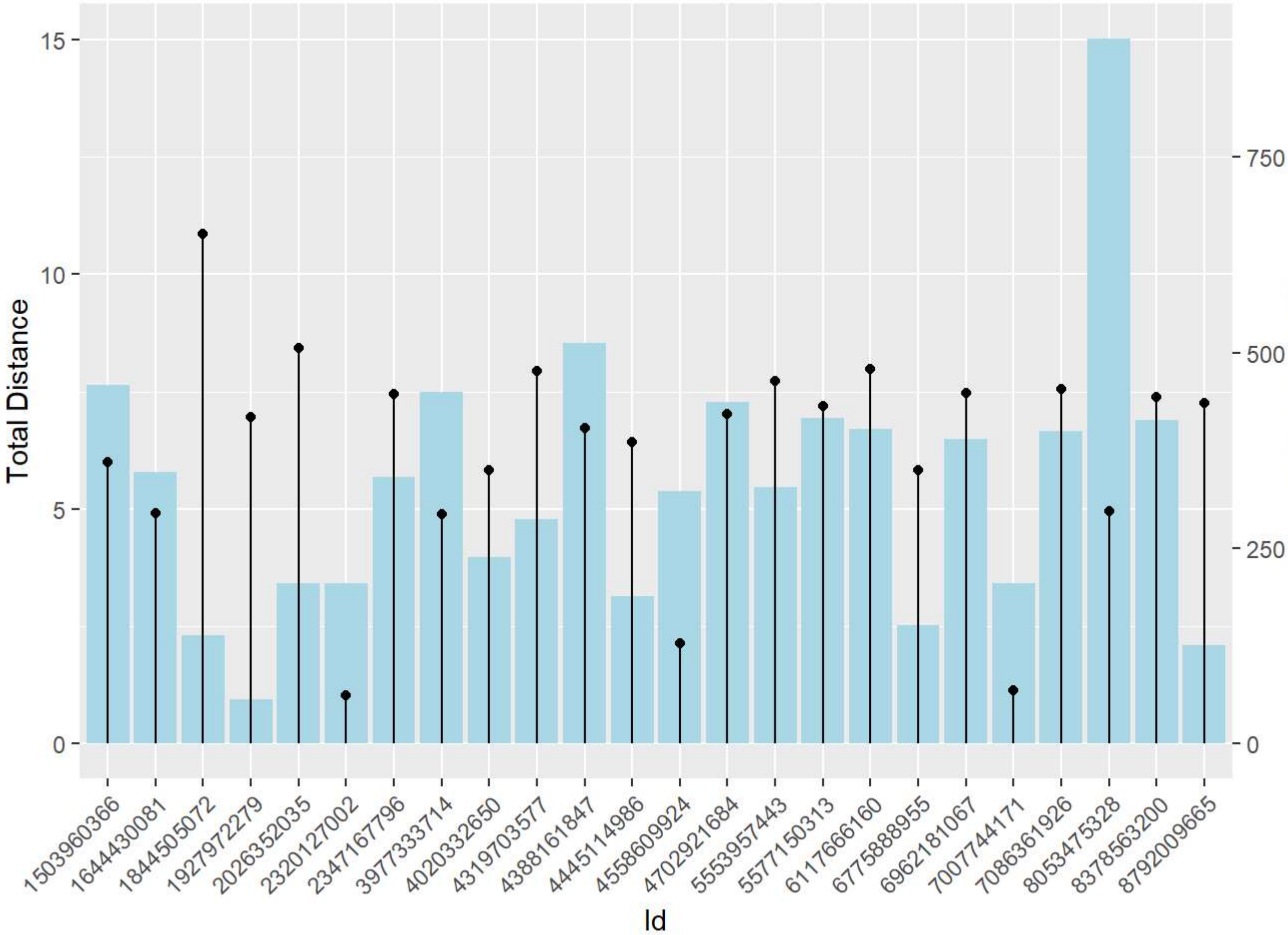
- As the weight of person increases the distance covered is reduced

Merging daily_Activity and daily_sleep to see the correlation between distance and sleep of respondents

```
dist_sleep <- merge(x = daily_Activity,y = daily_sleep[,c("Id","Date","TotalTimeInBed","TotalMin
utesAsleep")], by = c('Id', 'Date'))
```

4.8 Average distance vs. average sleep per user

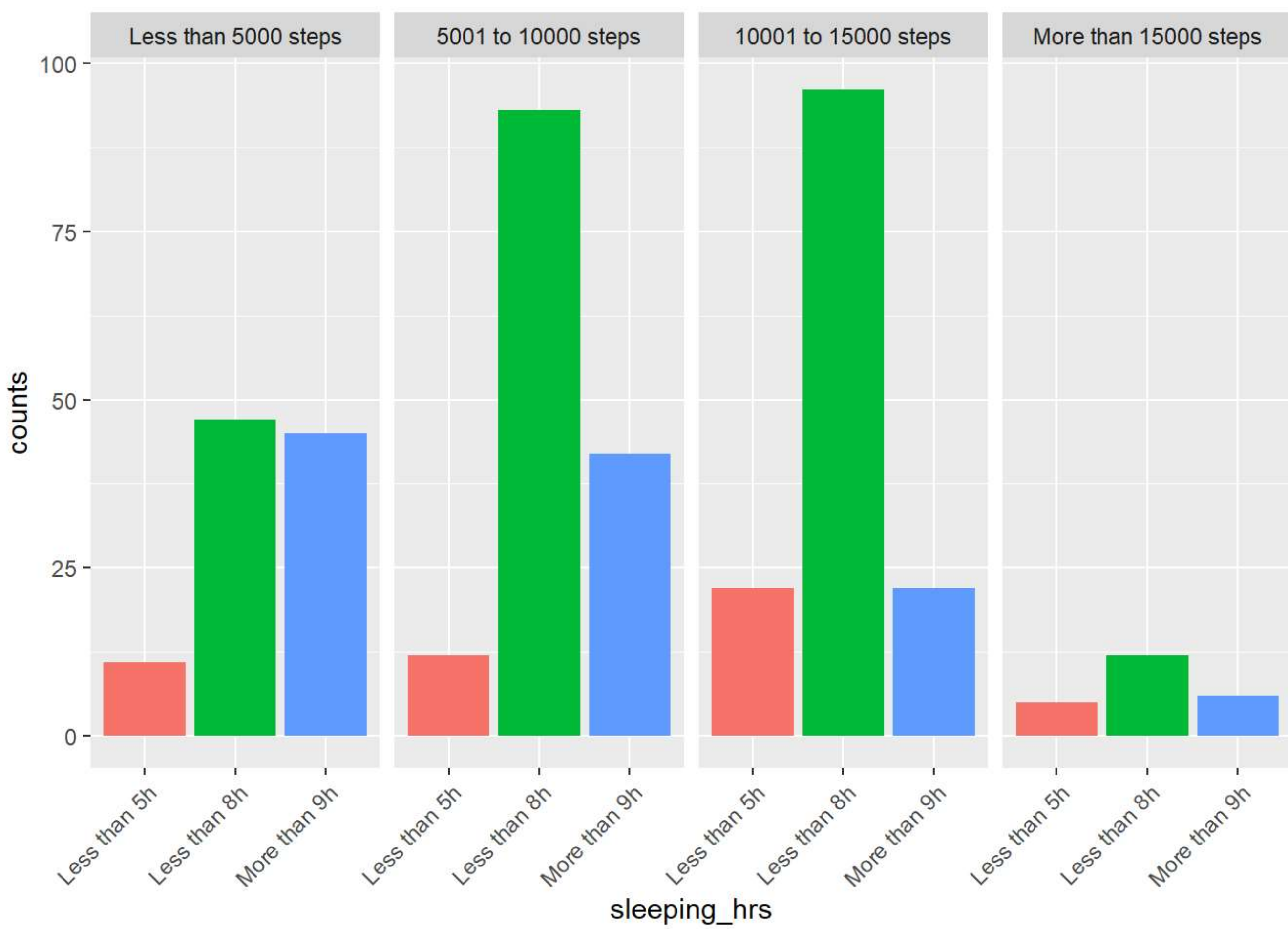
```
dist_sleep %>%
  select(Id, TotalMinutesAsleep, TotalDistance) %>%
  group_by(Id) %>%
  summarise_all(list(~mean(.))) %>%
  drop_na() %>%
  mutate(Id = factor(Id)) %>%
  ggplot()+
  geom_bar(aes(x = Id, y = TotalDistance), stat = "identity", fill = 'lightblue') +
  geom_point(aes(x = Id, y = TotalMinutesAsleep/60), color = 'black')+
  geom_segment(aes(x = Id,xend = Id, yend = TotalMinutesAsleep/60,y=0), color = 'black') +
  scale_y_continuous(name = "Total Distance",
    sec.axis = sec_axis(~.*60, name = "Sleep in minutes")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



- We can’t find any relation as there are respondents with good sleep without covering minimum distance and people covering greater distance who are not having good sleep

4.9 Sleep quality by steps

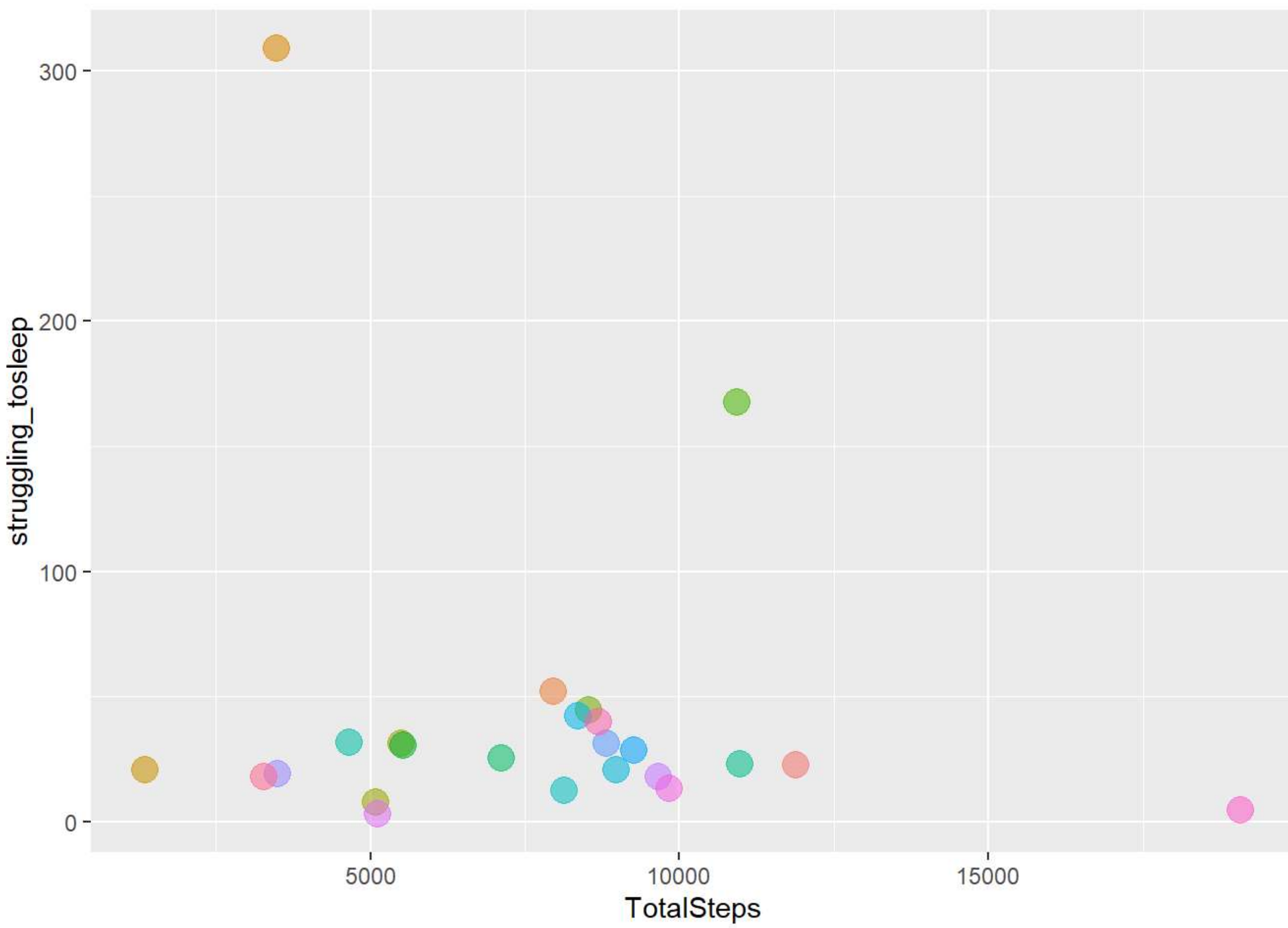
```
dist_sleep %>%
  select(TotalMinutesAsleep, TotalSteps) %>%
  mutate(sleeping_hrs = ifelse(TotalMinutesAsleep <= 300,'Less than 5h',
                        ifelse(TotalMinutesAsleep <= 480,'Less than 8h',
                        'More than 9h'))) %>%
  mutate(steps_taken = ifelse(TotalSteps <= 5000, 'Less than 5000 steps',
                       ifelse(TotalSteps <= 10000, '5001 to 10000 steps',
                       ifelse(TotalSteps <= 15000, '10001 to 15000 steps',
                              'More than 15000 steps')))) %>%
  group_by(sleeping_hrs,steps_taken) %>%
  summarise(counts = n(),.groups = "drop_last") %>%  # n() gives current group size # .groups dr
ops the steps_taken
  mutate(active_level = factor(steps_taken,
                               levels = c('Less than 5000 steps',
                                          '5001 to 10000 steps',
                                          '10001 to 15000 steps',
                                          'More than 15000 steps'))) %>%

  ggplot(aes(x = sleeping_hrs,
             y = counts,
             fill = sleeping_hrs)) +
  geom_bar(stat = "identity") +
  facet_wrap(~active_level, nrow = 1) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "none")
```



- Moderately Walking between 5000 to 15000 steps can help to have good sleep between 5 hrs to 9hrs
- people walking more than 15000 steps are not having enough sleep

4.10 struggling to sleep vs. Steps taken

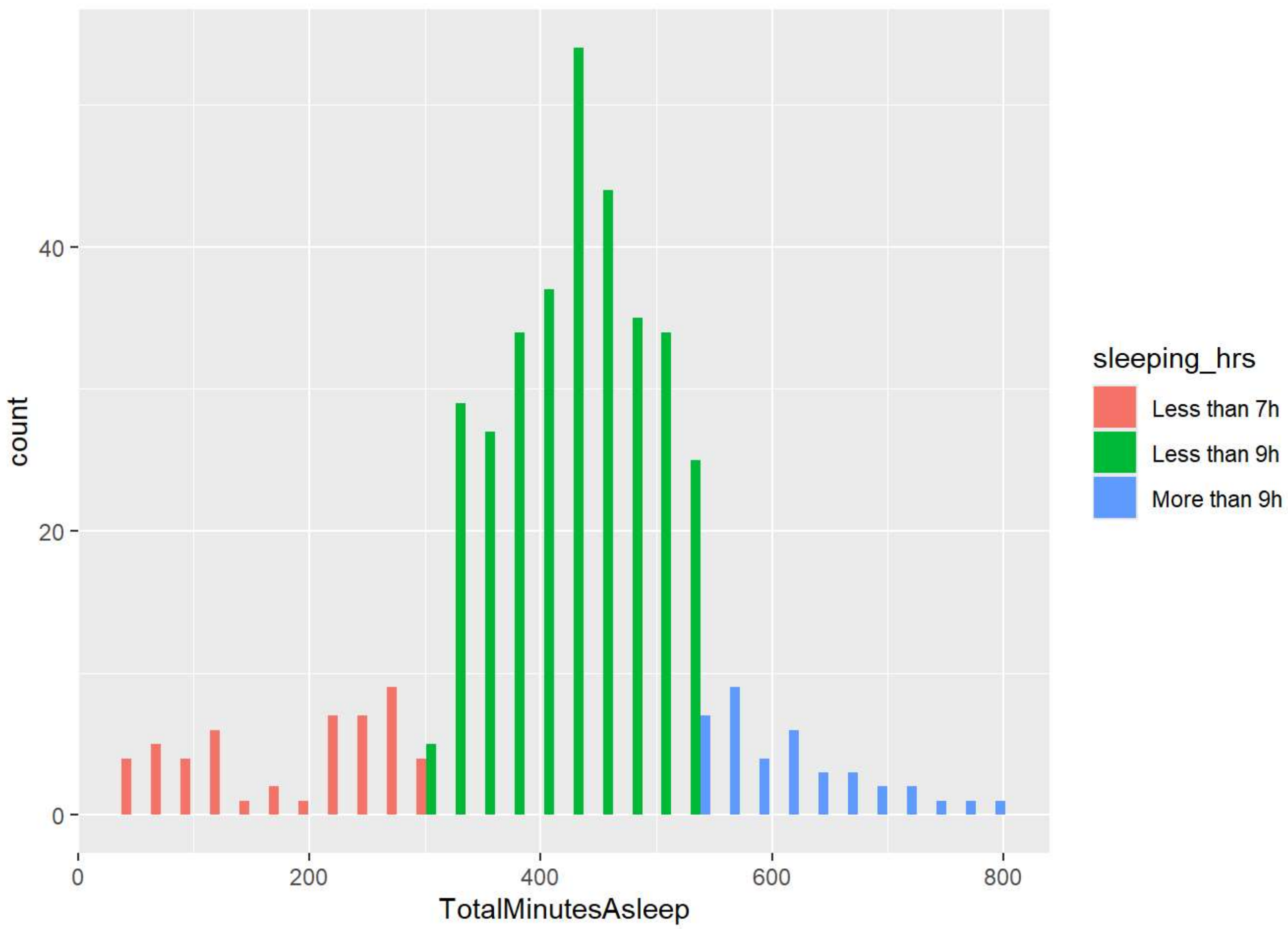
```
dist_sleep %>%
  select(Id,TotalSteps,TotalMinutesAsleep,TotalTimeInBed) %>%
  mutate(struggling_tosleep = TotalTimeInBed - TotalMinutesAsleep) %>%
  group_by(Id) %>%
  summarise_all(list(~mean(.))) %>%
  mutate(Id = factor(Id)) %>%
  ggplot(aes(x=TotalSteps,y =struggling_tosleep,color = Id,size =2,alpha = 0.5)) +
  geom_point() +
  theme(legend.position = "none")
```



- Majority of people are sleeping within 1 hour in bed and the extreme cases seem to be outliers

4.10 Average sleep per user

```
dist_sleep %>%
 select(TotalMinutesAsleep, TotalSteps) %>%
  mutate(sleeping_hrs = ifelse(TotalMinutesAsleep <= 300,'Less than 7h',
                        ifelse(TotalMinutesAsleep <= 540,'Less than 9h',
                        'More than 9h'))) %>%
  ggplot(aes(x=TotalMinutesAsleep,fill = sleeping_hrs)) +geom_histogram(position = 'dodge',bins
= 30)
```



- Most people have sleep between 300-520 minutes (i.e.,5hrs - 8.5 hrs)

4.11 Average heart_rate per user

As the heart_rate is in seconds, we are taking average heart_rate per day per respondent

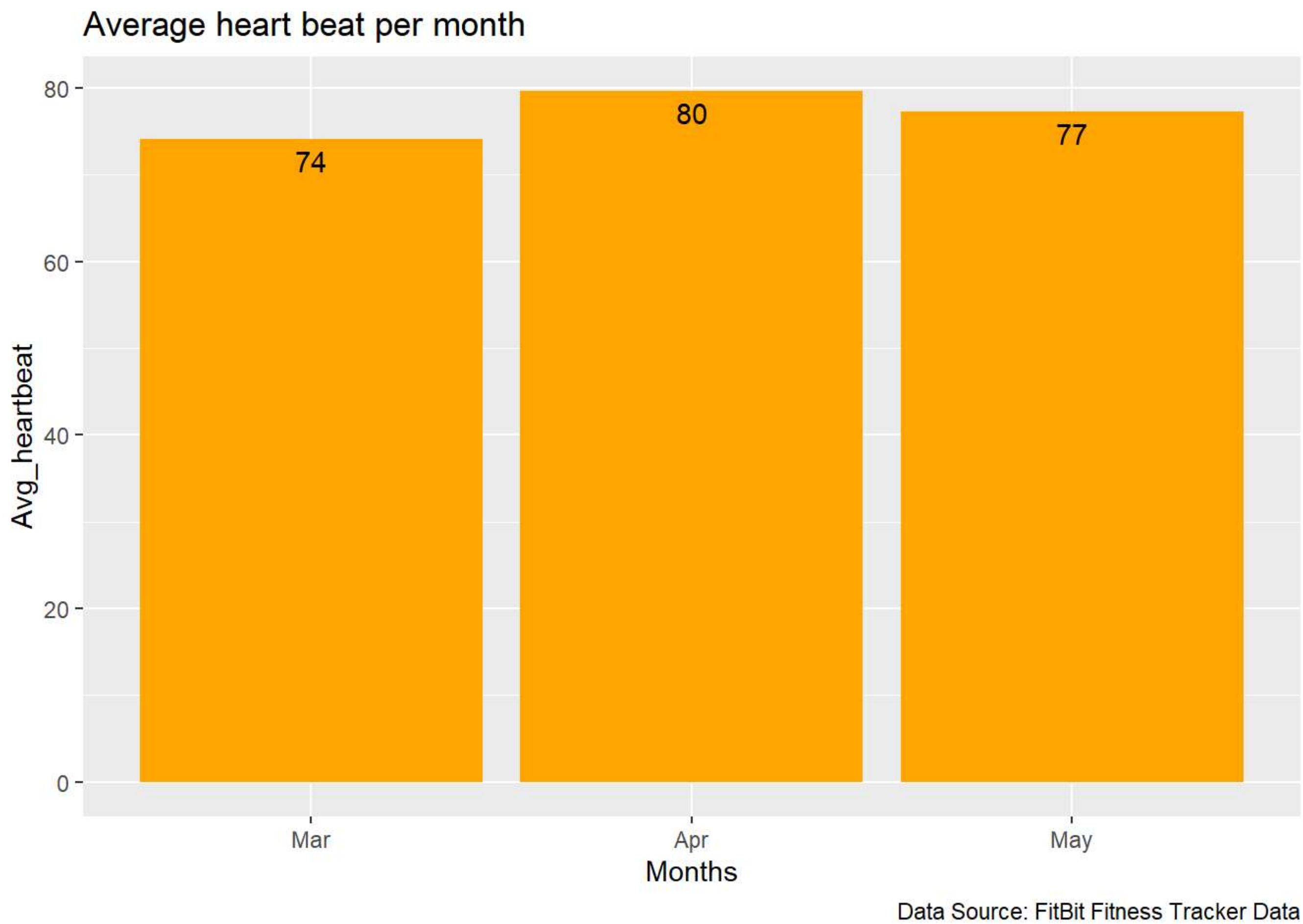
```
heart_rate_daily <- heart_rate %>% group_by(Id,Date) %>%  summarise(heart_rate = mean(Value), .g
roups = "drop_last")
```

Adding months and weekday column to the heart_rate data

```
heart_rate_daily$months <- format(heart_rate_daily$Date, "%b")
heart_rate_daily$weekday <- weekdays(as.Date(heart_rate_daily$Date))
```

4.11 Average heart_rate per month

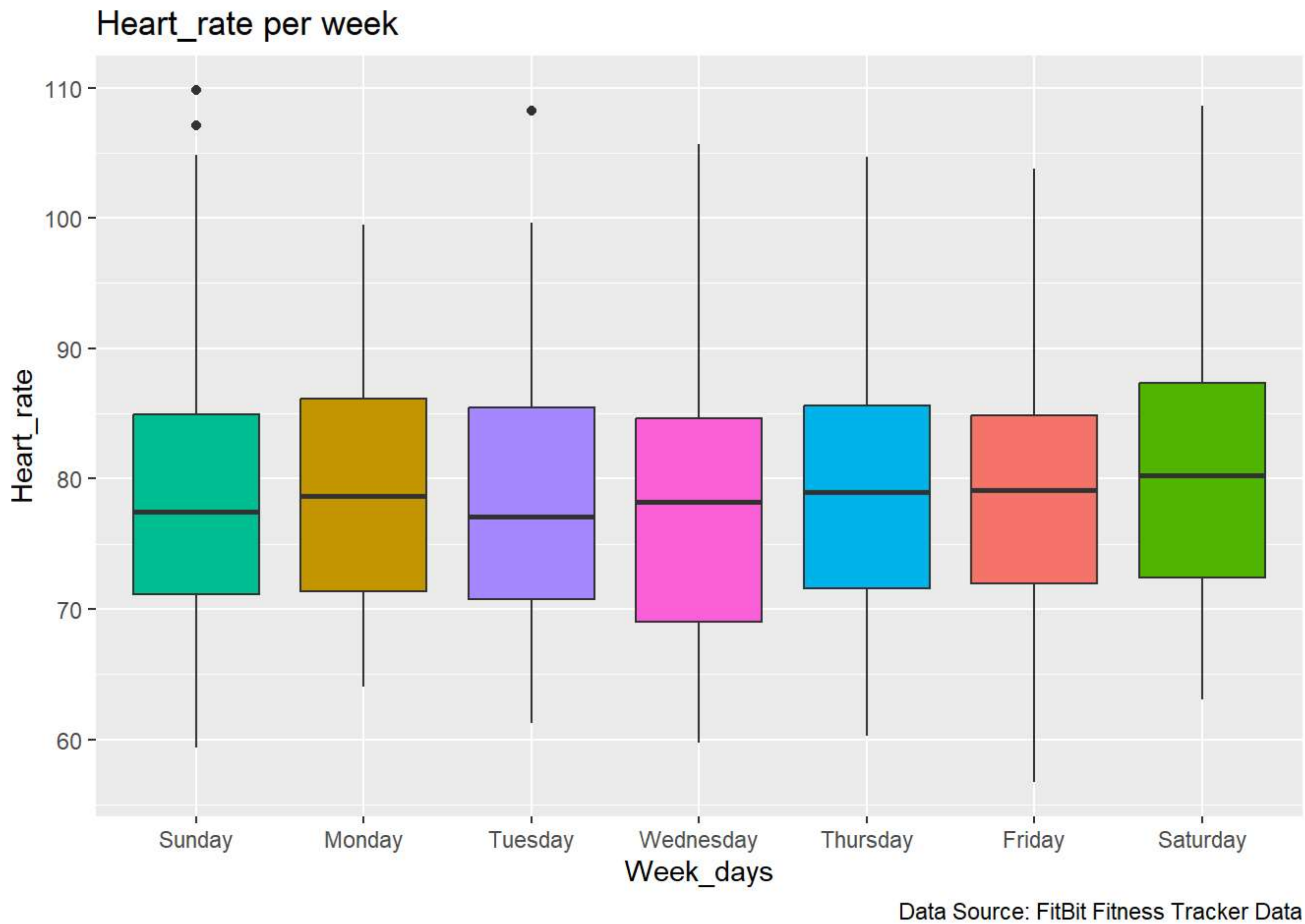
```
heart_rate_daily %>%
  group_by(months) %>%
  summarise(avg_heart_rate = mean(heart_rate)) %>%
ggplot(aes(x = months, y = avg_heart_rate)) +
  geom_col(fill = "orange") +
  scale_x_discrete(limits = c("Mar","Apr","May"))+
  geom_text(aes(label = round(avg_heart_rate),vjust = 1.5)) +
  labs(title = 'Average heart beat per month',
       x = 'Months',
       y = 'Avg_heartbeat',
       caption = 'Data Source: FitBit Fitness Tracker Data')
```



- A normal resting heart rate for adults ranges from 60 to 100 beats per minute.

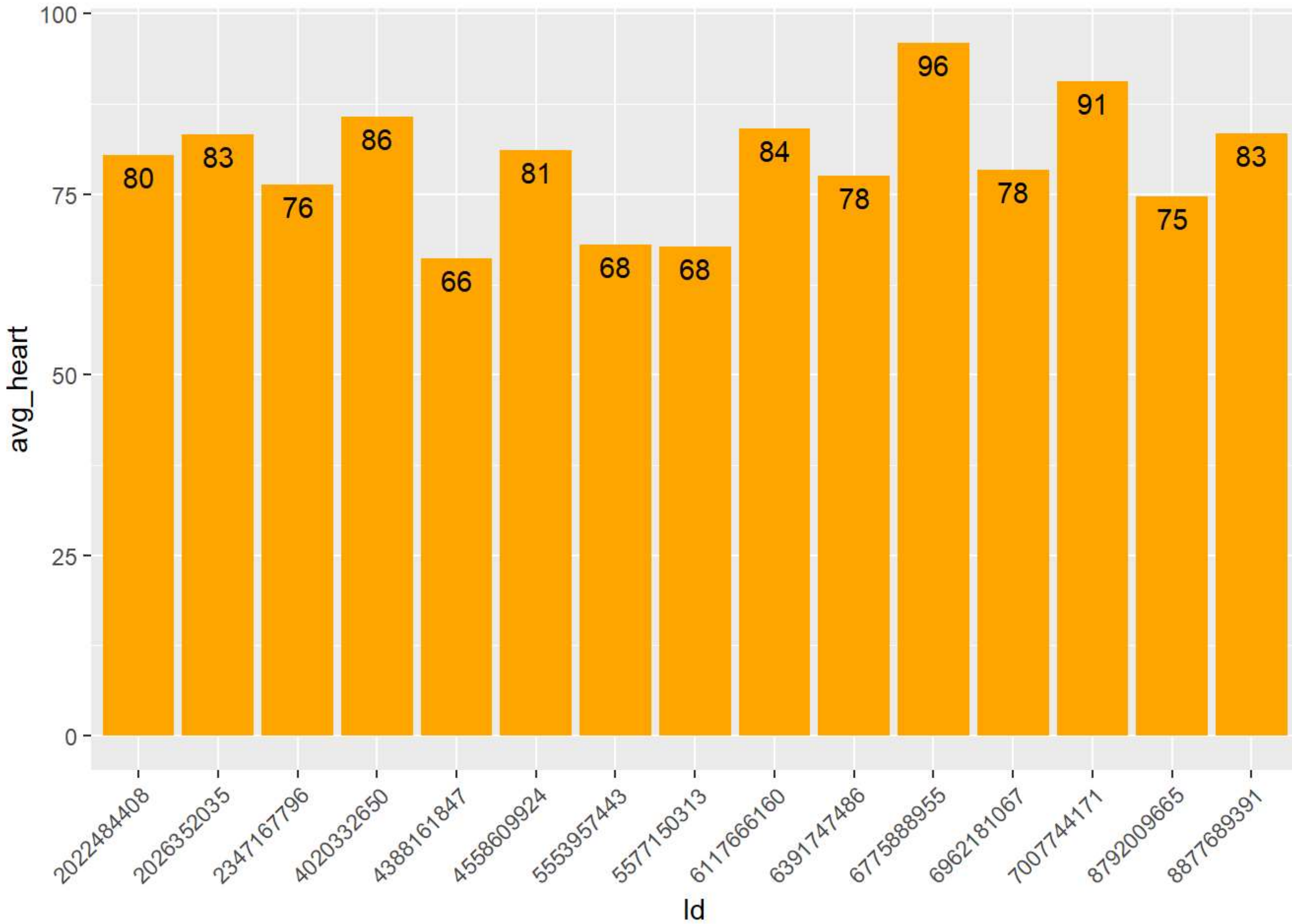
4.12 Average heart_rate per week

```
heart_rate_daily %>%
  ggplot(aes(weekday, heart_rate, fill = weekday)) +
  geom_boxplot() +
  scale_x_discrete(limits = c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Satur
day"))+
  labs(title = 'Heart_rate per week',
       x = 'Week_days',
       y = 'Heart_rate',
       caption = 'Data Source: FitBit Fitness Tracker Data') + theme(legend.position = "none")
```



4.13 Average heart_rate per user

```
heart_rate_daily %>%
 select(Id, heart_rate) %>%
 group_by(Id) %>%
 summarise(avg_heart = mean(heart_rate)) %>%
 drop_na() %>%
 mutate(Id = factor(Id)) %>%
 ggplot(aes(x = Id, y = avg_heart)) +
 geom_col(fill = "orange") +
 geom_text(aes(label = round(avg_heart),vjust = 1.5)) +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We can see customers with avg_heart_rate of 66 and 96 which is normal.

5 Share

Respondents are mostly using the device for tracking steps, heart-beat, sleep , weight info ,calories burnt

1. suggest customer to have regular activity of walking by giving personalized messages as they have forgot to walk on a particular day.
2. Customized messages of calories intake, calories burnt, weight info, sleep , heart-rate, steps taken on particular day.
3. Inform customers with unusual heart beat for check-up
4. Give suggestion to person with more weight on food habits and activity best suited to reduce calories.
5. Customized messages of how to sleep better who have sleep <5hrs