bellabeats_project_final

Archit

06/07/2021

BellaBeat is a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market.

Role: hypothetical junior data analyst. Stakeholders: marketing team, executive team and founders:- Urska Sersen, Sando Mur. Business task: to clean, analyse and visualize data and gain insights and give recommendations to stakeholders using visualizations to support findings Data sourse: Kaggle

Data Set name: FitBit Fitness Tracker Data Dataset Info: The data set has 18 csv files. The dataset was created by Mobius. Limitations: The data does not include demographic information such has gender, age, nationality etc The sample size are small as it only captured 30 users information. The dataset was outdated as the survey was conducted in 2016. The dataset was not first hand data. The data collection period was short as the duration was only 1 month. Too many missing value.

Starting By Importing Data

```
library(tidyverse)
```

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

## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("readxl")
library(readxl)
dailyActivity.33 <-read_excel("C:\\Users\\archi\\OneDrive\\Desktop\\New folder\\dailyActivity.xlsx")
calory_burn_per_day <- read_excel("C:\\Users\\archi\\OneDrive\\Desktop\\New folder\\calory_burn_per_day
Intensity_hour <- read_excel("C:\\Users\\archi\\OneDrive\\Desktop\\New folder\\Intensity_hour.xlsx")
sleep_day.24 <- read_excel("C:\\Users\\archi\\OneDrive\\Desktop\\New folder\\sleep_day.xlsx")
steps_walked_per_day.33 <- read_excel("C:\\Users\\archi\\OneDrive\\Desktop\\New folder\\steps_walked_pe
steps_walked_per_day.33
```

```
## # A tibble: 940 x 3
##            Id ActivityDay StepTotal
##         <dbl> <chr>           <dbl>
##  1 1503960366 42708           13162
##  2 1503960366 4/13/2016       10735
##  3 1503960366 4/14/2016       10460
##  4 1503960366 4/15/2016        9762
##  5 1503960366 4/16/2016       12669
##  6 1503960366 4/17/2016        9705
##  7 1503960366 4/18/2016       13019
##  8 1503960366 4/19/2016       15506
##  9 1503960366 4/20/2016       10544
## 10 1503960366 4/21/2016        9819
## # ... with 930 more rows
```

now the required data has been imported,

#checking if the data has N/A values,outliers and is consistent or not.

```
# checking for missing values
steps_walked_per_day.33 %>% count(Id)
```

```
## # A tibble: 33 x 2
##            Id     n
##         <dbl> <int>
##  1 1503960366    31
##  2 1624580081    31
##  3 1644430081    30
##  4 1844505072    31
##  5 1927972279    31
##  6 2022484408    31
##  7 2026352035    31
##  8 2320127002    31
##  9 2347167796    18
## 10 2873212765    31
## # ... with 23 more rows
```

```
sleep_day.24 %>% count(Id)
```

```
## # A tibble: 24 x 2
##            Id     n
##         <dbl> <int>
##  1 1503960366    25
##  2 1644430081     4
##  3 1844505072     3
##  4 1927972279     5
##  5 2026352035    28
##  6 2320127002     1
##  7 2347167796    15
##  8 3977333714    28
##  9 4020332650     8
## 10 4319703577    26
## # ... with 14 more rows
```

```
Intensity_hour %>% count(Id)
```

```
## # A tibble: 33 x 2
##            Id     n
##         <dbl> <int>
##  1 1503960366   717
##  2 1624580081   736
##  3 1644430081   708
##  4 1844505072   731
##  5 1927972279   736
##  6 2022484408   736
##  7 2026352035   736
##  8 2320127002   735
##  9 2347167796   414
## 10 2873212765   736
## # ... with 23 more rows
```

```
dailyActivity.33 %>% count(Id)
```

```
## # A tibble: 33 x 2
##            Id     n
##         <dbl> <int>
##  1 1503960366    31
##  2 1624580081    31
##  3 1644430081    30
##  4 1844505072    31
##  5 1927972279    31
##  6 2022484408    31
##  7 2026352035    31
##  8 2320127002    31
##  9 2347167796    18
## 10 2873212765    31
## # ... with 23 more rows
```

```
calory_burn_per_day %>% count(Id)
```

```
## # A tibble: 33 x 2
##            Id     n
##         <dbl> <int>
##  1 1503960366    31
##  2 1624580081    31
##  3 1644430081    30
##  4 1844505072    31
##  5 1927972279    31
##  6 2022484408    31
##  7 2026352035    31
##  8 2320127002    31
##  9 2347167796    18
## 10 2873212765    31
## # ... with 23 more rows
```

on noticing that a few Ids have insufficient data, i decide to remove those Ids from the data before analysing it # Removing misleading and incomplete data

```
# removing ids with insufficientd data
step_clean1 <- subset(steps_walked_per_day.33, Id!=2347167796 & Id!=3372868164 & Id!=4057192912 & Id!=82
Intensity_clean1 <- subset(Intensity_hour, Id!=2347167796 & Id!=3372868164 & Id!=4057192912 & Id!=825324
activity_clean1 <- subset(dailyActivity.33, Id!=2347167796 & Id!=3372868164 & Id!=4057192912 & Id!=82532
calory_clean1 <- subset(calory_burn_per_day, Id!=2347167796 & Id!=3372868164 & Id!=4057192912 & Id!=8253
```

Transforming data into usable form for analysis

```
# transforming and summarizing data for analysis
step_clean2 <- (step_clean1 %>%
                  group_by(Id) %>%
                  summarise(step_per_day_avr= mean(StepTotal)))
sleep_clean1 <- (sleep_day.24 %>%
                   group_by(Id) %>%
                   summarise(sleep_avr = mean(TotalMinutesAsleep)))
calory_clean2 <- (calory_clean1 %>%
                    group_by(Id) %>%
                    summarise(calory_avr = mean(Calories)))
#making the date time to correct format
Intensity_clean1$ActivityHour=as.POSIXct(Intensity_clean1$ActivityHour, format="%m/%d/%Y %I:%M:%S %p",
# divide the column to 2 separate columns
Intensity_clean1$time <- format(Intensity_clean1$ActivityHour, format = "%H:%M:%S")
Intensity_clean1$date <- format(Intensity_clean1$ActivityHour, format = "%m/%d/%y")
Intensity_clean2 <- (Intensity_clean1 %>%
                       group_by(time) %>%
                       summarise(Inten_avr = mean(TotalIntensity)))
```

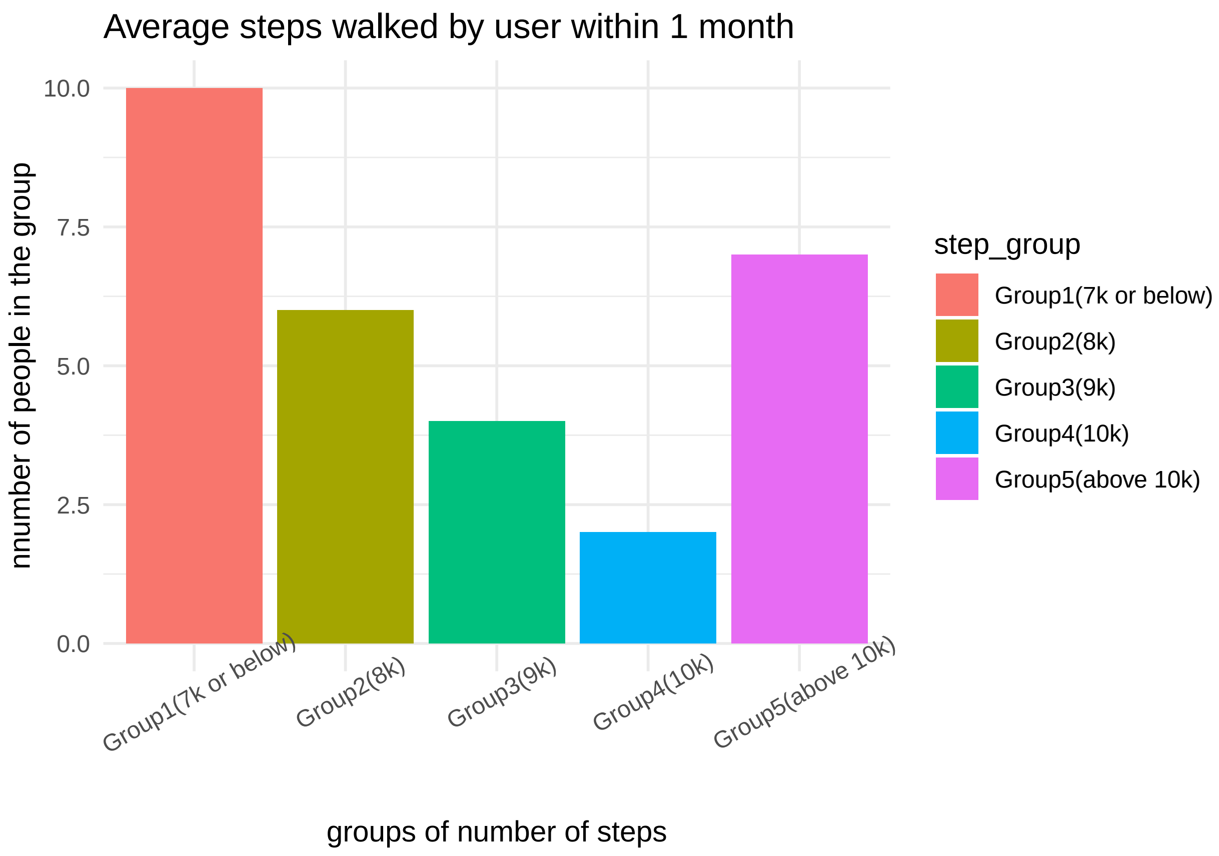
To analyse WALKING PATERNS in detail

grouping ids by number of steps walked

```
# finding out who completes atleast 10k steps by mutating a column
step_clean3 <- (step_clean2 %>%
                  mutate(step_group = case_when(step_per_day_avr<=7000 ~ "Group1(7k or below)",
                                                step_per_day_avr>7000 & step_per_day_avr<=8000 ~ "Group2
                                                step_per_day_avr>8000 & step_per_day_avr<=9000 ~  "Grou
                                                step_per_day_avr>9000 & step_per_day_avr<=10000 ~ "Grou
                                                step_per_day_avr>10000 ~ "Group5(above 10k)")))
step_clean4 <- (step_clean3 %>%
                  group_by(step_group)%>%
                  summarise(count = n()))
```

visualising the data using a bar-graph :

```
ggplot(data=step_clean4, aes(x=step_group, y=count, fill=step_group)) +
  geom_bar(stat="identity")+
  theme_minimal()  +
  ggtitle("Average steps walked by user within 1 month") +
  theme(axis.text.x = element_text(angle=30))+
  labs(x="groups of number of steps", y="nnumber of people in the group")
```



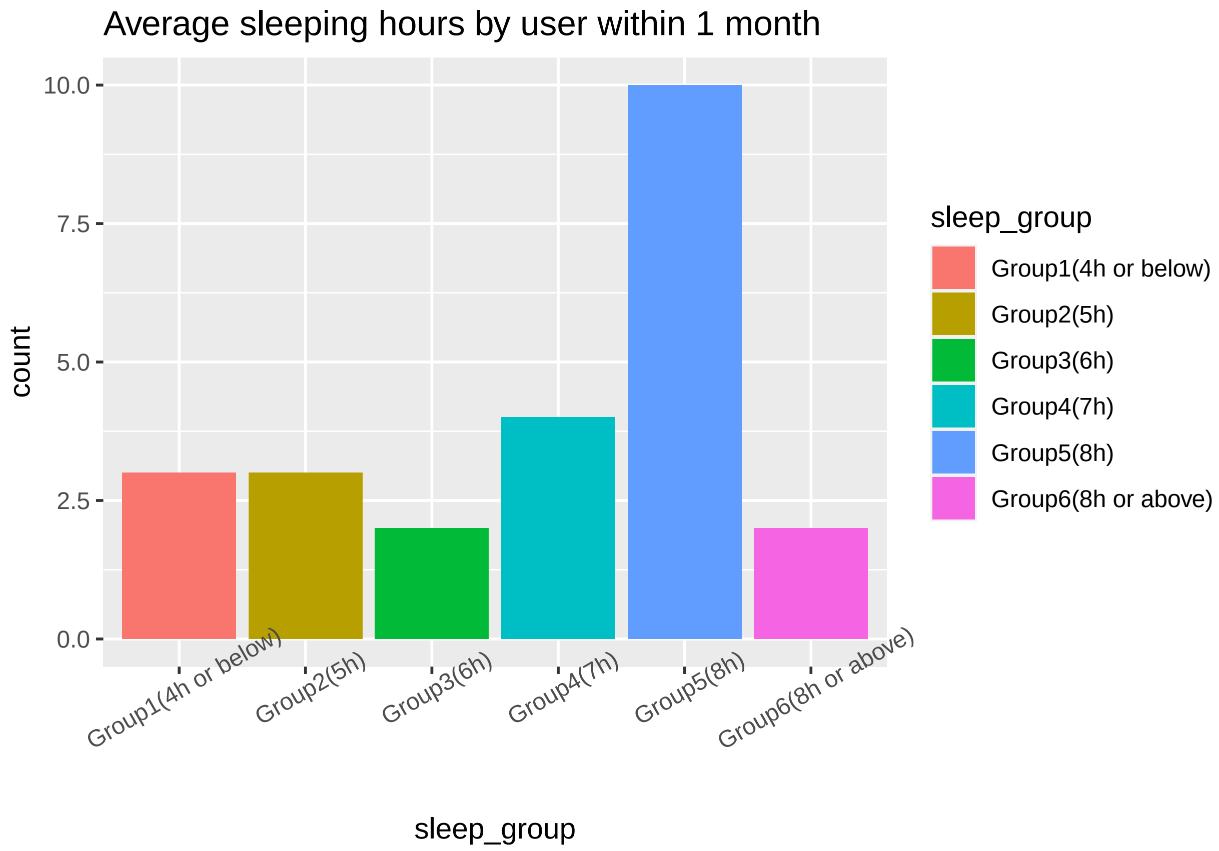
According to Fitbit experts the minimum number of steps that a person should walk is 10k. By observing the above graph we can infere that a significant number of people fall in the "less than 7k" category. the lowest observation is in the 9-10k steps category.

Now analysing sleeping patterns

```
sleep_clean2 <- sleep_clean1 %>% mutate(sleep_hour= sleep_avr/60)
sleep_clean3 <- (sleep_clean2 %>%
                   mutate(sleep_group = case_when(sleep_hour<=4 ~ "Group1(4h or below)",
                                                  sleep_hour>4 & sleep_hour<=5 ~ "Group2(5h)",
                                                  sleep_hour>5 & sleep_hour<=6 ~  "Group3(6h)",
                                                  sleep_hour>6 & sleep_hour<=7 ~ "Group4(7h)",
                                                  sleep_hour>7 & sleep_hour<=8 ~ "Group5(8h)",
                                                  sleep_hour>8 ~ "Group6(8h or above)")))
sleep_clean4 <- (sleep_clean3 %>%
                   group_by(sleep_group)%>%
                   summarise(count = n()))
```

visualising the above data to gain insights:

```
ggplot(data= sleep_clean4 , aes(x= sleep_group , y=  count, fill = sleep_group))+geom_bar(stat= "identi
  theme(axis.text.x = element_text(angle = 30))+ ggtitle("Average sleeping hours by user within 1 month
```



```
less_sleep <- nrow(subset(sleep_clean2, sleep_hour < "7"))
less_sleep
```

```
## [1] 13
```

```
total_ids <- nrow(sleep_clean2)
total_ids
```

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

```
sleep_deficient_per <- (less_sleep/total_ids)*100
sleep_deficient_per
```

```
## [1] 54.16667
```

experts say that 7-8 hours of sleep is necessary for an individual to remain healthy but 54.167% of individuals of our sample sleep for less that 7 hours.

To analyse the time taken to fall asleep

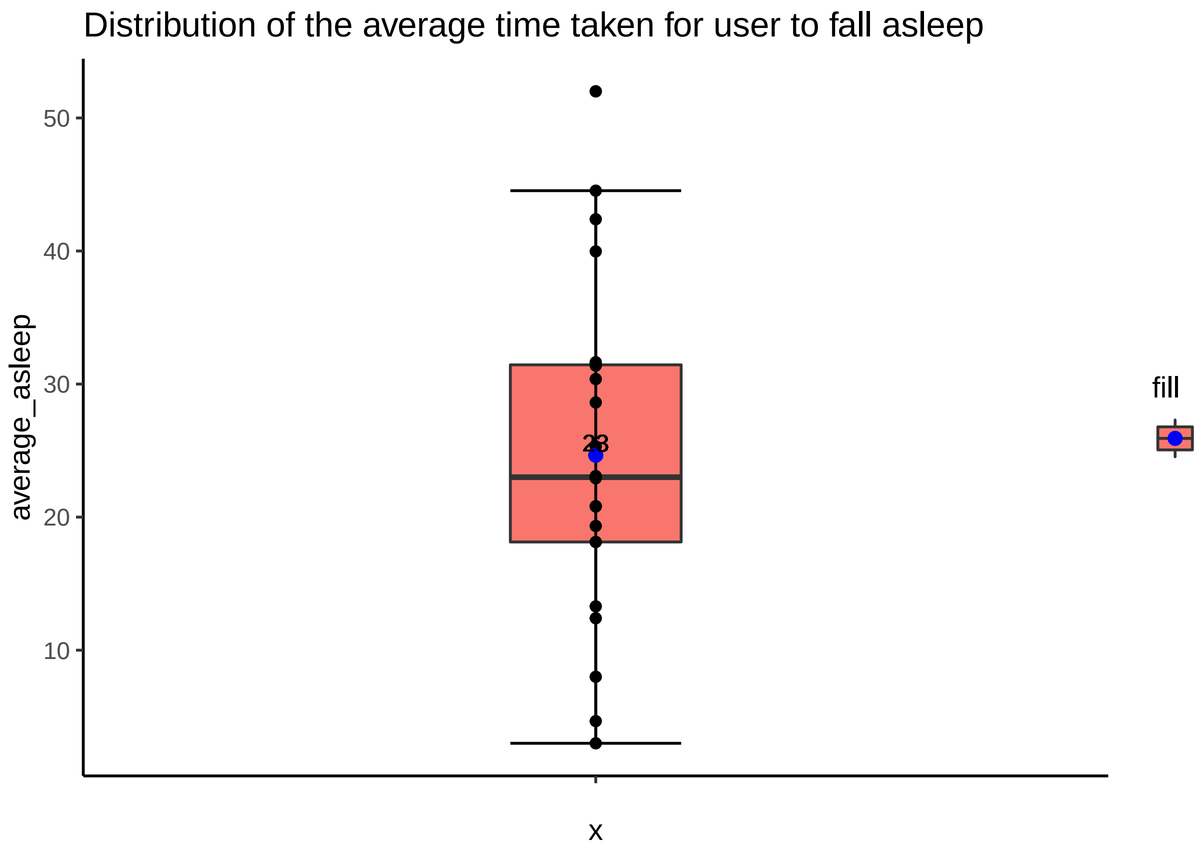
```
fall_asleep1 <- sleep_day.24 %>% mutate(asleep_min = TotalTimeInBed-TotalMinutesAsleep)
fall_asleep2 <- (fall_asleep1 %>%
                   group_by(Id) %>%
                   summarise(average_asleep = mean(asleep_min)))
View(fall_asleep2)
#to remove outliers
fall_asleep3 <- subset(fall_asleep2, Id!=1844505072 & Id!=3977333714)
```

visualise data using a boxplot for better understandings:

```
med = round(median(fall_asleep3$average_asleep),0)
mn= mean(fall_asleep3$average_asleep)
mn
```

```
## [1] 24.64828
```

```
ggplot(data = fall_asleep3, aes( x="",y=average_asleep, fill="")) +
  geom_boxplot(width=0.2) +geom_point()+
  labs(title = "Distribution of the average time taken for user to fall asleep")+
  theme_classic() +stat_boxplot(geom = "errorbar", width = 0.20)+
  stat_summary(fun=mean, geom = "point", colour="blue", size=2)+
  geom_text(data = fall_asleep3, aes(x = "", y = med, label =med),
            size = 3, vjust = -1.5)
```

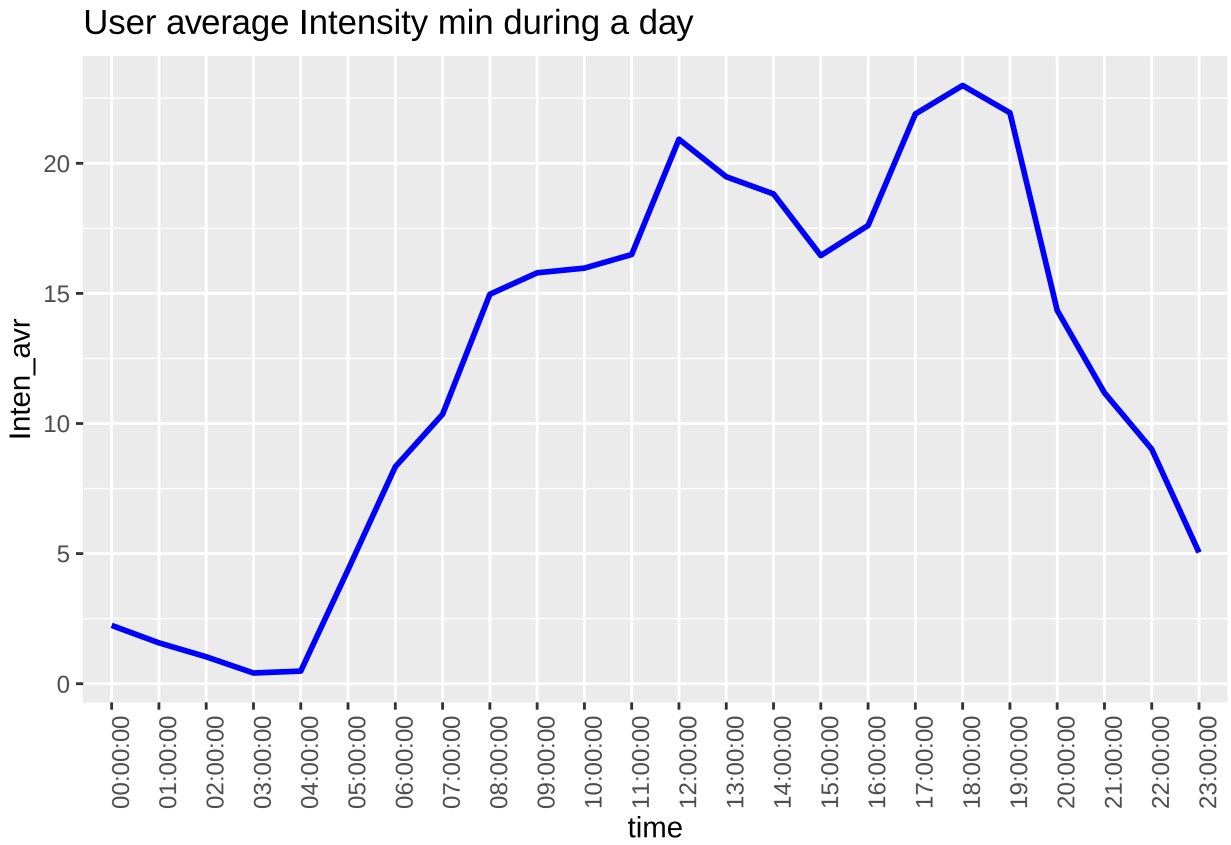


on average it takes 23-24 minutes for an individual to all asleep after in bed.

To analyse intensity throughout the day

```
ggplot(data= na.omit(Intensity_clean2), aes(x=time, y=Inten_avr, group="", na.rm= TRUE)) +
  geom_line(colour="blue", size=1)+

  theme(axis.text.x = element_text(angle = 90))+
  ggtitle("User average Intensity min during a day")
```



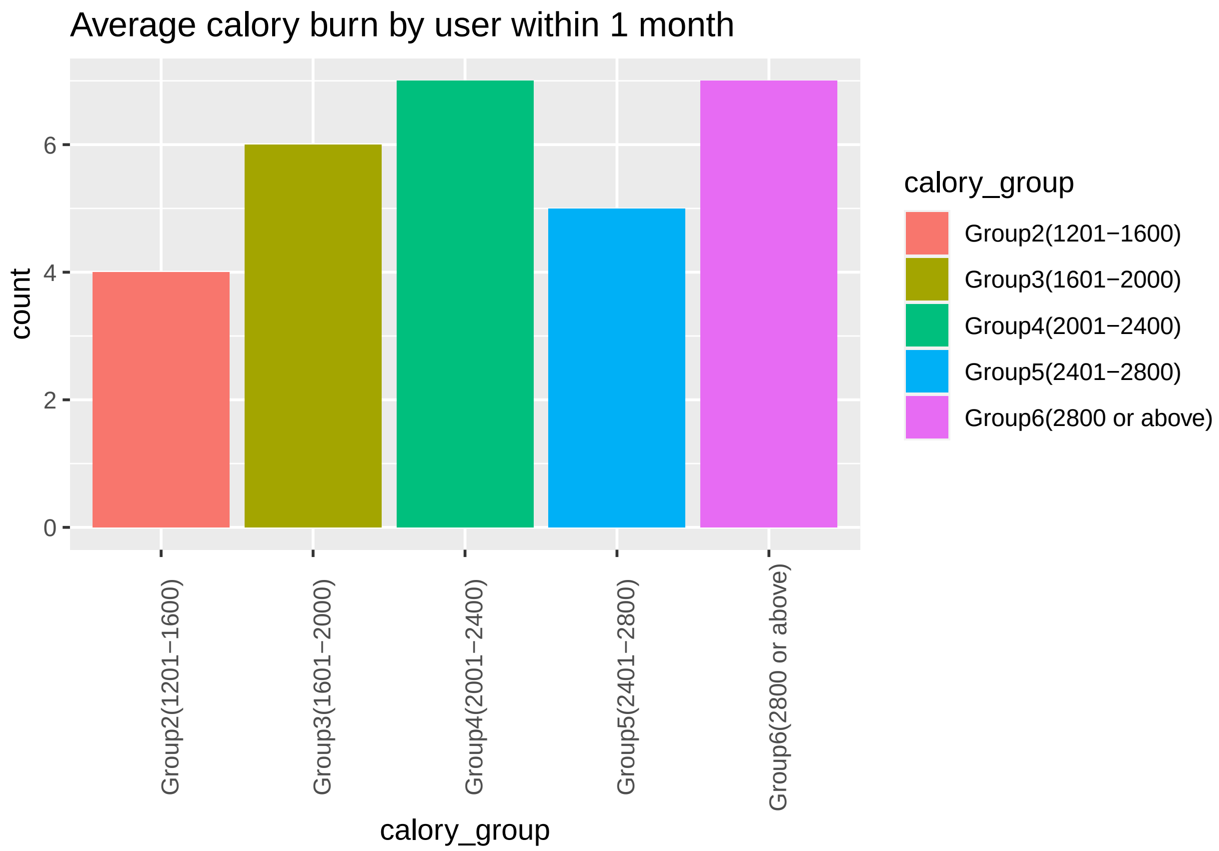
an average person starts their day's activity after 6:00 AM. The intensity varies throughout the day, for example, it decreases a bit after lunch and after 8:00 PM an average individual starts to rest their day.

Analysing the Calorie Inatake

```
calory_clean3 <- (calory_clean2 %>%
                    mutate(calory_group = case_when(calory_avr<=1200 ~ "Group1(1200 or below)",
                                                    calory_avr>1200 & calory_avr<=1600 ~ "Group2(1201-1
                                                    calory_avr>1600 & calory_avr<=2000 ~  "Group3(1601-2
                                                    calory_avr>2000 & calory_avr<=2400 ~ "Group4(2001-24
                                                    calory_avr>2400 & calory_avr<=2800 ~ "Group5(2401-28
                                                             calory_avr>2800 ~ "Group6(2800 or above)")))
calory_clean4 <- (calory_clean3 %>%
                    group_by(calory_group)%>%
                    summarise(count = n()))
head(calory_clean4)
```

```
## # A tibble: 5 x 2
##   calory_group          count
##   <chr>                 <int>
## 1 Group2(1201-1600)         4
## 2 Group3(1601-2000)         6
## 3 Group4(2001-2400)         7
## 4 Group5(2401-2800)         5
## 5 Group6(2800 or above)     7
```

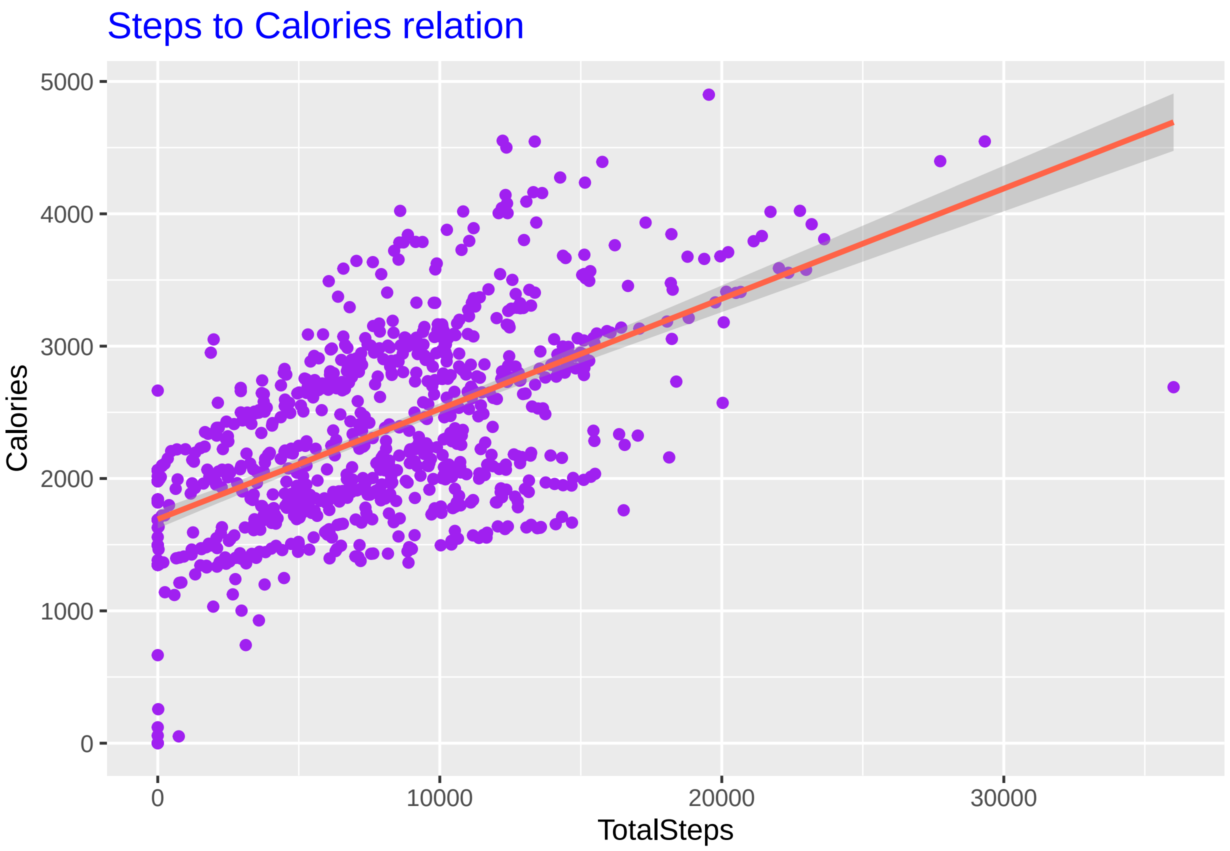
```
ggplot(data = calory_clean4, aes(x=calory_group, y=count, fill=calory_group))+
  geom_bar(stat= "identity")+ theme(axis.text.x= element_text(angle= 0))+
  theme(axis.text.x= element_text(angle=90))+
labs(title= "Average calory burn by user within 1 month")
```



Relation between total steps and calories burnt

```
ggplot(data= activity_clean1, aes(TotalSteps, Calories)) +
  geom_jitter(color = 'purple') +
  geom_smooth(method = 'lm',color="tomato")+
  ggtitle("Steps to Calories relation")+
  theme(plot.title = element_text(color="blue", size=14))
```

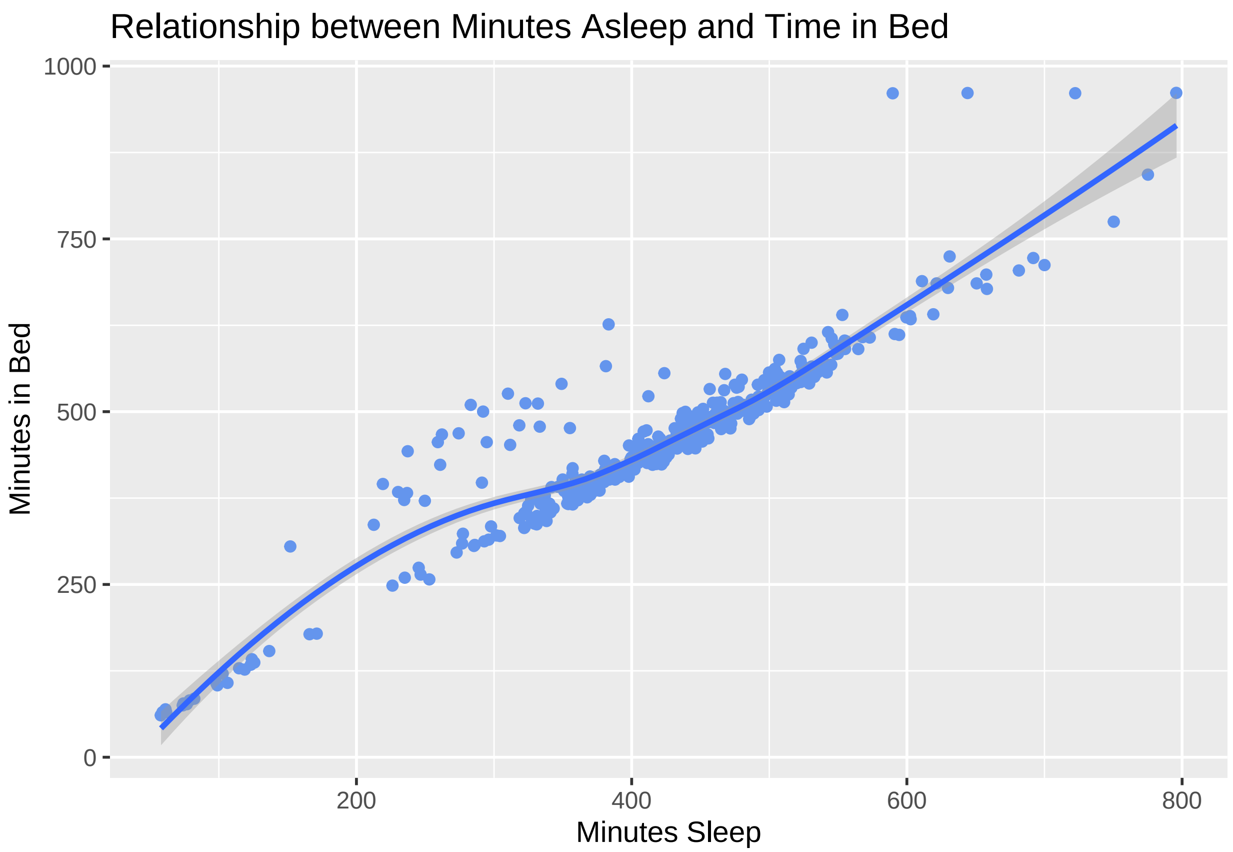
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Relation between time in bed and sleep

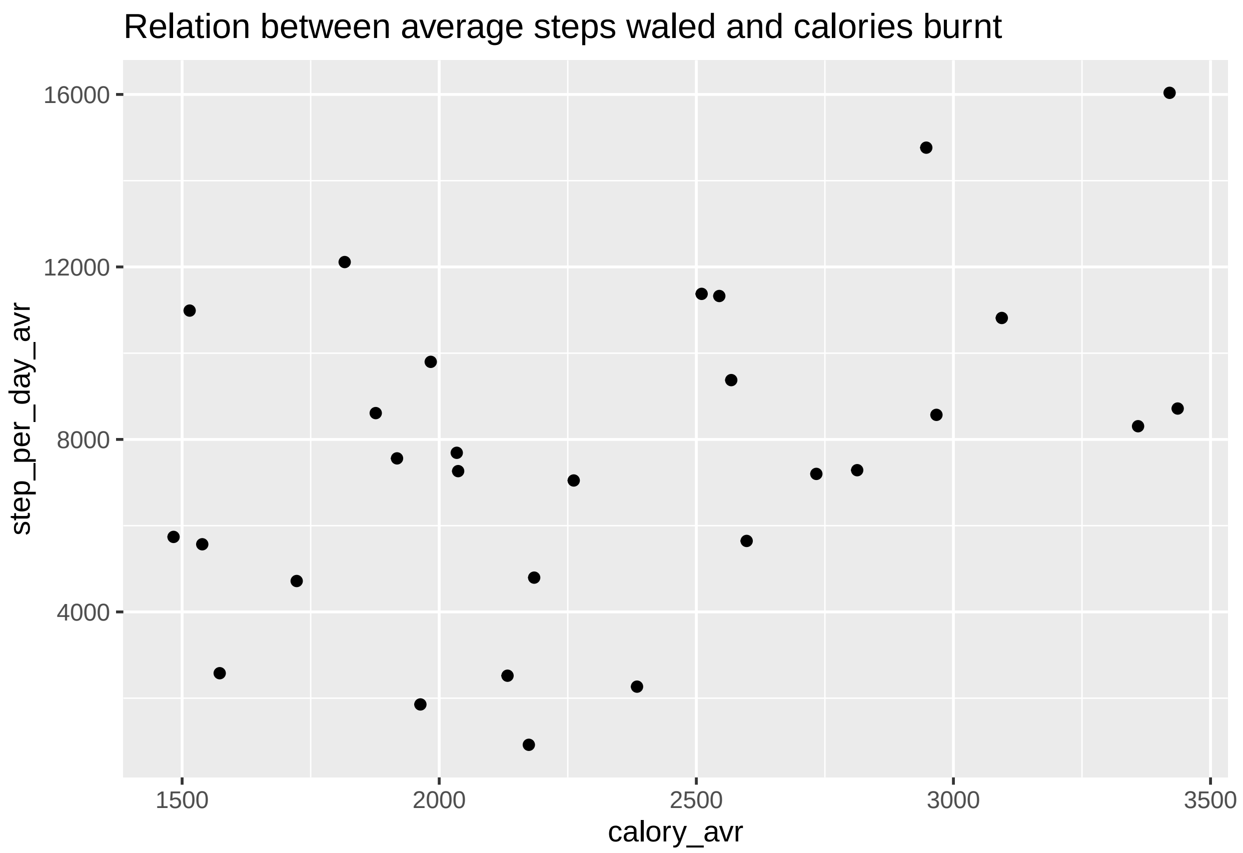
```
ggplot(data= na.omit(sleep_day.24), aes(x=TotalMinutesAsleep, y= TotalTimeInBed))+
  geom_jitter(color = "cornflowerblue")+
  labs(title = 'Relationship between Minutes Asleep and Time in Bed', x = 'Minutes Sleep', y = 'Minutes
  geom_smooth(method = 'loess')
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

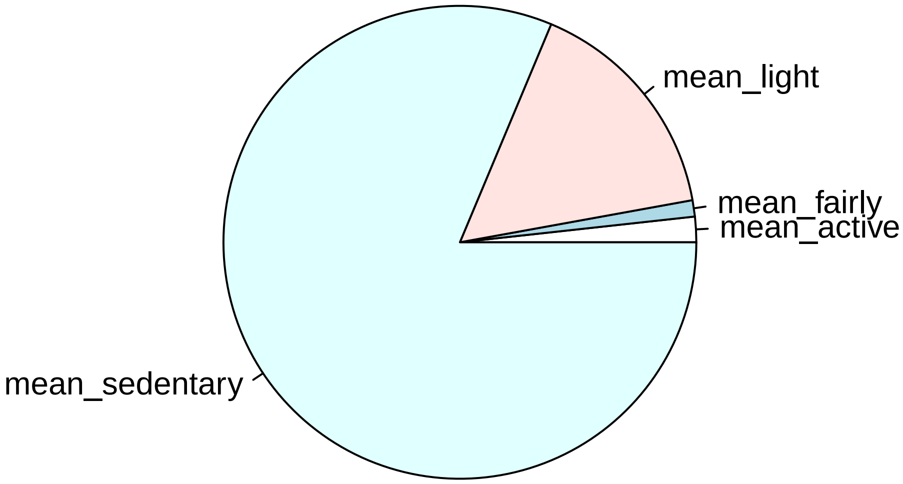


Relation between total steps and calory burnt

```
calory_step <- merge(calory_clean2,step_clean2, by="Id")
View(calory_step)
ggplot(data= calory_step, aes(x=calory_avr, y= step_per_day_avr))+ geom_jitter(color="black")+
  ggtitle("Relation between average steps waled and calories burnt")
```



```
#piechart on type of minutes spent
pie <- (dailyActivity.33 %>%
          summarise(mean_active= mean(VeryActiveMinutes),mean_fairly= mean(FairlyActiveMinutes),
                    mean_light= mean(LightlyActiveMinutes), mean_sedentary = mean(SedentaryMinutes)))
View(pie)
r <- c(21.2 , 13.6 , 193, 991)
col <- c(colnames(pie))
pie(r,col,radius=1)
```



# INSIGHTS AND RECOMMENDATIONS

Insight1:

By looking at the missing value from the dataset, we can see that user did not use the device during sleeping. ## Recommendation1: Bellabeat can smaller the size the device(leaf) which user can put it 24/7 as part the body like a necklace.

Insight2:

By looking at the walking steps average, majority of the user have less then 7k walking steps per day. ## Recommendation2: Bellabeat can encourage user to take lunch break walk by notifying them on the app or providing coffee vouchers to encourage user to walk to a fair distance coffee shop which is within the office area.

Insight3:

By looking at the Intensity level distribution within the month, the reading was very stable which means the user did not take any healthy improvment while using FitBit. ## Recommendation3: Bellabeat could take a longer obervation of the servey, for example 3 months time in order to observe activity level changes.

Insight4:

By looking at the calory burn level, we can see that the majority group is close to the average burn level of 2400. ## Recommendation4: Recalling the correlation matrix, inorder to increase calory burn, very active min has the strongest correlation, which means doing cardio activity is more effective than running or walking. Bellabeat can provide cardio video in the app by notifying user to workout at home.