Case Study Bellabeat

Mebin Mathew

11/02/2022

Introduction

Welcome to the Bellabeat data analysis case study! In this case study, i will perform many real-world tasks of a junior data analyst working for Bellabeat, a high-tech manufacturer of health-focused products for women. In order to answer the key business questions, i will follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act.This Capstone project is a part of Google Data Analytics Professional Certificate.

Scenario

I am a junior data analyst working on the marketing analyst team at Bellabeat, 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. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. I have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights I discover will then help guide marketing strategy for the company. I will present my analysis to the Bellabeat executive team along with my high-level recommendations for Bellabeat's marketing strategy.

Characters

Urška Sršen: Bellabeat's cofounder and Chief Creative Officer

Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team

Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy. You joined this team six months ago and have been busy learning about Bellabeat' 's mission and business goals — as well as how you, as a junior data analyst, can help Bellabeat achieve them.

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.

Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.

Time: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.

Spring: This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.

Step 1 - Ask

Ask step involves coming up with S.M.A.R.T Questions to understand the business problem/task,stakeholders expectations and to identify the audience we are presenting our analysis. In this case Sršen asks us 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.

Key Questions

What is the problem you are trying to solve?

Find key trends and insights from non-Bellabeat smart products and how consumers interact with those devices,compare it with one Brellabeat product to analyse the growth potential and opportunities

How can your insights drive business decisions?
Develop a marketing strategy to help the further growth of the company to become a larger player in global smart health device marker

Deliverable

A clear statement of the business task
objective is to find key trend and insights from non- Bellabeat smart health products and how consumers interact with those devices , compare it and develop a marketing strategy for potential growth and opportunities

Step 2 - Prepare
In this Step we decide if we will collect the data using your own resources or receive (and possibly purchase it) from another party or from a third party .check whether the data is unbiased and credible. In this case Sršen encourages you to use public data that explores smart device users' daily habits. She points you to a specific data set:FitBit Fitness Tracker Data-This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits. Sršen tells you that this data set might have some limitations, and encourage to consider adding another data to help address those limitations as you begin to work more with this data.

Key Questions
Where is your data stored?
The data is located in kaggle dataset.
How is the data organized? Is it in long or wide format?
Data is Organised in daily,hourly,minutes , mostly in long format
Are there issues with bias or credibility in this data?
Data set is too small for accurate trend detection and other personal health factors not considered
How are you addressing licensing, privacy, security, and accessibility?
company have there dataset ,no personally identifying information included
How did you verify the data's integrity? All column are consistent and labelled ,all data type is correct
How does it help you answer your question?
Data have insights in enough insights
Are there any problems with the data?
More information is need more accurate trend and insight

Deliverable
A description of all data sources used
This dataset 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. Individual reports can be parsed by export session ID (column A) or timestamp (column B). Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences.

Step 3 - Process
In this step we will focus on ensuring data integrity,Understanding and Cleaning data using R

Key Questions
What tools are you choosing and why?
R programming,r has various tools in cleaning and analyzing our data
Have you ensured your data's integrity?
data integrity is checked using excel
What steps have you taken to ensure that your data is clean?
import,merged,eliminated the duplicate,identify the null values

How can you verify that your data is clean and ready to analyze?
verify using basic analysis and visualization
Have you documented your cleaning process so you can review and share those results?
Documented using R markdown

Code

Exploring the data
In this Study Daily ,hourly and weight logs datas are considered

```
#Loading r packages for cleaning and manipulation of data#

library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
library(lubridate)
```

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

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

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
#Read csv from directory , view data set and check the integrity #

#Import the dailyActivity_merged.csv#
#convert date to mdy and create a new date column #
D_activity <-  read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.cs
str(D_activity)
```

```
## 'data.frame':    939 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "04-12-2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```r
New_date<- c(mdy(D_activity$ActivityDate))
D_activity$New_date <- New_date
D_activity$ActivityDate <- NULL
View(D_activity)
str(D_activity)
```

```
## 'data.frame':    939 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
##  $ New_date                : Date, format: "2016-04-12" "2016-04-13" ...
```

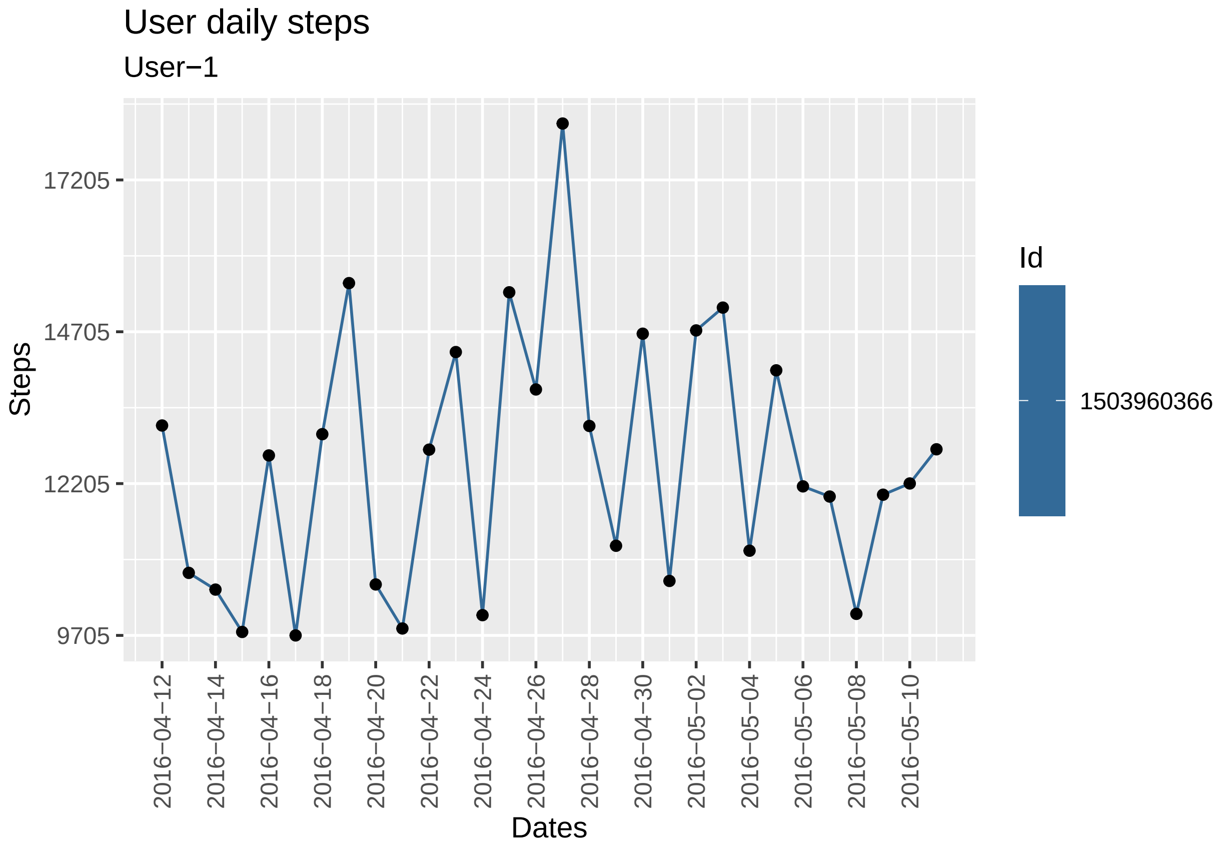
```
#Create a data frame for distinct value of Id for reference and simplify the user id 1 -33 #
distinctid_df<-D_activity%>% distinct(Id)
distinctid_df$Simple_Id <- seq(1, 33, length.out = dim(distinctid_df)[1])
view(distinctid_df)
str(distinctid_df)
```

```
## 'data.frame':    33 obs. of  2 variables:
##  $ Id       : num  1.50e+09 1.62e+09 1.64e+09 1.84e+09 1.93e+09 ...
##  $ Simple_Id: num  1 2 3 4 5 6 7 8 9 10 ...
```

```
#Create a data frame for each user reusable code, extract the corresponding data from D_activity#
user_df<- filter(D_activity,D_activity$Id == distinctid_df[1,1])
view(user_df)
str(user_df)
```

```
## 'data.frame':    30 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
##  $ New_date                : Date, format: "2016-04-12" "2016-04-13" ...
```

```
#load ggplot2 ,create scatter and line plot for user 1 -33 based  on different variable and compare#


 ggplot(data = user_df) + geom_line(mapping = aes(x=New_date,y=TotalSteps,color =Id))+
 geom_point(mapping =aes(x=New_date,y=TotalSteps) )+
 scale_x_continuous(breaks = round(seq(min(user_df$New_date), max(user_df$New_date), by = 2),1)) +
 scale_y_continuous(breaks = round(seq(min(user_df$TotalSteps), max(user_df$TotalSteps), by = 2500),1))
 labs(title ="User daily steps",subtitle = "User-1") + xlab("Dates") + ylab("Steps")+
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
#importing hourlySteps_merged.csv,hourlyCalories_merged.csv,hourlyIntensities_merged.csv#
#Create data frames using csv files and merging it into one data set#
H_step <-  read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/hourlySteps_merged.csv")
H_calories <- read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/hourlyCalories_merged.c
H_instensities <- read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/hourlyIntensities_me
H_merged1 <- merge.data.frame(H_step,H_calories, all.x = TRUE)
H_merged <- merge.data.frame(H_merged1,H_instensities, all.x = TRUE)
str(H_merged)
```

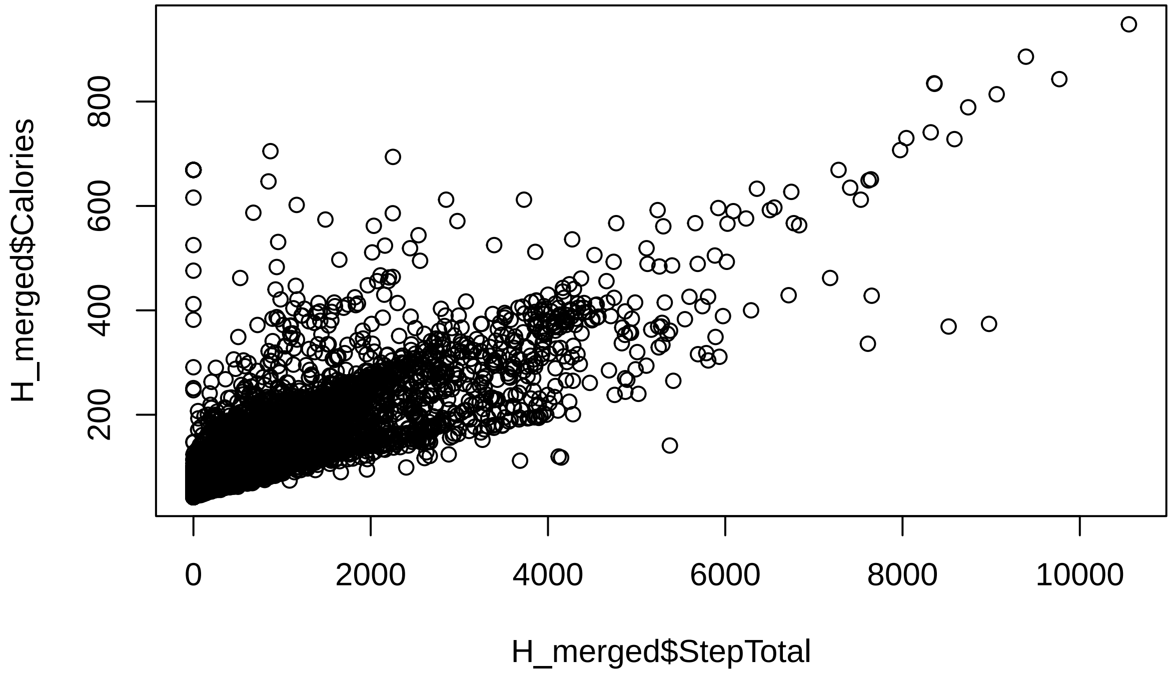
```
## 'data.frame':    22099 obs. of  6 variables:
##  $ Id              : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour    : chr  "4/12/2016 1:00:00 AM" "4/12/2016 1:00:00 PM" "4/12/2016 10:00:00 AM" "4/1
##  $ StepTotal       : int  160 221 676 89 360 338 373 253 151 1166 ...
##  $ Calories        : int  61 66 99 65 76 81 81 73 59 110 ...
##  $ TotalIntensity  : int  8 6 29 9 12 21 20 11 7 36 ...
##  $ AverageIntensity: num  0.133 0.1 0.483 0.15 0.2 ...
```

```
#converting hours using function mdy_hms and create a new column#
New_Hr<- c(mdy_hms(H_merged$ActivityHour))
H_merged$New_Hr <- New_Hr
H_merged$ActivityHour <- NULL
view(H_merged)
str(H_merged)
```

```
## 'data.frame':    22099 obs. of  6 variables:
```

```
##  $ Id              : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ StepTotal       : int  160 221 676 89 360 338 373 253 151 1166 ...
##  $ Calories        : int  61 66 99 65 76 81 81 73 59 110 ...
##  $ TotalIntensity  : int  8 6 29 9 12 21 20 11 7 36 ...
##  $ AverageIntensity: num  0.133 0.1 0.483 0.15 0.2 ...
##  $ New_Hr          : POSIXct, format: "2016-04-12 01:00:00" "2016-04-12 13:00:00" ...
```

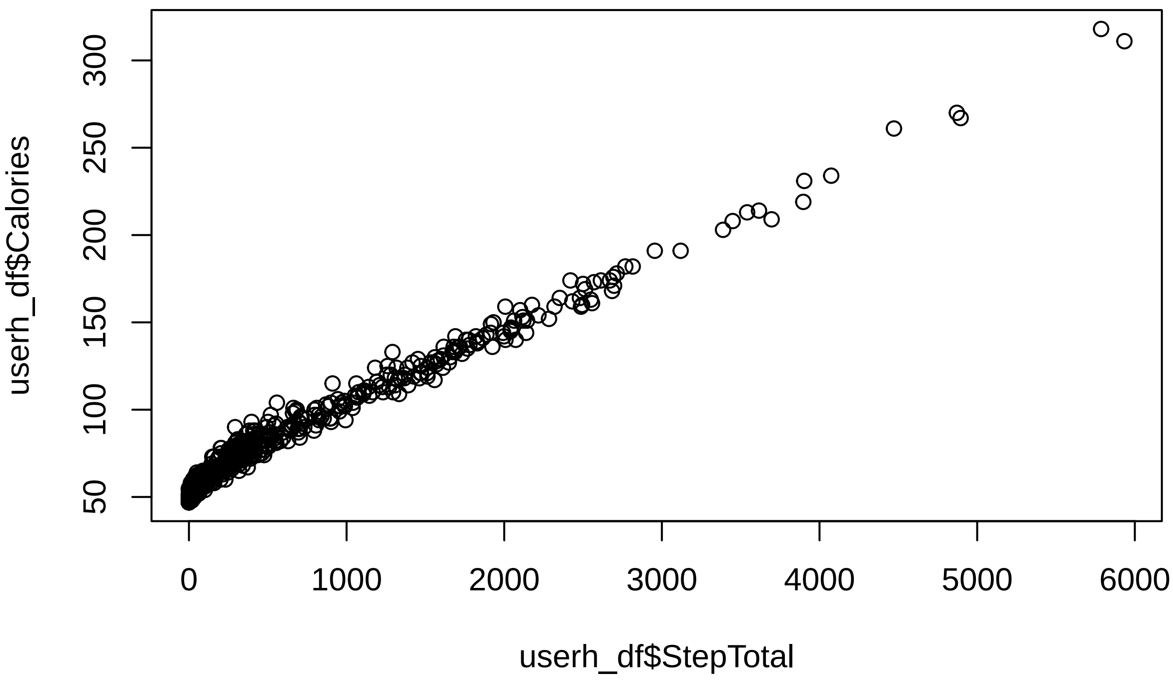
```
#Visualize total step vs calories#
plot(x=H_merged$StepTotal,y=H_merged$Calories)
```



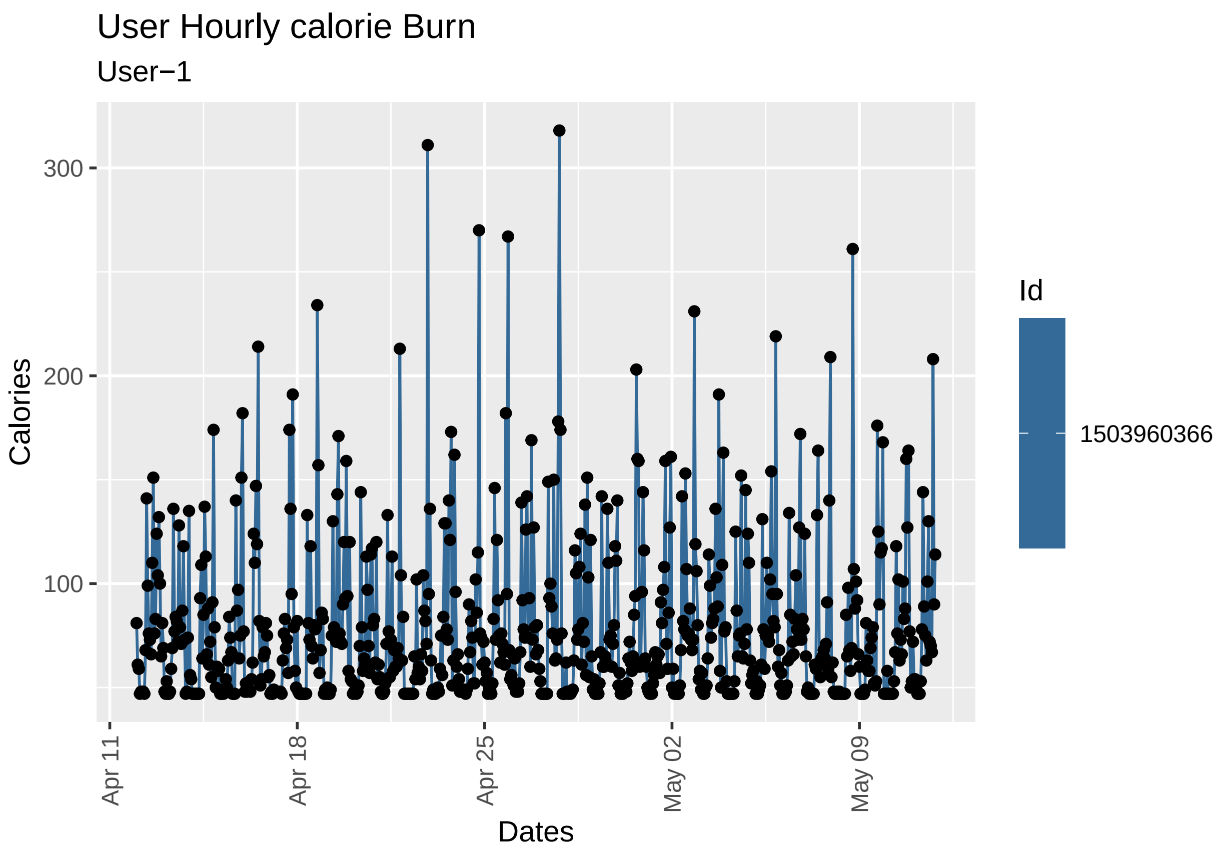
```
#Creating user data frame reusable code#
userh_df<- filter(H_merged,H_merged$Id == distinctid_df[1,1])
view(userh_df)
str(userh_df)
```

```
## 'data.frame':    717 obs. of  6 variables:
##  $ Id              : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ StepTotal       : int  160 221 676 89 360 338 373 253 151 1166 ...
##  $ Calories        : int  61 66 99 65 76 81 81 73 59 110 ...
##  $ TotalIntensity  : int  8 6 29 9 12 21 20 11 7 36 ...
##  $ AverageIntensity: num  0.133 0.1 0.483 0.15 0.2 ...
##  $ New_Hr          : POSIXct, format: "2016-04-12 01:00:00" "2016-04-12 13:00:00" ...
```

```
#Visualize total step vs calories on user 1-33#
plot(x=userh_df$StepTotal,y=userh_df$Calories)
```



```
#Visualize Hours vs calories on user 1-33#
ggplot(data = userh_df) + geom_line(mapping = aes(x=New_Hr,y= Calories,color = Id))+
geom_point(mapping = aes(x=New_Hr,y=Calories))+
labs(title ="User Hourly calorie Burn",subtitle = "User-1") + xlab("Dates") + ylab("Calories")+
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
#importing sleepDay_merged.csv,weightLogInfo_merged.csv#
D_Sleep <-  read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
weightLog <-  read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.cs
view(D_Sleep)
str(D_Sleep)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
View(weightLog)
str(weightLog)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/20
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

```
#converting D_Sleep -date using function mdy_hms and create a new date column#
Newsleep_Day<- c(mdy_hms(D_Sleep$SleepDay))
D_Sleep$Newsleep_Day <- Newsleep_Day
D_Sleep$SleepDay <- NULL
View(D_Sleep)
str(D_Sleep)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
##  $ Newsleep_Day      : POSIXct, format: "2016-04-12" "2016-04-13" ...
```

```
#converting D_Sleep -date using function mdy_hms and create a new date column#
NewDate1<- c(mdy_hms(weightLog$Date))
weightLog$NewDate1 <- NewDate1
weightLog$Date <- NULL
View(weightLog)
str(weightLog)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
##  $ NewDate1      : POSIXct, format: "2016-05-02 23:59:59" "2016-05-03 23:59:59" ...
```

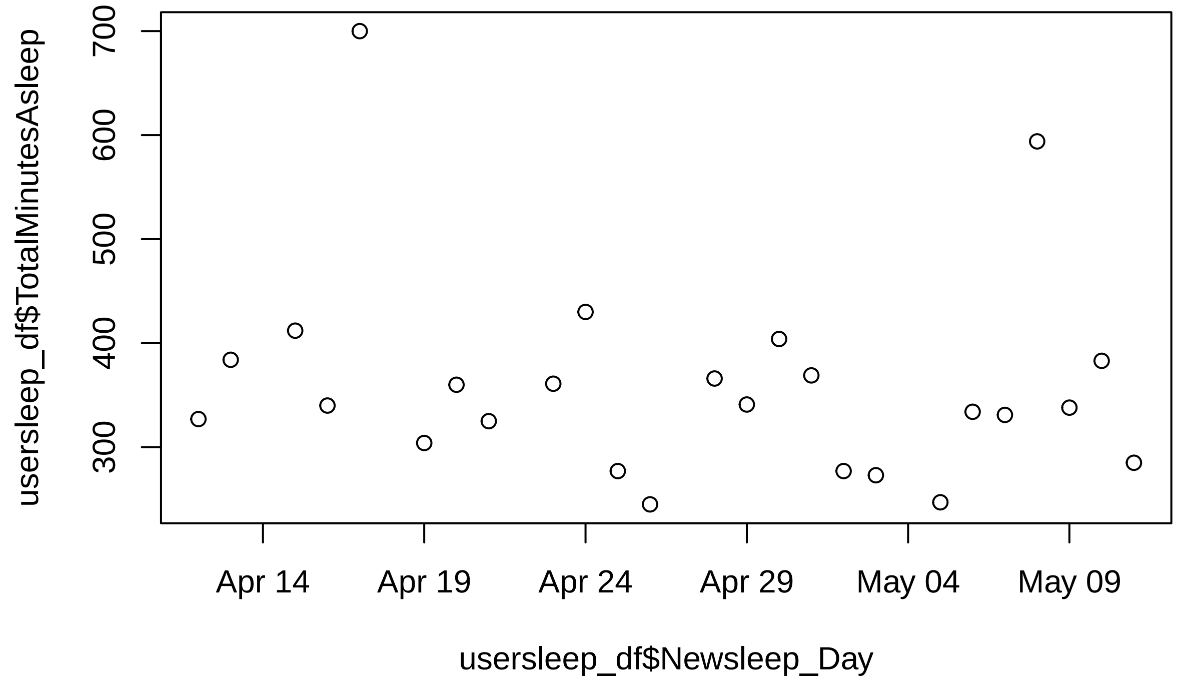
```
#Creating user data frame reusable code#
usersleep_df<- filter(D_Sleep,D_Sleep$Id == distinctid_df[1,1])
view(usersleep_df)
str(usersleep_df)
```

```
## 'data.frame':    25 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
##  $ Newsleep_Day      : POSIXct, format: "2016-04-12" "2016-04-13" ...
```

```
#Creating user data frame reusable code#
userweight_df<- filter(weightLog,weightLog$Id == distinctid_df[1,1])
view(userweight_df)
str(userweight_df)
```

```
## 'data.frame':    2 obs. of  8 variables:
##  $ Id            : num  1.5e+09 1.5e+09
##  $ WeightKg      : num  52.6 52.6
##  $ WeightPounds   : num  116 116
##  $ Fat            : int  22 NA
##  $ BMI            : num  22.6 22.6
##  $ IsManualReport: chr  "True" "True"
##  $ LogId          : num  1.46e+12 1.46e+12
##  $ NewDate1       : POSIXct, format: "2016-05-02 23:59:59" "2016-05-03 23:59:59"
```

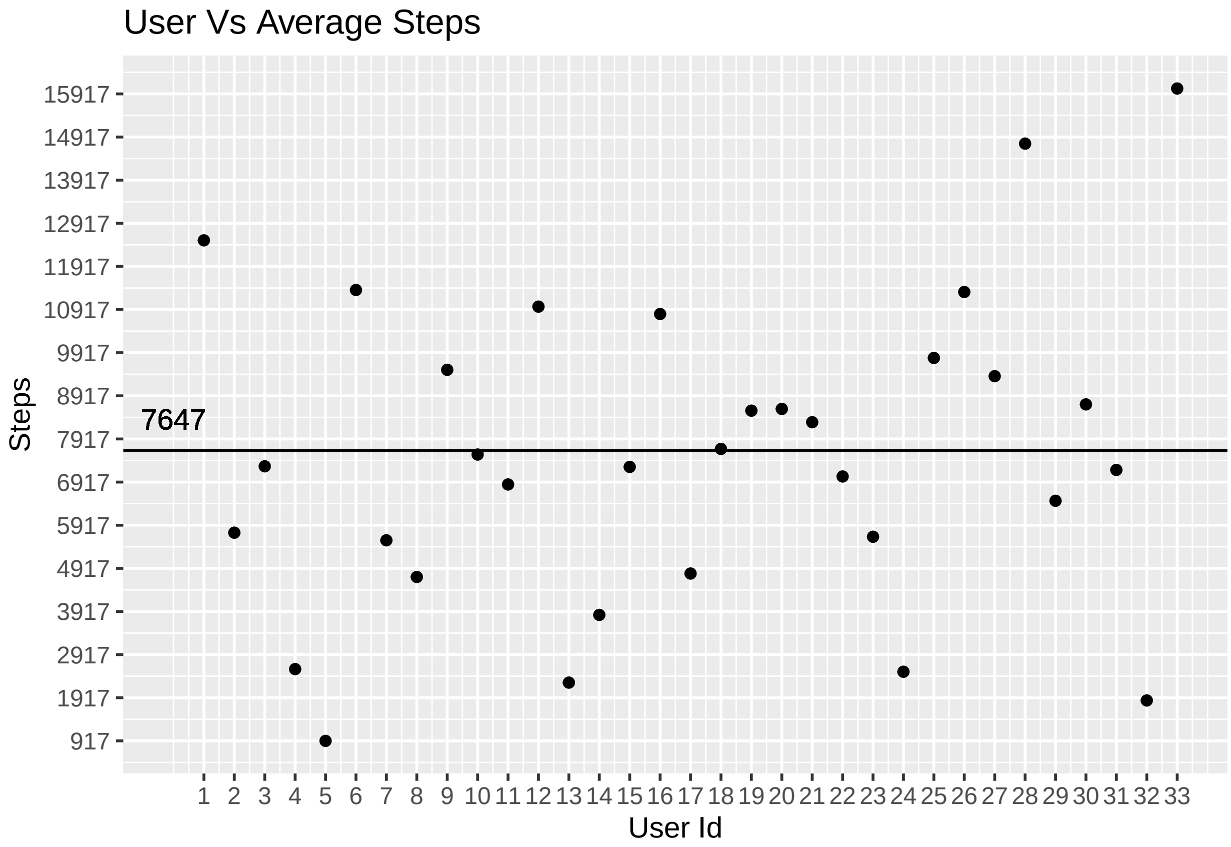
```
#Visualize days vs total minutes asleep  on user 1-33#

plot(x=usersleep_df$Newsleep_Day,y=usersleep_df$TotalMinutesAsleep)
```



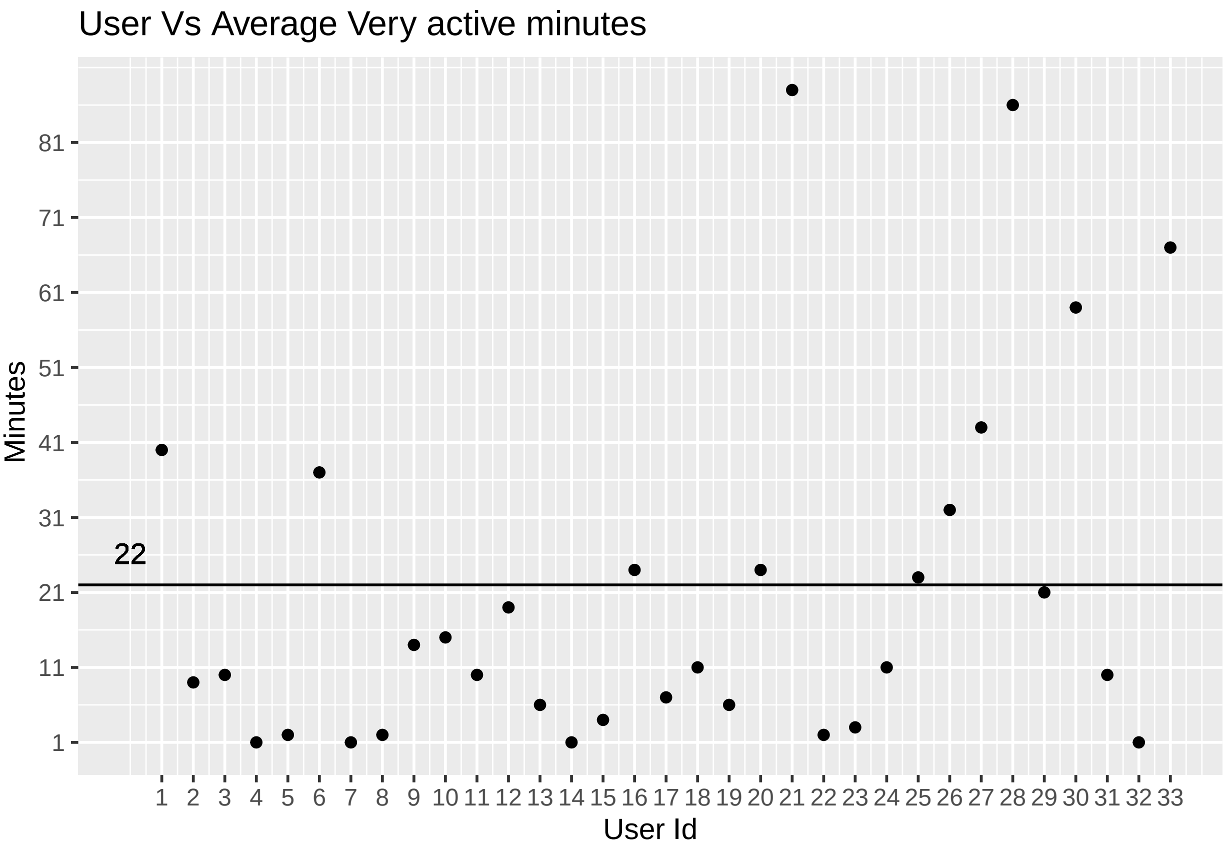
```
## Step 4 -Analyze###
#In this step we will perform calculations and analysis on processed data


#Visualize and summarize the information for analysis
#save data frame final1 as csv and is used in final analysis and visualization
final1 <-  read.csv("~/R/R Raw Data/Case_Study/Fitabase Data 4.12.16-5.12.16/final1.csv")

view(final1)
str(final1)
```

```
## 'data.frame':    33 obs. of  7 variables:
##  $ X                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Id                   : num  1.50e+09 1.62e+09 1.64e+09 1.84e+09 1.93e+09 ...
##  $ Avgstep             : int  12521 5744 7283 2581 917 11371 5567 4717 9520 7556 ...
##  $ AvgVeryactiveminutes: int  40 9 10 1 2 37 1 2 14 15 ...
##  $ AvgSedentaryMinutes : int  829 1258 1162 1207 1318 1113 690 1221 688 1098 ...
##  $ AvgCalories         : int  1877 1484 2812 1574 2173 2510 1541 1725 2044 1917 ...
##  $ Simple_Id           : int  1 2 3 4 5 6 7 8 9 10 ...
```

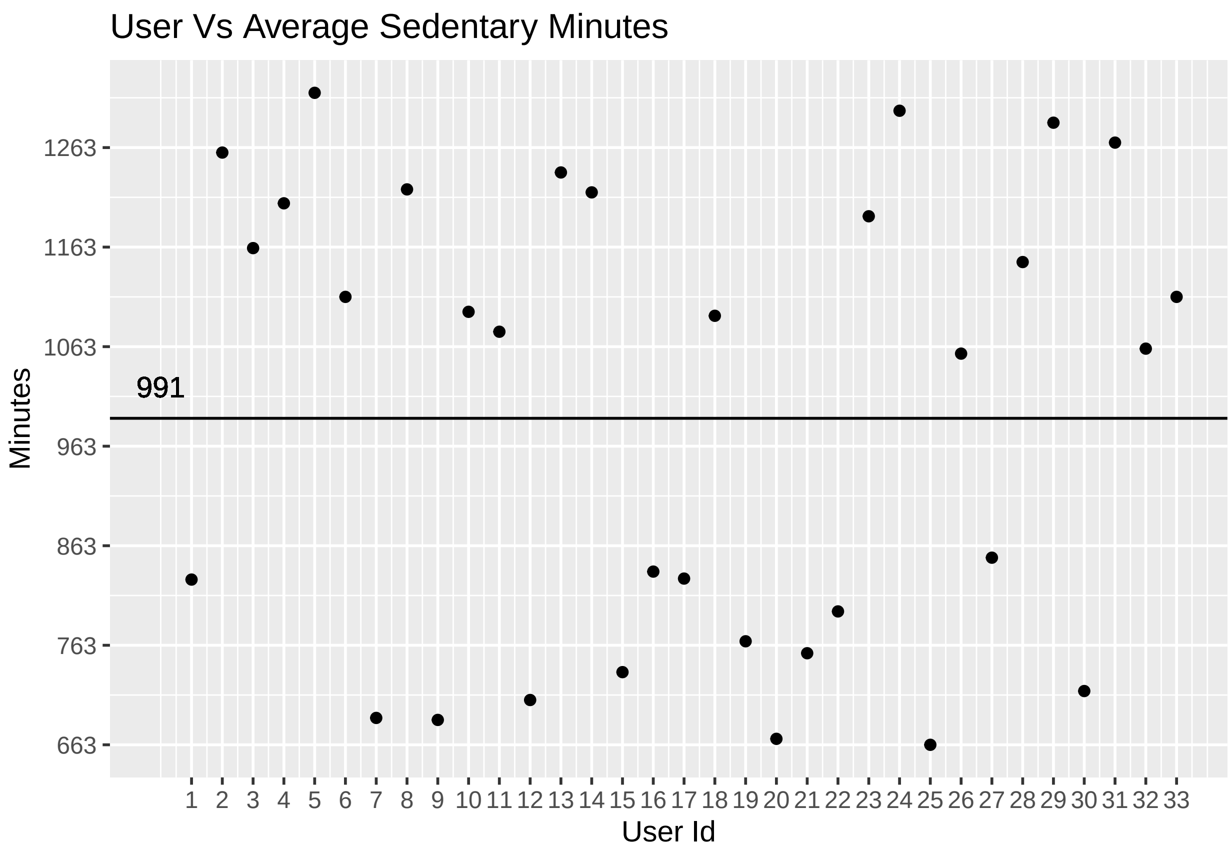
```
##Mean Value of Total steps
M1<-mean(D_activity$TotalSteps)
M1<-ceiling(M1)

##Plot from final1 User Id Vs Avg Step
P1<-ggplot(data = final1)+geom_point(mapping =aes(x=Simple_Id,y=Avgstep))+
    geom_hline(yintercept = M1)+ geom_text(aes(0,M1,label = M1, vjust = -1))+
    scale_x_continuous(breaks = round(seq(min(final1$Simple_Id), max(final1$Simple_Id), by = 1),1))+
    scale_y_continuous(breaks = round(seq(min(final1$Avgstep), max(final1$Avgstep), by = 1000),1))
    P1+xlab("User Id") + ylab("Steps")+labs(title ="User Vs Average Steps")
```



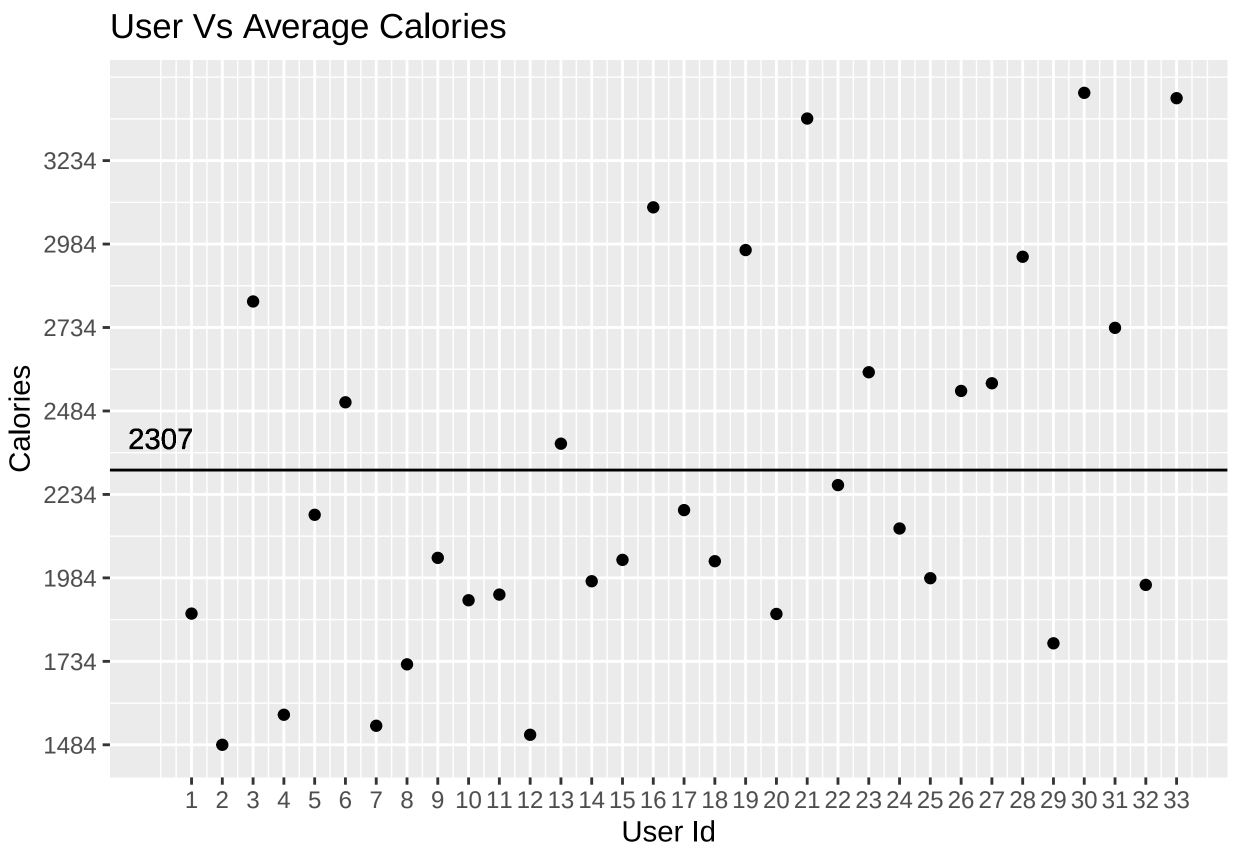
```
##Mean Value of VeryActiveMinutes
M2<- mean(D_activity$VeryActiveMinutes)
M2<-ceiling(M2)

##Plot from final1 User Id Vs AvgVeryactiveminutes
P2<-ggplot(data = final1)+geom_point(mapping =aes(x=Simple_Id,y=AvgVeryactiveminutes))+
    geom_hline(yintercept = M2)+ geom_text(aes(0,M2,label = M2, vjust = -1))+
    scale_x_continuous(breaks = round(seq(min(final1$Simple_Id), max(final1$Simple_Id), by = 1),1))+
    scale_y_continuous(breaks = round(seq(min(final1$AvgVeryactiveminutes), max(final1$AvgVeryactivemin
    P2+xlab("User Id") + ylab(" Minutes")+labs(title ="User Vs Average Very active minutes")
```



```
##Mean Value of SedentaryMinutes
M3<- mean(D_activity$SedentaryMinutes)
M3<- ceiling(M3)

##Plot from final1 User Id Vs AvgSedentaryMinutes
P3<-ggplot(data = final1)+geom_point(mapping =aes(x=Simple_Id,y=AvgSedentaryMinutes))+
    geom_hline(yintercept = M3)+ geom_text(aes(0,M3,label = M3, vjust = -1))+
    scale_x_continuous(breaks = round(seq(min(final1$Simple_Id), max(final1$Simple_Id), by = 1),1))+
    scale_y_continuous(breaks = round(seq(min(final1$AvgSedentaryMinutes), max(final1$AvgSedentaryMinut
    P3+xlab("User Id") + ylab("Minutes")+  labs(title ="User Vs Average Sedentary Minutes")
```

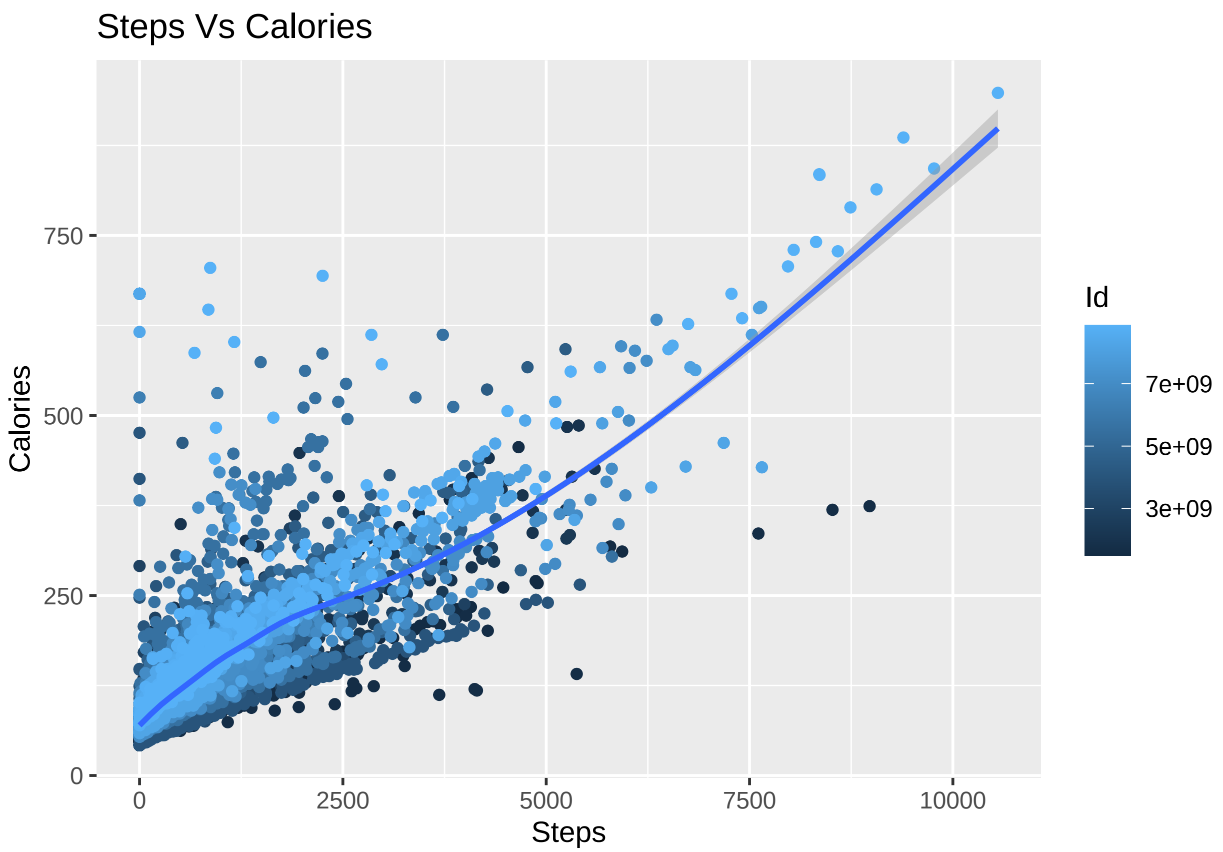


```
##Mean Value of Calories burned
M4<- mean(D_activity$Calories)
M4<- ceiling(M4)

##Plot from final1 User Id Vs AvgCalories
P4<-ggplot(data = final1)+geom_point(mapping =aes(x=Simple_Id,y=AvgCalories))+
    geom_hline(yintercept = M4)+ geom_text(aes(0,M4,label = M4, vjust = -1))+
    scale_x_continuous(breaks = round(seq(min(final1$Simple_Id), max(final1$Simple_Id), by = 1),1))+
    scale_y_continuous(breaks = round(seq(min(final1$AvgCalories), max(final1$AvgCalories), by = 250),1
    P4+xlab("User Id") + ylab("Calories")+  labs(title ="User Vs Average Calories")
```



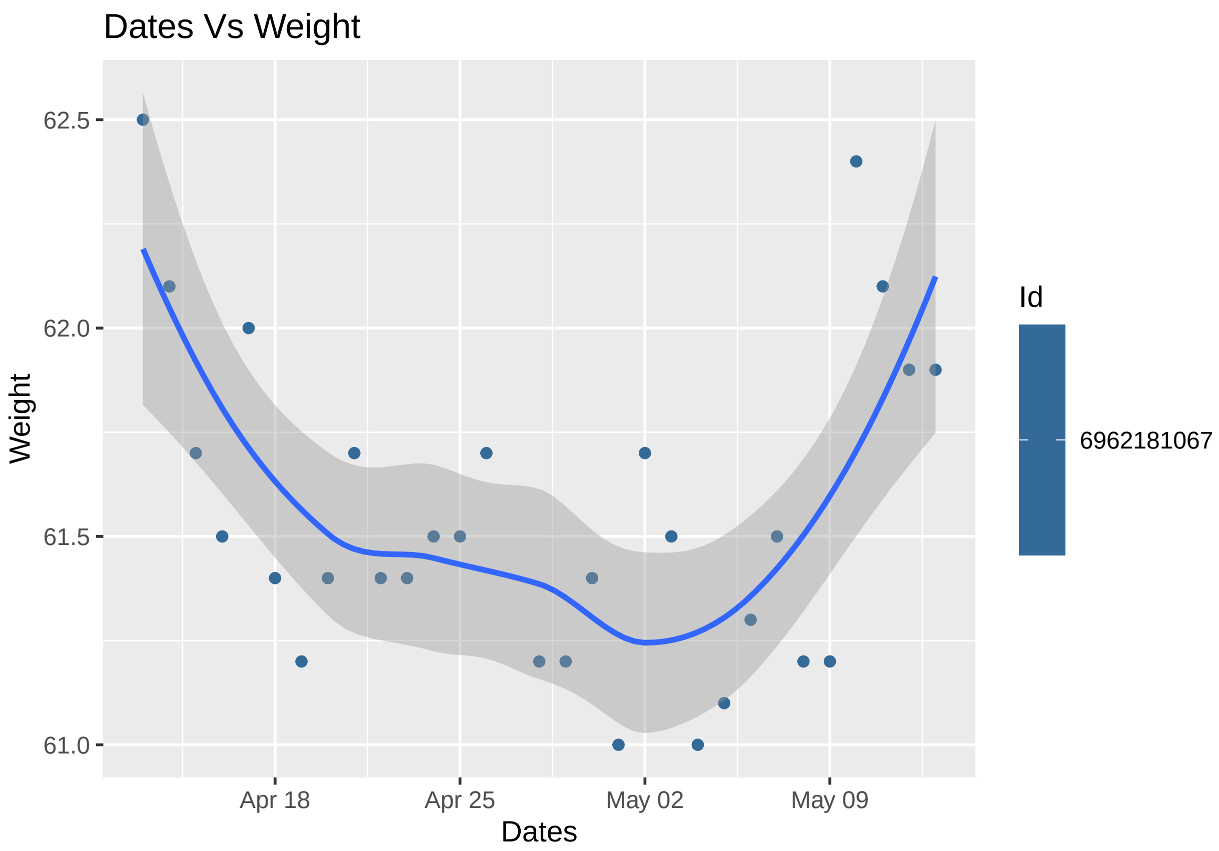
```
##Plot from final1 StepTotal Vs AvgCalories
P5<-ggplot(data = H_merged)+geom_point(mapping = aes(x=StepTotal,y=Calories,color=Id))+
    geom_smooth(mapping = aes(x=StepTotal,y=Calories,color=Id))
    P5+xlab("Steps") + ylab("Calories")+  labs(title ="Steps Vs Calories")
```

```
## 'geom_smooth()' using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



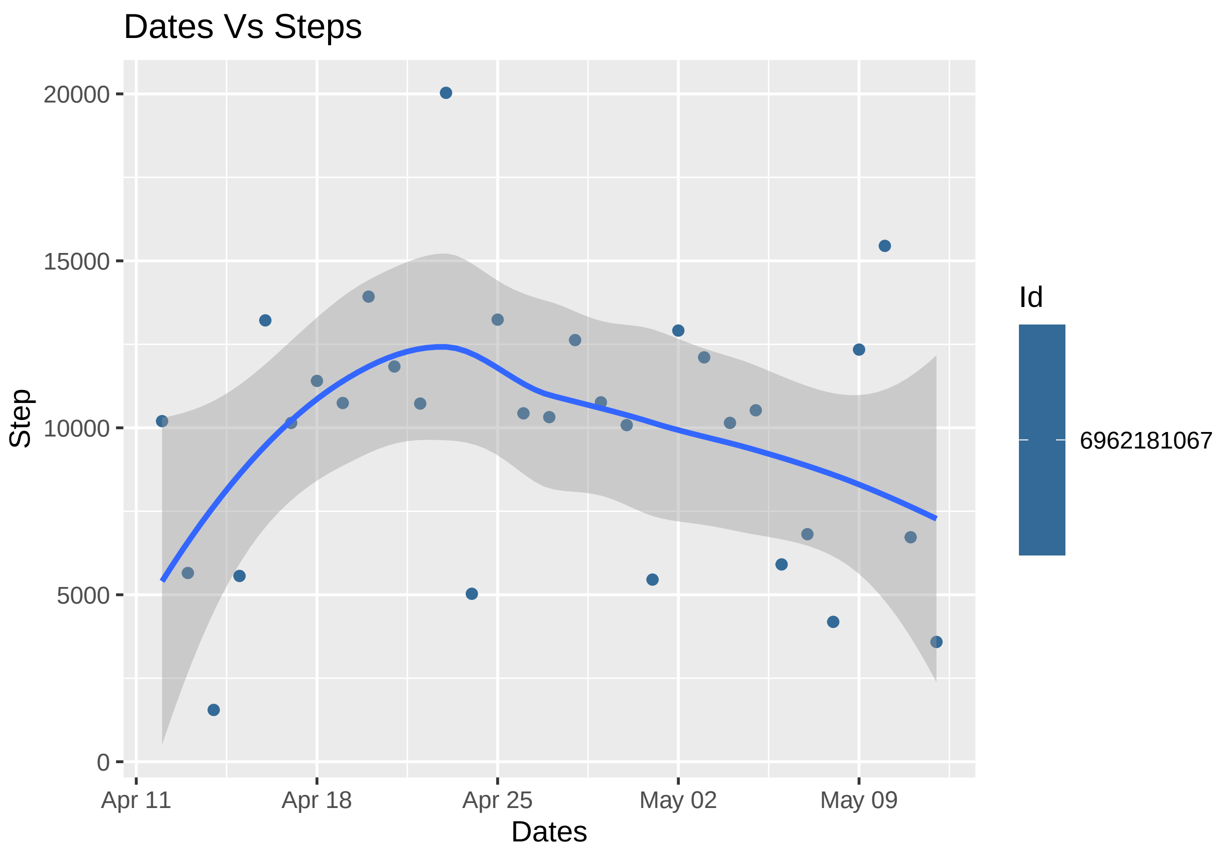
```
##plot from userweight_df Dates Vs Weight for user 25
 userweight_df<- filter(weightLog,weightLog$Id == distinctid_df[25,1])

 ggplot(data = userweight_df) +geom_point(mapping = aes(x=NewDate1,y=WeightKg,color = Id))+
 geom_smooth(mapping = aes(x=NewDate1,y=WeightKg))+
 xlab("Dates") + ylab("Weight")+  labs(title ="Dates Vs Weight")
```

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



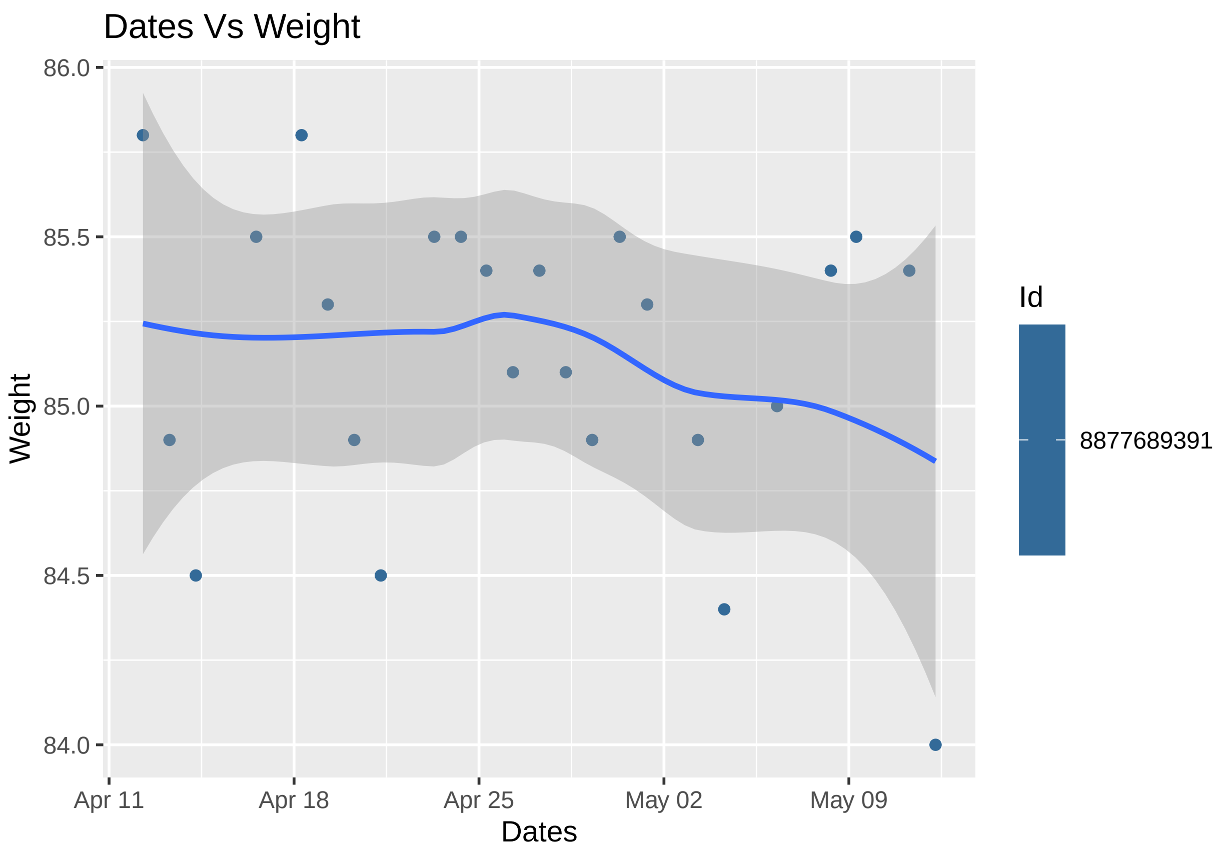
```
##plot from D_activity Dates Vs steps for user 25
 user_df<- filter(D_activity,D_activity$Id == distinctid_df[25,1])

 ggplot(data = user_df)+geom_point(mapping = aes(x=New_date,y=TotalSteps,color = Id))+
 geom_smooth(mapping = aes(x=New_date,y=TotalSteps))+
 xlab("Dates") + ylab("Step")+  labs(title ="Dates Vs Steps")
```

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



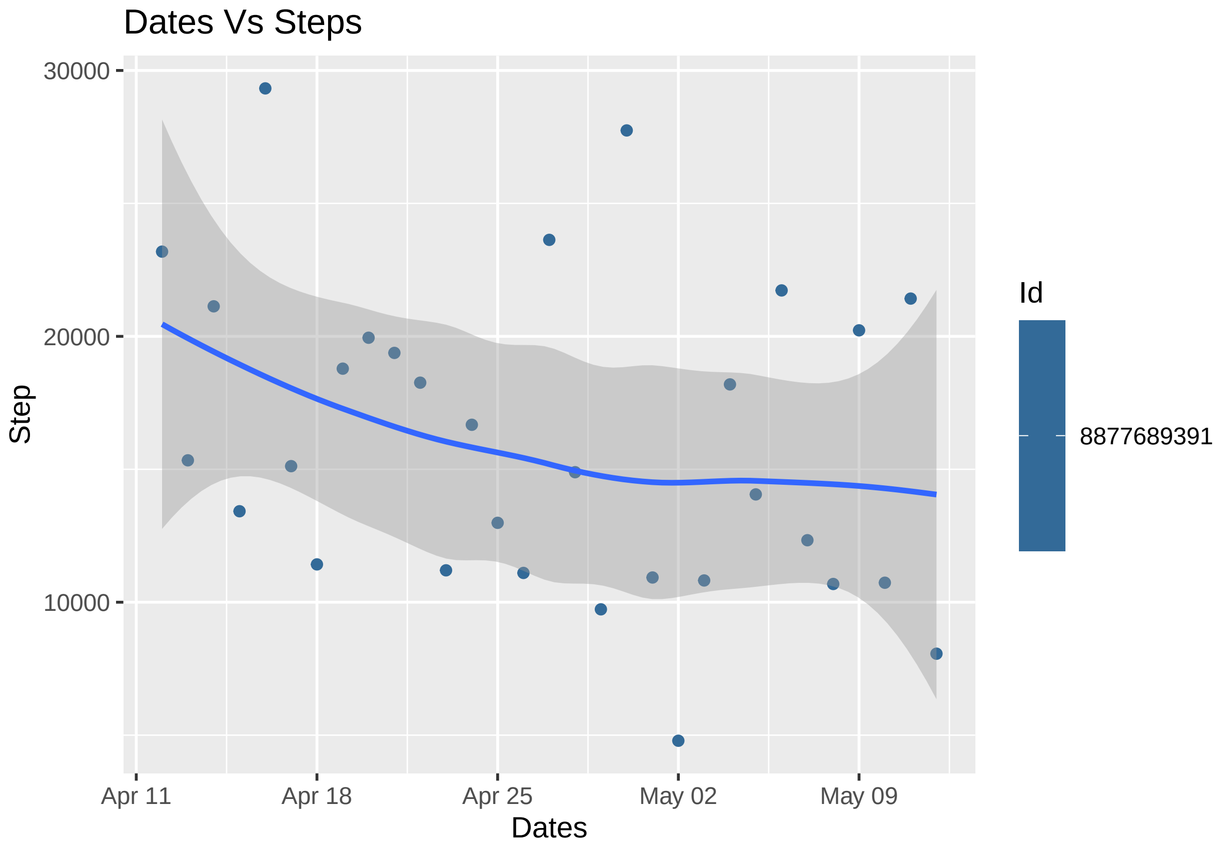
```
##plot from userweight_df Dates Vs Weight for user 33
userweight_df<- filter(weightLog,weightLog$Id == distinctid_df[33,1])

ggplot(data = userweight_df) +geom_point(mapping = aes(x=NewDate1,y=WeightKg,color = Id))+
geom_smooth(mapping = aes(x=NewDate1,y=WeightKg))+
xlab("Dates") + ylab("Weight")+  labs(title ="Dates Vs Weight")
```

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



```
##plot from D_activity Dates Vs steps for user 33
 user_df<- filter(D_activity,D_activity$Id == distinctid_df[33,1])

 ggplot(data = user_df)+geom_point(mapping = aes(x=New_date,y=TotalSteps,color = Id))+
 geom_smooth(mapping = aes(x=New_date,y=TotalSteps))+
 xlab("Dates") + ylab("Step")+  labs(title ="Dates Vs Steps")
```

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



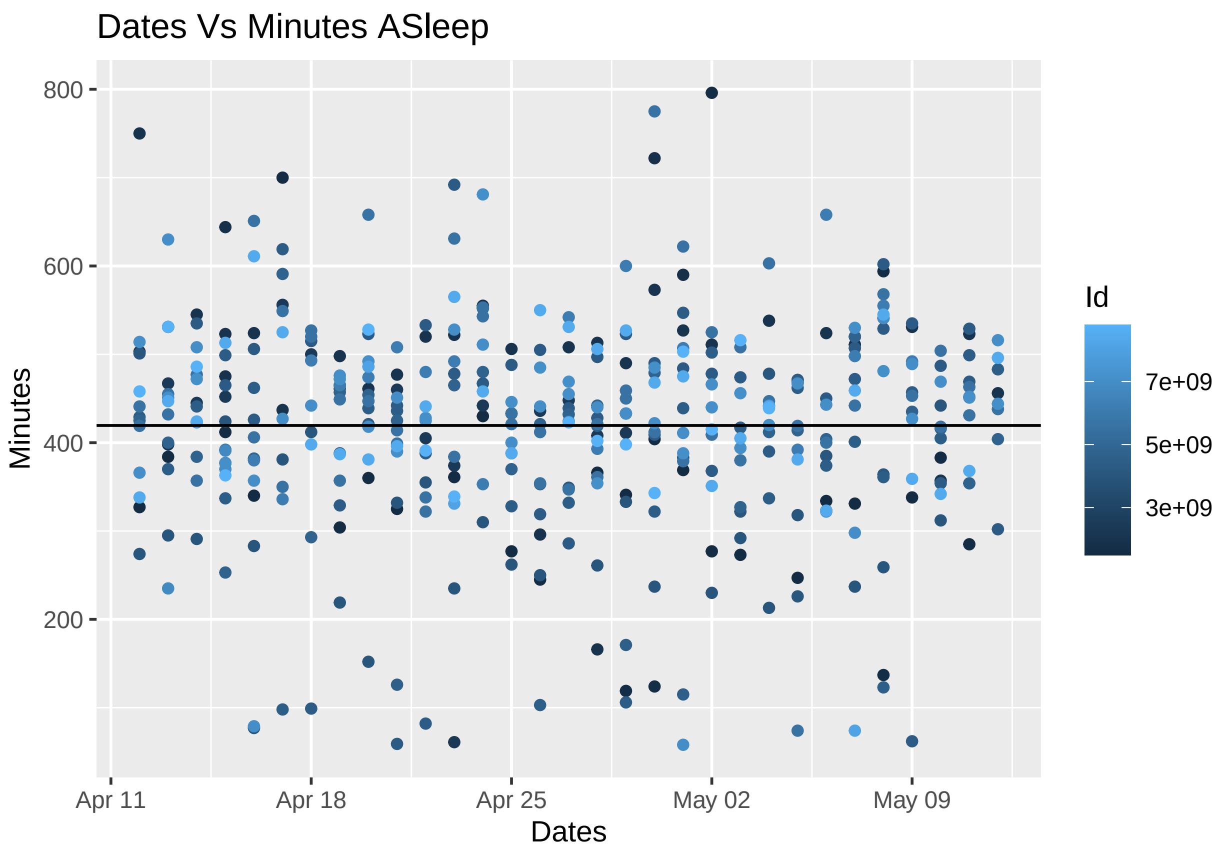
```
##calculate the difference and mean of TotalTimeInBed and TotalMinutesAsleep
D <-D_Sleep$TotalTimeInBed - D_Sleep$TotalMinutesAsleep
mean(D)
```

```
## [1] 39.17191
```

```
##Mean of TotalMinutesAsleep
M5<-mean(D_Sleep$TotalMinutesAsleep)
M5
```

```
## [1] 419.4673
```

```
 ##plot from D_sleep Dates Vs Minutes ASleep steps for user 33
   ggplot(D_Sleep, aes(x=Newsleep_Day)) +
   geom_point(aes(y = TotalMinutesAsleep ,color = Id ))+
   geom_hline(yintercept = M5) +
   xlab("Dates") + ylab("Minutes")+  labs(title ="Dates Vs Minutes ASleep")
```



Step 5 - Share

Assumptions and Limitations
1) User gender and Age group data are not available ,it is assumed to be adults from 18-65 years old
2) Data from 2016 is not sufficient for addressing current trend in 2022
3) Underlying health problems and food habits are not considered
4) Data is collected from 12/4/2016 - 12/5/2016 on 33 users which is small for effective analysis
5) Any tracking error is not considered

Insights

1)From the data,users recorded a average of 7647 steps/day and 50% users recorded a below average number of steps

2)From the data, users recorded a average of 22 minutes of Very Active Minutes/day and 67% users recorded a below average number of Very Active Minutes

3)From the data, users recorded a average 2307 calories burned/day and 60% users recorded a below average calories burn

4)From the data, users recorded a average 991 sedentary minutes/day and 57% users recorded above average sedentary minutes

5) Steps walked and Calories burned shows positive correlation which is obvious

6)users 25(id 6962181067) and 33(id 8877689391) weight loss is analysed shows a decrease in weight over time but it can't be concluded that no of steps resulted in weight loss . Not enough data to support it. further analysis needed.

7) From the data, users recorded a average of 420 minutes/day of sleep over the time period

Step 6 - Act

Recommendations

1) World Health Organisation recommends 10000steps/day for active and healthy lifestyle but the data shows average steps taken per day is below recommended level. We can use this for Bellabeat marketing strategy by introducing a feature that notify the user about the number of steps needed to complete WHO recommendations.

2) 67 % of Users recorded a below Average very active minutes that means majority of users doesn't have a active workout plan ,we need to have a feature to support daily workout plan by setting goals

3) Average calories burned is almost equal to global average calories (2500) consumed . for users looking for weight loss minimum steps recommendation should be higher

4) Average Sedentary hours is 16.5 which means users are working for long hour in office without any activities , push notifications should be given in a timely intervals to notify the user to take a break and move around.

5) Using mobile or other gadgets is not recommended in bed .if mobile is used for long time in bed it is recommended to create a alert on over usage

6) Highlighting the above mentioned feature and strategies in our upcoming product line , appeal to potential new and existing consumer with this brand new features in a advertising campaign.