BellaBeat Capstone Project

Sumita Pathania

2022-05-03

Case Study 2: How Can a Wellness Technology Company Play It Smart? BELLABEAT

Bellabeat is a high-tech manufacturer of health-focused products for women. As a junior data analyst working with 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. Urška Sršen is confident that an analysis of non-Bellebeat consumer data (ie. FitBit fitness tracker usage data) would reveal more opportunities for growth. The insights from the data will help to guide marketing strategy for the company. I have performed analysis on data along with high level recommendations for Bellabeat's marketing strategy.

Business Task:

Analyze FitBit fitness tracker data to gain insights into how consumers are using the FitBit app and discover trends for Bellabeat marketing strategy.

Step 1: Ask Phase

Key Stakeholders:

1. Urška Sršen: Bellabeat's co-founder and Chief Creative Officer
2. Sando Mur: Mathematician and Bellabeat's co-founder; key member of the Bellabeat executive team
3. Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

Business Objectives:

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?

Step 2: Prepare Phase

Sršen encouraged me to use public data that explores smart device users' daily habits. She points me to a specific data set:

FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits. Data is publicly available on Kaggle: FitBit Fitness Tracker Data and stored in 18 csv files.

In the Prepare phase, we identify the data being used and its limitations.

- Data is collected 7 years ago in 2016. Users' daily activity, fitness and sleeping habits, diet and food consumption may have changed since then. Data may not be timely or relevant.
- Sample size of 30 FitBit users is not representative of the entire fitness population.
- Dataset can be downloaded by Clicking Here

Is Data ROCCC?

A good data source is ROCCC which stands for Reliable, Original, Comprehensive, Current, and Cited.

- Reliable — LOW — Not reliable as it only has 30 respondents
- Original — LOW — Third party provider (Amazon Mechanical Turk)
- Comprehensive — MED — Parameters match most of Bellabeat products' parameters
- Current — LOW — Data is 7 years old and may not be relevant
- Cited — HIGH – data collector and source is well documented

Step 3: Process Phase

In this phase we will process the data by cleaning and ensuring that it is correct,relevant,complete and error free.

We have to check if data contains any missing or null values Transform the data into format we want for the analysis

Tool:

I have used RStudio for data cleaning,data transformation,data analysis and visualization.

Firstly, we need to install and read the packages we need for analysis: I have all packages installed, so I read all the packages simultaneously.

Setting Up Environment

```r
install.packages("tidyverse")
install.packages("ggplot2")
install.packages("skimr")
install.packages("sqldf")
install.packages("janitor")

library(tidyverse)
library(ggplot2)
library(lubridate)           #for dates and times
library(ggplot2)             #for data viz
library(dplyr)               #for data manipulation
library(skimr)               #for summarizing data
library(sqldf)               #for using SQL queries
library(janitor)
```

We can read the data stored from secured hard disk with help of command read.csv and store them in a variable of our choice.

```r
daily_activity = read.csv("/cloud/project/dailyActivity_merged.csv")
daily_sleep = read.csv("/cloud/project/sleepDay_merged.csv")
weight_log = read.csv("/cloud/project/weightLogInfo_merged.csv")
```

We need to see if there are any null or missing values in the data. We can check this using the following commands.

```
str(daily_activity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "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 ...
```

```
skim(daily_activity)
```

Table 1: Data summary

Name	daily_activity
Number of rows	940
Number of columns	15
Column type frequency:	
character	1
numeric	14
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDate	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Id	0	1	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09	
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04	
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01	
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01	
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00	
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480000e+00	
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01	
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01	
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02	
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02	
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02	
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440000e+03	
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03	

```r
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```r
str(daily_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 ...
```

```r
skim(daily_sleep)
```

Table 4: Data summary

Name	daily_sleep
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
SleepDay	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Id	0	1	5.000979e+09	2.06036e+09	1503960366	3977333714	4702921684	6962181067	8792009665	
TotalSleepRecords	0	1	1.120000e+00	3.50000e-01	1	1	1	1	3	
TotalMinutesAsleep	0	1	4.194700e+02	1.18340e+02	58	361	433	490	796	
TotalTimeInBed	0	1	4.586400e+02	1.27100e+02	61	403	463	526	961	

```
head(daily_sleep)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```
str(weight_log)
```

```
## '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 ...
```

```r
skim(weight_log)
```

Table 7: Data summary

Name	weight_log
Number of rows	67
Number of columns	8
Column type frequency:	
character	2
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Date	0	1	19	21	0	56	0
IsManualReport	0	1	4	5	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Id	0	1.00	7.009282e+09	1.950322e+09	1.503960e+09	6.962181e+09	6.962181e+09	8.877689e+09	8.877689e+09	
WeightKg	0	1.00	7.204000e+01	1.392000e+01	5.260000e+01	6.140000e+01	6.250000e+01	8.505000e+01	1.335000e+02	
WeightPounds	0	1.00	1.588100e+02	3.070000e+01	1.159600e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02	
Fat	65	0.03	2.350000e+01	2.120000e+00	2.200000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+01	
BMI	0	1.00	2.519000e+01	3.070000e+00	2.145000e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01	
LogId	0	1.00	1.461772e+12	7.829948e+08	1.460444e+12	1.461079e+12	1.461802e+12	1.462375e+12	1.463098e+12	

```r
head(weight_log)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

After executing these commands we found out the:

- Number of records and columns.
- Number of null and non null values.
- Data type of every columns.

So we get to know that there are 940 records in daily__activity data, 413 in daily__sleep and 67 in weight__log. There are no null values present in any of the data set, So there is no requirement to clean the data. But the date column is in character format, so we need to convert it into datetime64 type. I have also created month and day of week column as we need them in analysis.

```
daily_activity$Rec_Date <- as.Date(daily_activity$ActivityDate,"%m/%d/%y")
daily_activity$month <- format(daily_activity$Rec_Date,"%B")
daily_activity$day_of_week <- format(daily_activity$Rec_Date,"%A")
head(daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories   Rec_Date
## 1                  13                  328              728     1985 2020-04-12
## 2                  19                  217              776     1797 2020-04-13
## 3                  11                  181             1218     1776 2020-04-14
## 4                  34                  209              726     1745 2020-04-15
## 5                  10                  221              773     1863 2020-04-16
## 6                  20                  164              539     1728 2020-04-17
##   month day_of_week
## 1 April      Sunday
## 2 April      Monday
## 3 April     Tuesday
## 4 April   Wednesday
## 5 April    Thursday
## 6 April      Friday
```

We are also going to count unique IDs to confirm whether data has 30 IDs as claimed by the survey.

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

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

There are 33 unique IDs, instead of 30 unique IDs as expected.Some users may have created additional IDs during the survey period.

Now the data cleaning and manipulation is done.Now data is ready to be analyzed.

Step 4 : Analyze Phase

Now, we need to summarize the data. So that we can find some insights about the data.

Statistical analysis of daily_activity Dataset

```
daily_activity %>%
  select(TotalSteps,TotalDistance,SedentaryMinutes,VeryActiveMinutes) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    SedentaryMinutes VeryActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :1057.5   Median :  4.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   : 21.16
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.: 32.00
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :210.00
```

Findings:

1. The average count of recorded steps is **7638** which is less than recommended **10000** steps and average of total distance covered is **5.490 km** which is also less than recommended **8 km** mark.
2. The average sedentary minutes is 991.2 minutes or **16.52 hours** which is very high as it should be at most **7 hours**.Even if you are doing enough physical activity, sitting for more than 7 to 10 hours a day is bad for your health. (source: HealthyWA article).
3. The average of very active minutes is **21.16** which is less than target of **30** minutes per day. (source:verywell fit)

Statistical analysis of weight_log Dataset

```
weight_log %>%
  select(WeightKg,BMI) %>%
  summary()
```

```
##     WeightKg           BMI
##  Min.   : 52.60   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:23.96
##  Median : 62.50   Median :24.39
##  Mean   : 72.04   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:25.56
##  Max.   :133.50   Max.   :47.54
```

Findings:

1. We can not conclude healthiness of person just by knowing there weight, There are other factors like height,fat percentage affect in the health.
2. The average of BMI is **25.19** which is slightly grater than the healthy BMI range which is between **18 and 24.9.**

Statistical analysis of Avg_minutes_asleep Dataset

```
Avg_minutes_asleep <- sqldf("SELECT SUM(TotalSleepRecords),SUM(TotalMinutesAsleep)
/SUM(TotalSleepRecords)
As avg_sleeptime FROM daily_sleep")
Avg_minutes_asleep
```

```
##   SUM(TotalSleepRecords) avg_sleeptime
## 1                    462           374
```

Statistical analysis of Average minutes asleep Dataset

```
Avg_TimeInBed <- sqldf("SELECT SUM(TotalTimeInBed)/SUM(TotalSleepRecords)
                 As avg_timeInBed FROM daily_sleep")

Avg_TimeInBed
```

```
##   avg_timeInBed
## 1           409
```

Findings :

There is difference of 35 minutes between time in bed and sleep time that means it takes on an average 20 to 30 minutes to fall asleep for peoples.

We will also calculate number of distinct records in daily sleep and weight log data.

```
n_distinct(daily_sleep$Id)
```

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

```
n_distinct(weight_log$Id)
```

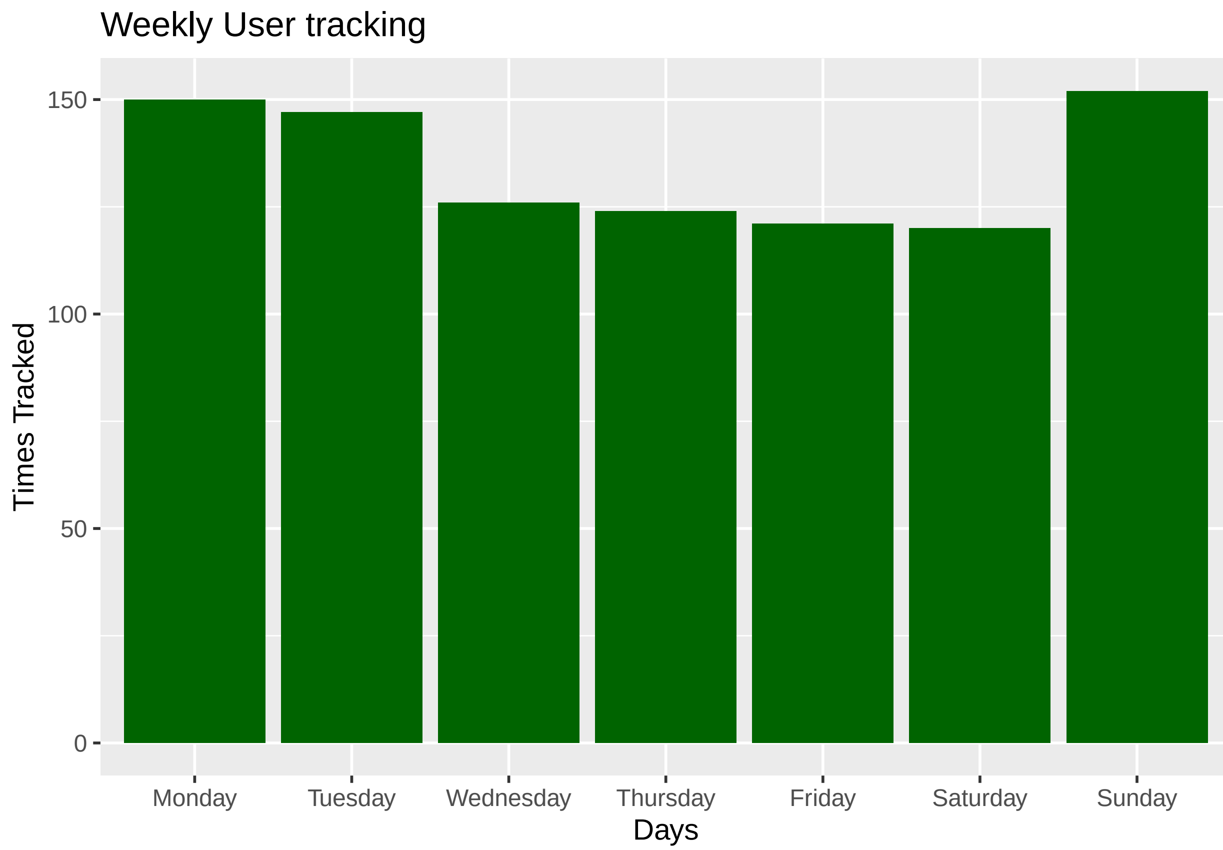
```
## [1] 8
```

Step 5: Share Phase

In this step, we will create and share some Data visualizations based on our analysis and goals of the project.

```
daily_activity$day_of_week <- ordered(daily_activity$day_of_week,levels=c("Monday","Tuesday",
                                                                          "Wednesday","Thursday","Frida

ggplot(data=daily_activity) + geom_bar(mapping = aes(x=day_of_week),fill="Dark Green") +
  labs(x="Days",y="Times Tracked",title="Weekly User tracking")
```



As we can see, the frequency of usage of FitBit fitness tracker application is high on **Sunday, Monday and Tuesday** than other week days.I think this behavior is because people get busier in week end days due to work pressure and they don't get enough time to track their activity.That's why people are more active on Sunday and starting 2 days of week.

Calculating Average steps walked

```
mean_steps <- mean(daily_activity$TotalSteps)
mean_steps
```

```
## [1] 7637.911
```

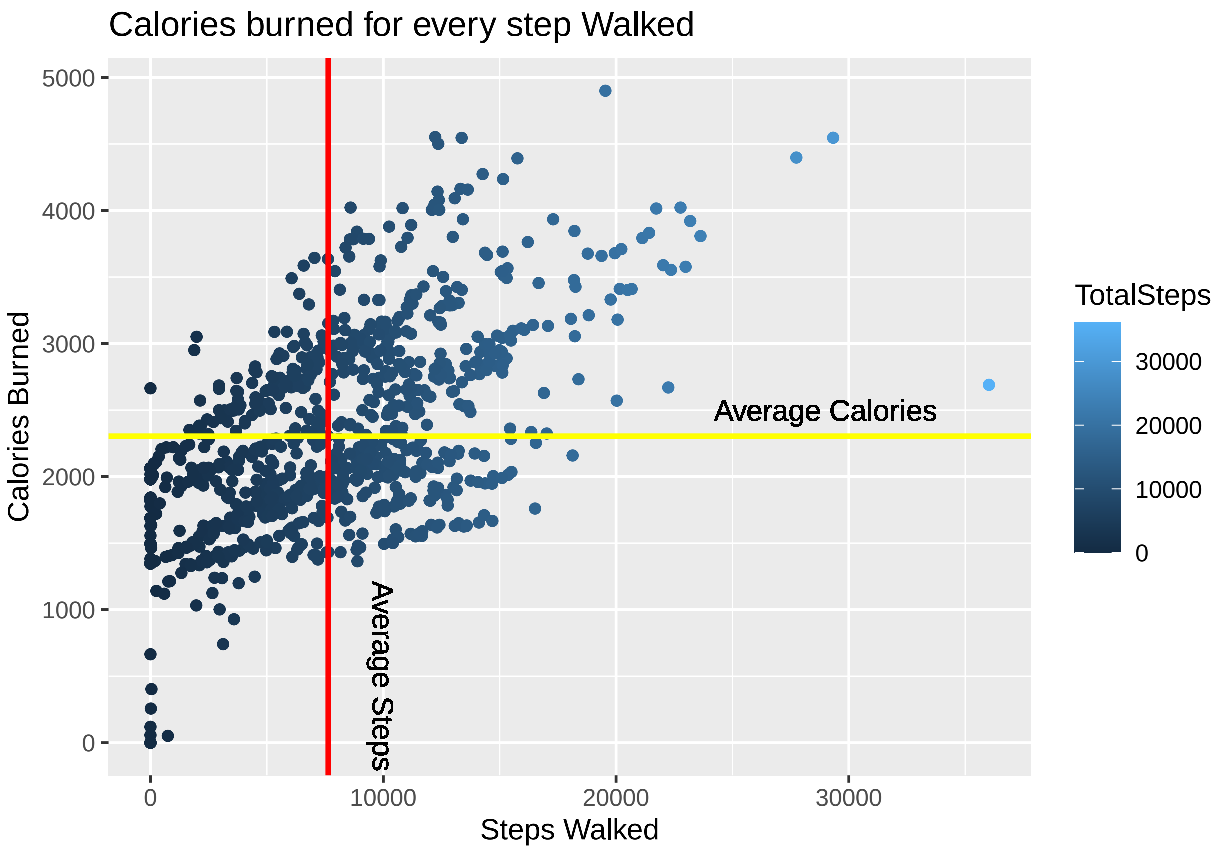
Calculating Average Calories Burned

```
mean_calories <- mean(daily_activity$Calories)
mean_calories
```

```
## [1] 2303.61
```

Calories burned for every step walked

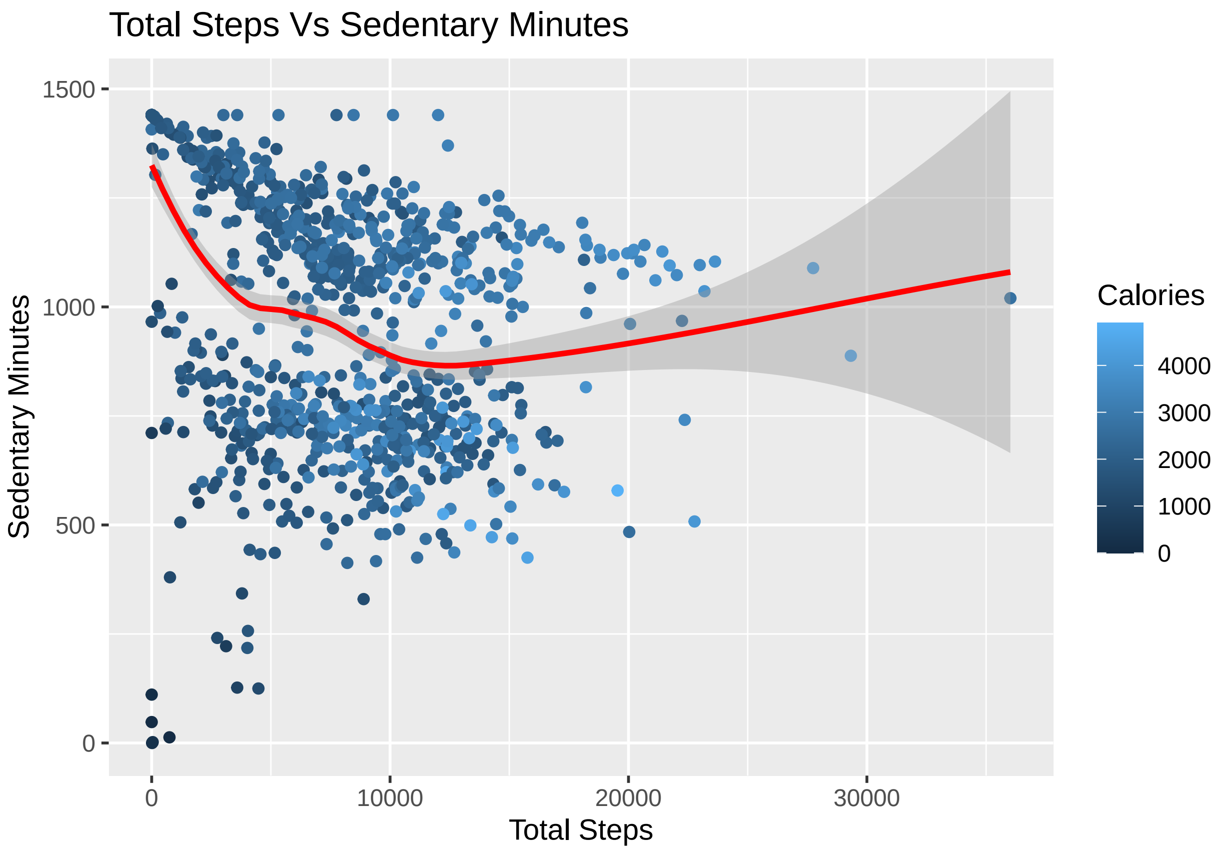
```
ggplot(data=daily_activity) + geom_point(mapping=aes(x=TotalSteps, y=Calories, color=TotalSteps)) +
geom_hline(mapping = aes(yintercept=mean_calories),color="yellow",lwd=1.0) +
  geom_vline(mapping = aes(xintercept=mean_steps),color="red",lwd=1.0) +
  geom_text(mapping = aes(x=10000,y=500,label="Average Steps",srt=-90)) +
  geom_text(mapping = aes(x=29000,y=2500,label="Average Calories")) +
  labs(x="Steps Walked",y="Calories Burned",title = "Calories burned for every step Walked")
```



- It is a positive correlation with some outliers at bottom and top of scatter plot.
- It is clear from the plot that intensity of calories burned increase with number of steps taken.

```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes, color = Calories)) + geom_point() +
geom_smooth(method = "loess",color="red") +
labs(x="Total Steps",y="Sedentary Minutes",title="Total Steps Vs Sedentary Minutes")
```

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



I was expecting a totally inverse relationship between steps taken and sedentary minutes.

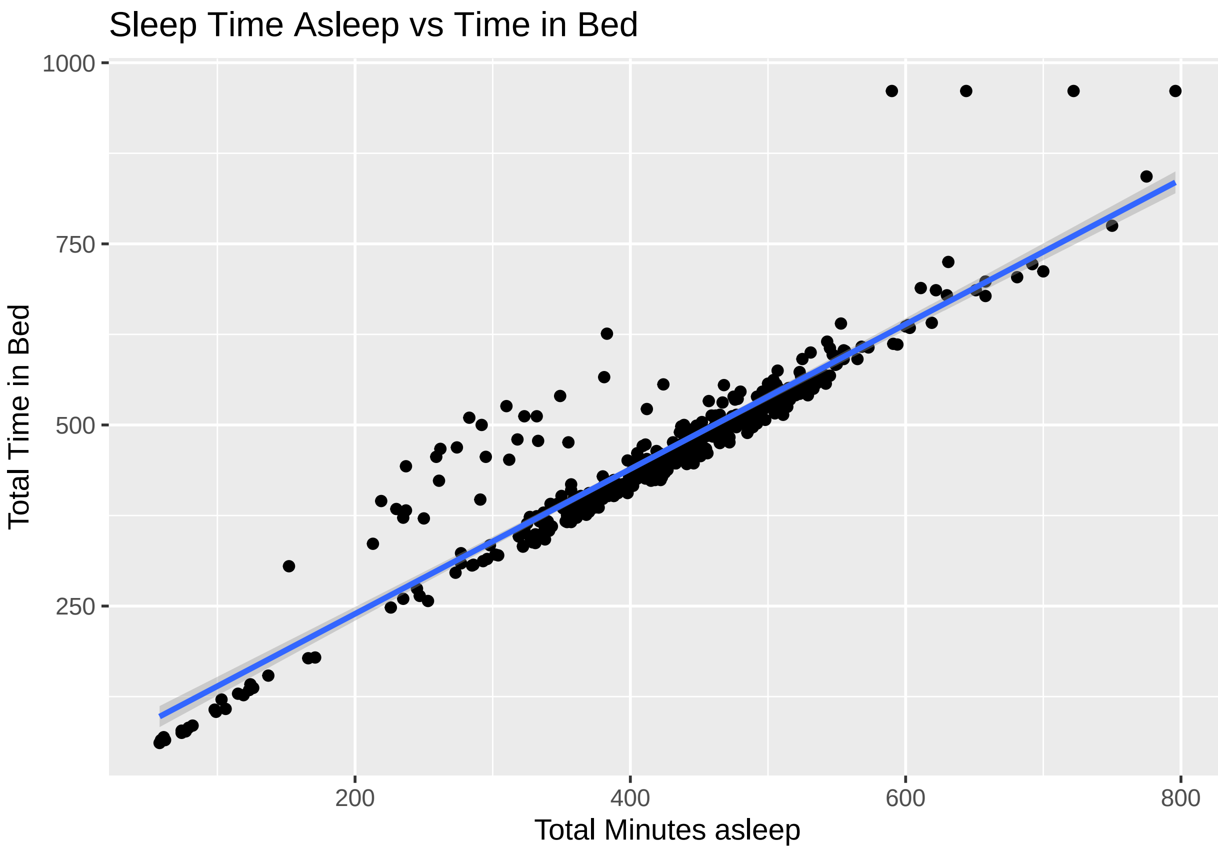
1. At the start when the steps taken are less than 10000 the relation between them is inverse, but as number of steps increases beyond 10000 there is no drastic change in relation.

2. Relation between steps and sedentary minutes after 15000 steps becomes slightly positive.

Relation between sleep and time in bed

```
ggplot(data=daily_sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed )) + geom_point() +          stat_sm
  labs(x="Total Minutes asleep", y="Total Time in Bed", title = "Sleep Time Asleep vs Time in Bed")
```

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



As we can see, there is a strong positive correlation between **TotalMinutesAsleep** and **TotalTimeInBed**, but there are some outliers in data in the middle and top of plot.

The outliers are one who spend lot of time in bed but didn't actually sleep.There can be different reasons for that.

Relation between Active minutes and Calories burned

```
ggplot(data=daily_activity,aes(x = VeryActiveMinutes, y = Calories, color = Calories)) + geom_point() +
geom_smooth(method = "loess",color="Red") +
labs(x="Active Minutes",y="Calories Burned",title = "Active Minutes vs Calories Burned")
```

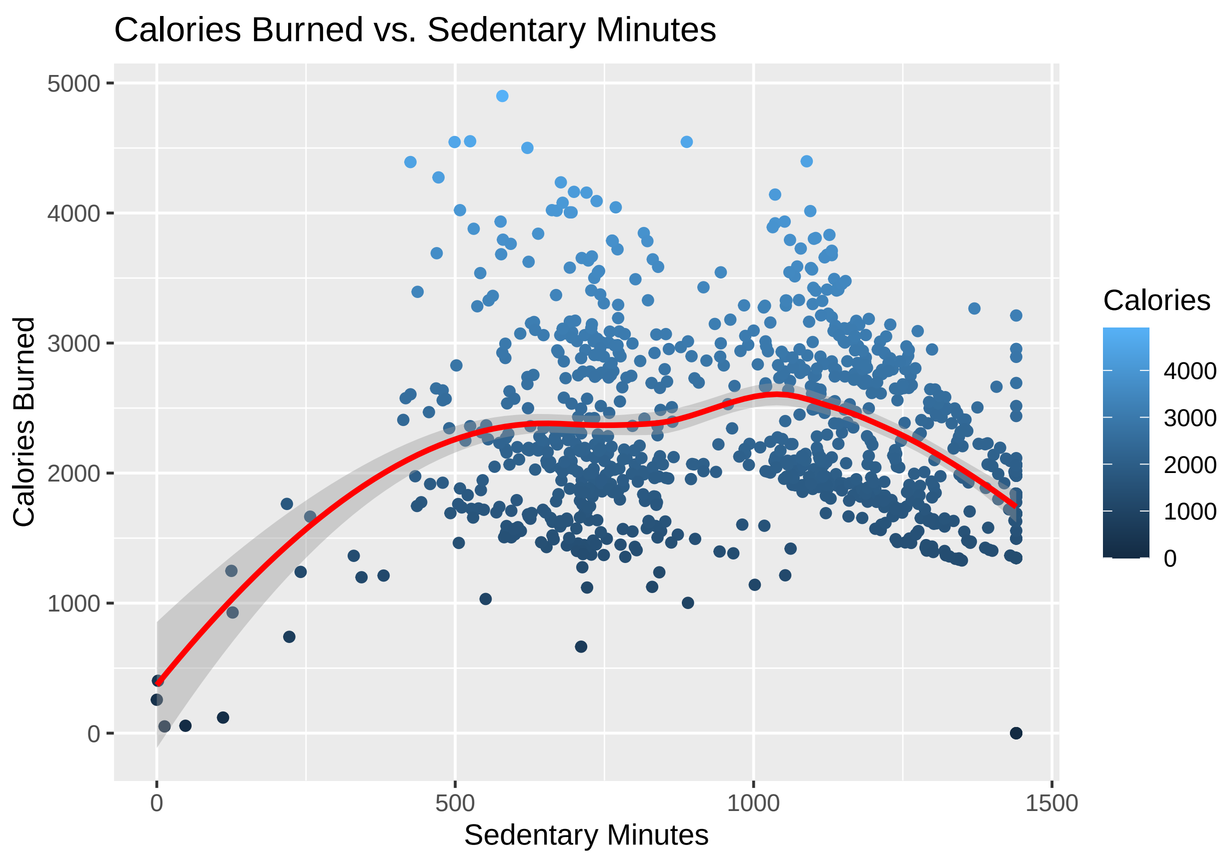
```
## `geom_smooth()` using formula 'y ~ x'
```

As we can see,active minutes and calories burned are highly correlated with each other with some outliers at bottom left and top left of the plot.

Relation between sedentary minutes and calories burned

```
ggplot(data=daily_activity,aes(x=SedentaryMinutes,y=Calories,color=Calories)) + geom_point() +
geom_smooth(method="loess",color="red") +
labs(y="Calories Burned", x="Sedentary Minutes", title="Calories Burned vs. Sedentary Minutes")
```

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



I was expecting the relation between sedentary minutes and calories burned to be totally inverse in nature.

The data is showing positive correlation up to 1000 sedentary minutes. After 1000 sedentary minutes the relation is inverse as I expected. Now,we will calculate the sum of individual minute column from daily activity data.

```
activity_min <- sqldf("SELECT SUM(VeryActiveMinutes),SUM(FairlyActiveMinutes),
      SUM(LightlyActiveMinutes),SUM(SedentaryMinutes)
      FROM daily_activity")
activity_min
```

```
##   SUM(VeryActiveMinutes) SUM(FairlyActiveMinutes) SUM(LightlyActiveMinutes)
## 1                  19895                    12751                    181244
##   SUM(SedentaryMinutes)
## 1                931738
```

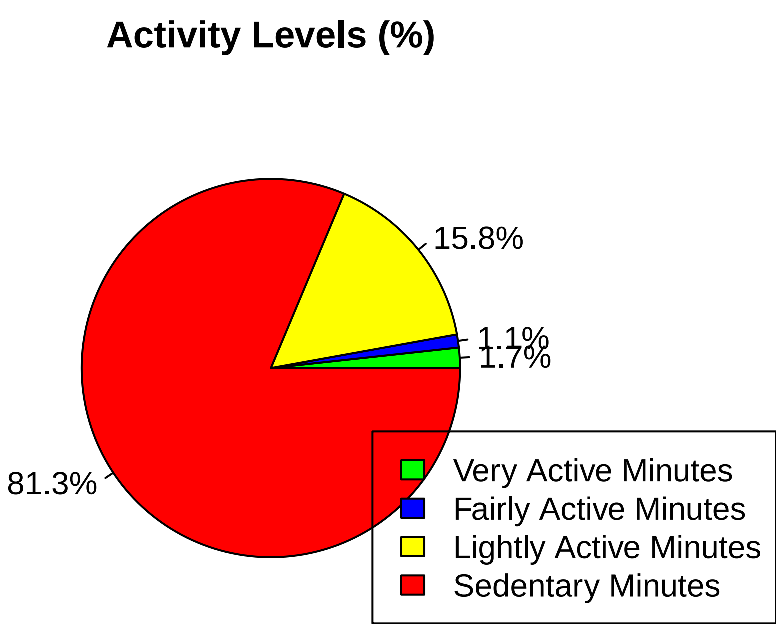
As we got the values, we will use these values to plot a pie chart to compare the percentage of activity by minutes.

```
x <- c(19895,12751,181244,931738)
x
```

```
## [1]  19895  12751 181244 931738
```

```
piepercent <- round(100*x / sum(x), 1)
colors = c("green","blue","yellow","red")

pie(x,labels = paste0(piepercent,"%"),col=colors,main = "Activity Levels (%)")
legend("bottomright",c("Very Active Minutes","Fairly Active Minutes","Lightly Active Minutes","Sedentar
```



1. The percentage of sedentary minutes is very high than all other,which covers 81.3 % of pie this indicates that people have Sedentary Lifestyle.
2. The percentage of very active and fairly active minutes is very less ie. 1.7%,1.1% respectively,which is very less.

Now, we will calculate sum of different distance values from daily activity data:

```
activity_dist <- sqldf("SELECT SUM(ModeratelyActiveDistance),SUM(LightActiveDistance),
      SUM(VeryActiveDistance),SUM(SedentaryActiveDistance)
      FROM daily_activity")
activity_dist
```

```
##   SUM(ModeratelyActiveDistance) SUM(LightActiveDistance)
## 1                        533.49                  3140.37
##   SUM(VeryActiveDistance) SUM(SedentaryActiveDistance)
## 1                 1412.52                         1.51
```

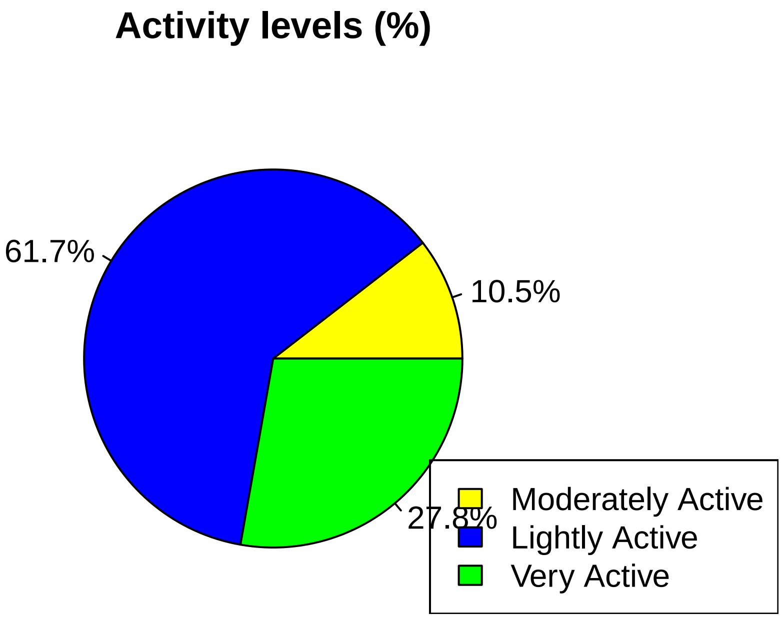
As we can see that the values of sedentaryActiveDistance is very less as compare to other distances,So I am excluding it in drawing a 3D pie chart to compare the percentage of activity in minutes.

```
y <- c(533.49,3140.37,1412.52)
y
```

```
## [1]  533.49 3140.37 1412.52
```

```
piepercent <- round(100*y / sum(y), 1)
colors = c("yellow","blue","green")

pie(y,labels = paste0(piepercent,"%"),col=colors,main = "Activity levels (%)")
legend("bottomright",c("Moderately Active","Lightly Active","Very Active"),cex=1.0,fill = colors)
```



1. The percentage of lightly active people is highest with 61.7% and that of moderately active people is 10.5%.
2. The percentage of very active people is 27.8% which is good, but it can be increased further so that people can achieve their fitness goals.

Now, we will calculate over weight people: The BMI for healthy person is between 18.5 and 24.9 and the persons who's BMI is above 24.9 are considered to be overweight.(source:CDC)

Calculating number of people who are Overweight

```
count_overweight <- sqldf("SELECT COUNT(DISTINCT(Id))
                          FROM weight_log
                          WHERE BMI > 24.9")
count_overweight
```

```
##   COUNT(DISTINCT(Id))
## 1                   5
```

As we got the values, we will use these values to plot a pie chart to compare overweight people vs healthy people.

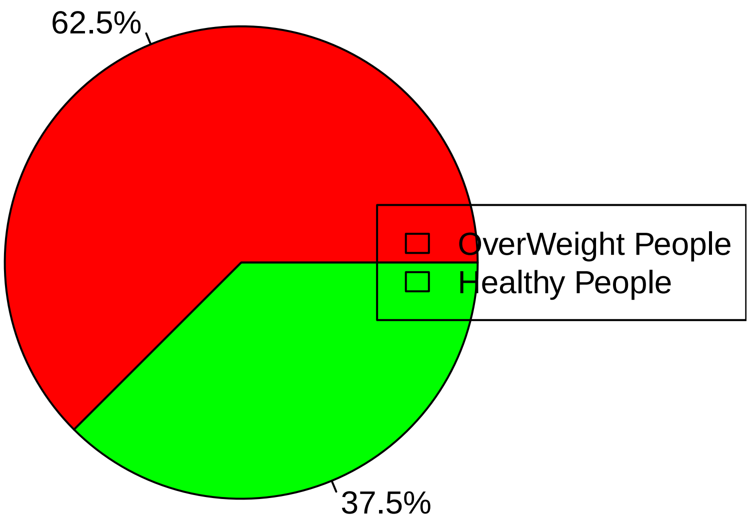
```
z <- c(5,3)
z
```

```
## [1] 5 3
```

```
piepercent <- round(100*z / sum(z),1)
colors = c("red","green")

pie(z,labels=paste0(piepercent,"%"),explode=0.1,col=colors,radius=1,main="OverWeight vs Healthy People")
legend("right",c("OverWeight People","Healthy People"),cex=1.0,fill=colors)
```

OverWeight vs Healthy People



The percentage of people with over weight is 62.5% which is high as compared to percentage of people with healthy weigh which is 37.5%. So, there is a very good opportunity to increase the percentage of people with healthy weight.

Step 6: Act Phase

Based on analysis I have following recommendations:

1. We have analysed that most of the people use application to track the steps and calories burned;less number of people use it to track sleep and very few use it to track weight records.So, I will suggest to focus on step,calories and sleep tracking more in application.
2. Majority of users 81.3% who are using the FitBit app are inactive for longer period of time and not using it for tracking their health habits.So, this can be a great chance to use this information for market strategy as Bellabeat can alert people about their sedentary behavior time to time either on application or on tracker itself .
3. Majority of the users 62.5% who are using fitness tracker are overweight.So, there is an opportunity to influence and motivate people so that they can become healthier.Also, this shows fitness products can be marketed towards people who wants to get healthy.
4. Bellabeat marketing team can encourage users by educating and equipping them with knowledge about fitness benefits, suggest different types of exercises, calories intake and burn rate information on Bellabeat application.