Google Capstone Project: How Can Bellabeat ,A Wellness Technology Company Play It Smart

Sandeep Singh

3/5/2022

Steps-1 : Ask

We define the issue, the goals of our case study, and the intended result in this step.

1.0 Background

Since 2013, Bellabeat has been a high-tech manufacturer of gorgeously crafted smart devices for women with a focus on their health. Bellabeat has quickly expanded and established itself as a tech-driven wellness brand for women by educating and empowering women with knowledge about their own health and habits.

Urka Sren, co-founder and chief creative officer, is sure that examination of non-Bellebeat customer data, such as usage data from FitBit fitness trackers, will find further development prospects.

1.2 Business Task:

Examine FitBit fitness tracker data to learn how users interact with the FitBit app and identify trends for Bellabeat marketing strategy.

1.3 Business Objectives:

- What patterns have been found?
- How might Bellabeat customers be affected by these trends?
- How can these developments affect Bellabeat's marketing plan?

1.4 Deliverables

- A concise description of the business task
- A list of all the data sources that were used, along with documentation of any data cleaning or manipulation, and a summary of the analysis
- supporting images and important findings
- Recommendations for high-level material based on the analysis

1.5 Key Stakeholders

- Bellabeat's cofounder and chief creative officer, Urka Sren
- Mathematician, co-founder of Bellabeat, and essential member of the Bellabeat management team, Sando Mur
- The Bellabeat analytics team for marketing: Data analysts overseeing Bellabeat's marketing plan.

Steps-2 : Prepare

We identify the data being used and its constraints during the Prepare step.

Step 2.1 Information on Data Source:

- 18 csv files containing data from the FitBit fitness tracker are freely accessible on Kaggle.
- generated by survey participants using Amazon Mechanical Turk between March 12 and May 12, 2016.
- 30 FitBit users gave their permission for personal tracker data to be submitted.
- The information gathered includes minute-by-minute records of physical activity, heart rate, sleep patterns, daily activities, and steps.

2.2 Limitation of Data Set

- Data was gathered in 2016 five years ago. Since then, users' routines for everyday activity, eating, exercising, and sleeping may have changed. Data might not be current or pertinent.
- A sample size of 30 FitBit users does not accurately represent the fitness market as a whole.
- We are unable to verify the integrity or correctness of the data because it is acquired through a survey.

2.3 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 5 years old and may not be relevant
- Cited — LOW — Data collected from third party, hence unknown

Overall, the dataset is regarded as having low quality data, and it is not advised to base business suggestions on it.

2.4 Data Selection

The next file is chosen and copied for examination.

2.5 Tool

We are using R for data Cleaning ,transformation and Visualization

'dailyActivity_merged.csv'

Step-3 Process

Here, we will process the data by cleaning it and making sure it is accurate, pertinent, comprehensive, free of error, and free of outliers by carrying out the following actions:

- Examine and watch the data
- Examine and deal with any missing or null values.
- Data transformation: format the data type
- Carrying out a preliminary statistical analysis

```
install.packages("Rtools", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Sandeep SIngh/Documents/R/win-
library/4.1'
## (as 'lib' is unspecified)

install.packages("tidyverse", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Sandeep SIngh/Documents/R/win-
library/4.1'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Sandeep SIngh\AppData\Local\Temp\RtmpuIVRgU\downloaded_packages

install.packages("plotrix", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Sandeep SIngh/Documents/R/win-
library/4.1'
## (as 'lib' is unspecified)

## package 'plotrix' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Sandeep SIngh\AppData\Local\Temp\RtmpuIVRgU\downloaded_packages

install.packages("treemap", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Sandeep SIngh/Documents/R/win-
library/4.1'
## (as 'lib' is unspecified)

## package 'treemap' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Sandeep SIngh\AppData\Local\Temp\RtmpuIVRgU\downloaded_packages
```

3.1 Preparing the Enviroment

The R libraries are installed.

Importing data set

```
dataset<- read.csv("dailyActivity_merged.csv",header=TRUE,sep = ",")
```

3.3 Cleaning and modifying data

1 Observe and get acquainted with the data 2 Verify any missing or empty values. 3 Run a sanity check on the data. ##### Previewing using glimpse function on daily_activity to familiarise with the data

```
str(dataset)

## '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 ...
```

Cleaning Column Names

```
dataset<-clean_names(dataset)
colnames(dataset)

##  [1] "id"                         "activity_date"
##  [3] "total_steps"                "total_distance"
##  [5] "tracker_distance"           "logged_activities_distance"
##  [7] "very_active_distance"       "moderately_active_distance"
##  [9] "light_active_distance"      "sedentary_active_distance"
## [11] "very_active_minutes"        "fairly_active_minutes"
## [13] "lightly_active_minutes"     "sedentary_minutes"
## [15] "calories"
```

Finding Out Basic Information of Data

No. of Rows and Columns * Columns Names * Non Null Count * Data Type

Finding Data Type of Each Column

```
str(dataset)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ id                        : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09
1.5e+09 ...
##  $ activity_date             : chr  "4/12/2016" "4/13/2016" "4/14/2016"
"4/15/2016" ...
##  $ total_steps               : int  13162 10735 10460 9762 12669 9705
13019 15506 10544 9819 ...
##  $ total_distance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ tracker_distance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ logged_activities_distance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_distance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ moderately_active_distance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ light_active_distance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ sedentary_active_distance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_minutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ fairly_active_minutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ lightly_active_minutes    : int  328 217 181 209 221 164 233 264 205
211 ...
##  $ sedentary_minutes         : int  728 776 1218 726 773 539 1149 775 818
838 ...
##  $ calories                  : int  1985 1797 1776 1745 1863 1728 1921
2035 1786 1775 ...
```

####Finding Unique ID

```
dataset %>%
  distinct(id)
```

```
##            id
## 1  1503960366
## 2  1624580081
## 3  1644430081
## 4  1844505072
## 5  1927972279
## 6  2022484408
## 7  2026352035
## 8  2320127002
## 9  2347167796
## 10 2873212765
## 11 3372868164
## 12 3977333714
## 13 4020332650
## 14 4057192912
## 15 4319703577
## 16 4388161847
## 17 4445114986
## 18 4558609924
## 19 4702921684
## 20 5553957443
## 21 5577150313
## 22 6117666160
## 23 6290855005
## 24 6775888955
## 25 6962181067
## 26 7007744171
## 27 7086361926
## 28 8053475328
## 29 8253242879
## 30 8378563200
## 31 8583815059
## 32 8792009665
## 33 8877689391
```

Finding Null and Missing Values

```
sum(is.na(dataset))

## [1] 0
```

From the Above observation , we noted that

1. There Zero Null or Missing Values
2. Data has 15 Columns 940 Rows
3. ActivityData is wrongly classified as Object dtype and has to be converted into datatime64 dtype
4. Instead of the predicted 30 unique IDs, there are 33 unique IDs. It's possible that some users made more IDs while the survey was being conducted.

Once the corrupt data has been located, we will manipulate or change the data.

1.Activity Date should be changed to datatime64 dtype. 2.Change Activity Date's format to yyyy-mm-dd. 3.For additional research, create a new column called DayOfTheWeek by producing dates as days of the week. 4.Adding the total of the VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, and SedentaryMinutes columns will give you TotalMins.

By changing the new column TotalMins in number 4 to the number of hours, create a new column called TotalHours. Rename and rearrange the columns.

Prior to converting ActivityDate to yyyy-mm-dd, we will first convert ActivityDate from an object to a datatime64 dtype. Then we check to see whether ActivityDate has changed to yyyy-mm-dd and Datatime64 Dtype.

Converting Activity Date to datetime64 and format to YYYY-MM-DD

```
View(dataset)

dataset$activity_date <- as.Date(dataset$activity_date, format = "%m/%d/%Y")
```

Printing First Ten Rows of Dataset to check the changes

```
head(dataset)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    2016-04-12       13162           8.50             8.50
## 2 1503960366    2016-04-13       10735           6.97             6.97
## 3 1503960366    2016-04-14       10460           6.74             6.74
## 4 1503960366    2016-04-15        9762           6.28             6.28
## 5 1503960366    2016-04-16       12669           8.16             8.16
## 6 1503960366    2016-04-17        9705           6.48             6.48
##   logged_activities_distance very_active_distance
moderately_active_distance
## 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
##   light_active_distance sedentary_active_distance very_active_minutes
## 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
##   fairly_active_minutes lightly_active_minutes sedentary_minutes 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
```

Creating new list

```
activity_Day <- wday(dataset$activity_date, label=TRUE)
dataset['activity_Day']<-activity_Day
totactive_Minutes<-(dataset$very_active_minutes +
dataset$fairly_active_minutes +dataset$lightly_active_minutes
+dataset$sedentary_minutes)
dataset['totactive_Minutes']<-totactive_Minutes
totactive_Hours<-ceiling((totactive_Minutes/60))
dataset['totactive_Hours']<-totactive_Hours
View(dataset)

#mutate(dataset, activity_Hours= (dataset$totactive_Minutes/60))
```

Step-4 : Analyse

4.1 Perform calculation

Pulling Calculation 1. Count - No. of Rows 2. Mean - Average 3. Standard Deviation 4. Min and Max 5.Percentiles 25%, 50%, 75%

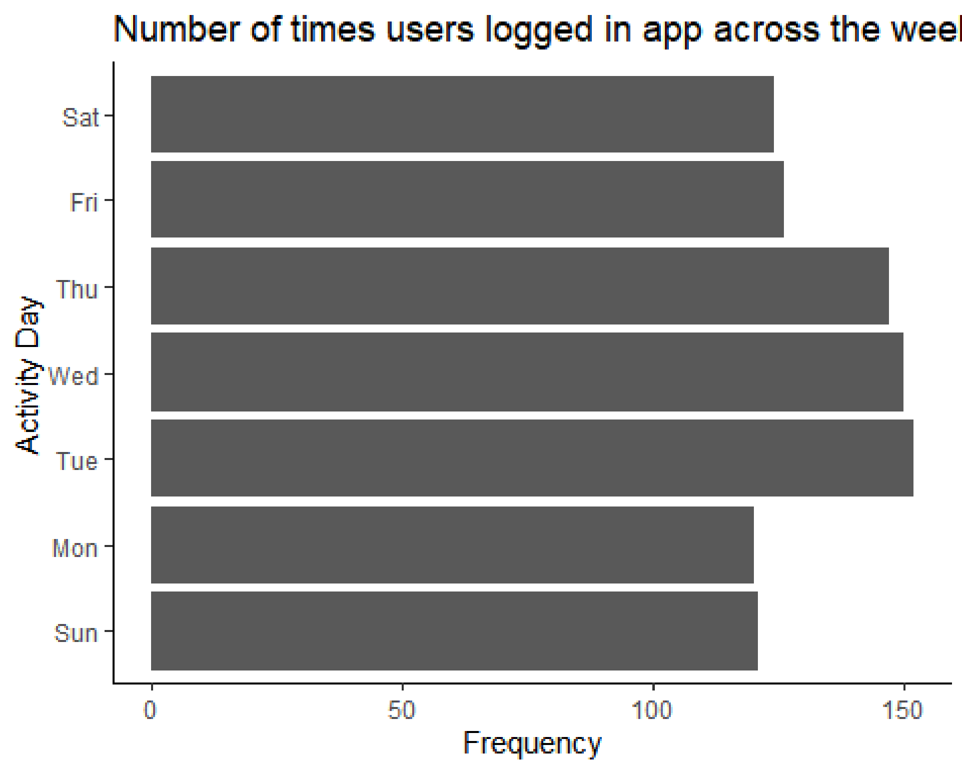
```
summary(dataset)

##        id             activity_date         total_steps    total_distance
##  Min.   :1.504e+09   Min.   :2016-04-12   Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   1st Qu.:2016-04-19   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Median :2016-04-26   Median : 7406   Median : 5.245
##  Mean   :4.855e+09   Mean   :2016-04-26   Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04   3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :36019   Max.   :28.030
##
##  tracker_distance logged_activities_distance very_active_distance
##  Min.   : 0.000   Min.   :0.0000             Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000             1st Qu.: 0.000
##  Median : 5.245   Median :0.0000             Median : 0.210
##  Mean   : 5.475   Mean   :0.1082             Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000             3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421             Max.   :21.920
##
##  moderately_active_distance light_active_distance
sedentary_active_distance
##  Min.   :0.0000             Min.   : 0.000        Min.   :0.000000
##  1st Qu.:0.0000             1st Qu.: 1.945        1st Qu.:0.000000
##  Median :0.2400             Median : 3.365        Median :0.000000
##  Mean   :0.5675             Mean   : 3.341        Mean   :0.001606
##  3rd Qu.:0.8000             3rd Qu.: 4.782        3rd Qu.:0.000000
##  Max.   :6.4800             Max.   :10.710        Max.   :0.110000
##
##  very_active_minutes fairly_active_minutes lightly_active_minutes
##  Min.   :  0.00      Min.   :  0.00        Min.   :  0.0
##  1st Qu.:  0.00      1st Qu.:  0.00        1st Qu.:127.0
##  Median :  4.00      Median :  6.00        Median :199.0
##  Mean   : 21.16      Mean   : 13.56        Mean   :192.8
##  3rd Qu.: 32.00      3rd Qu.: 19.00        3rd Qu.:264.0
##  Max.   :210.00      Max.   :143.00        Max.   :518.0
##
##  sedentary_minutes    calories    activity_Day totactive_Minutes
##  Min.   :   0.0    Min.   :   0   Sun:121      Min.   :   2.0
##  1st Qu.: 729.8    1st Qu.:1828   Mon:120      1st Qu.: 989.8
##  Median :1057.5    Median :2134   Tue:152      Median :1440.0
##  Mean   : 991.2    Mean   :2304   Wed:150      Mean   :1218.8
##  3rd Qu.:1229.5    3rd Qu.:2793   Thu:147      3rd Qu.:1440.0
##  Max.   :1440.0    Max.   :4900   Fri:126      Max.   :1440.0
##                                   Sat:124
##  totactive_Hours
##  Min.   : 1.00
##  1st Qu.:17.00
##  Median :24.00
##  Mean   :20.55
##  3rd Qu.:24.00
##  Max.   :24.00
##
```

Interpreting statistical findings:

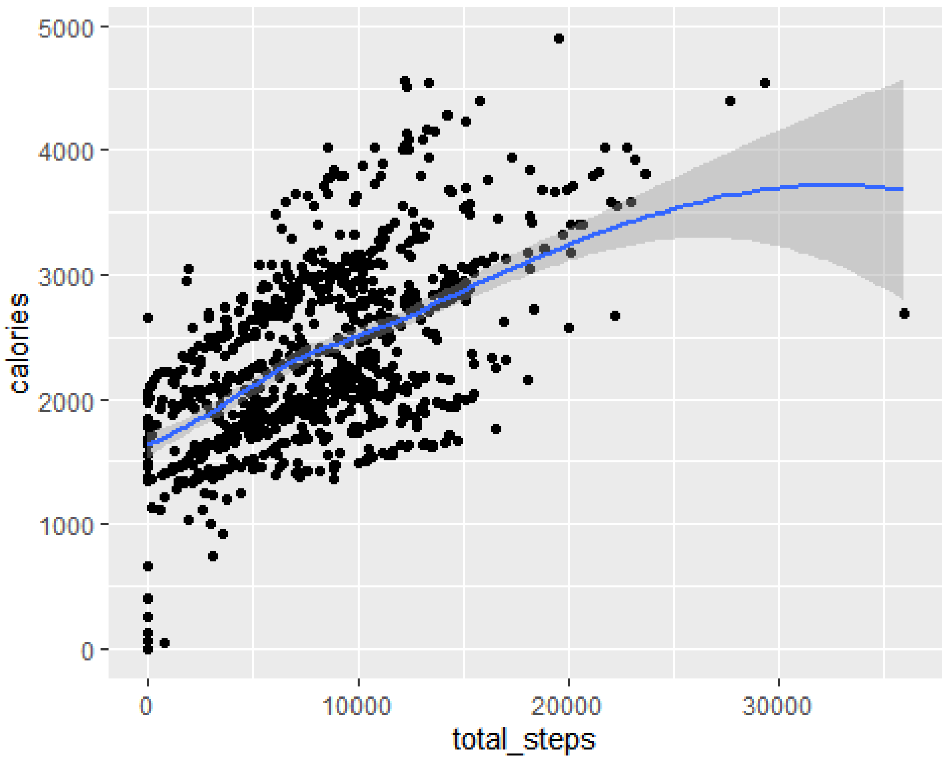
STEP 5: Share

Number of Times users logged in App Across the Week

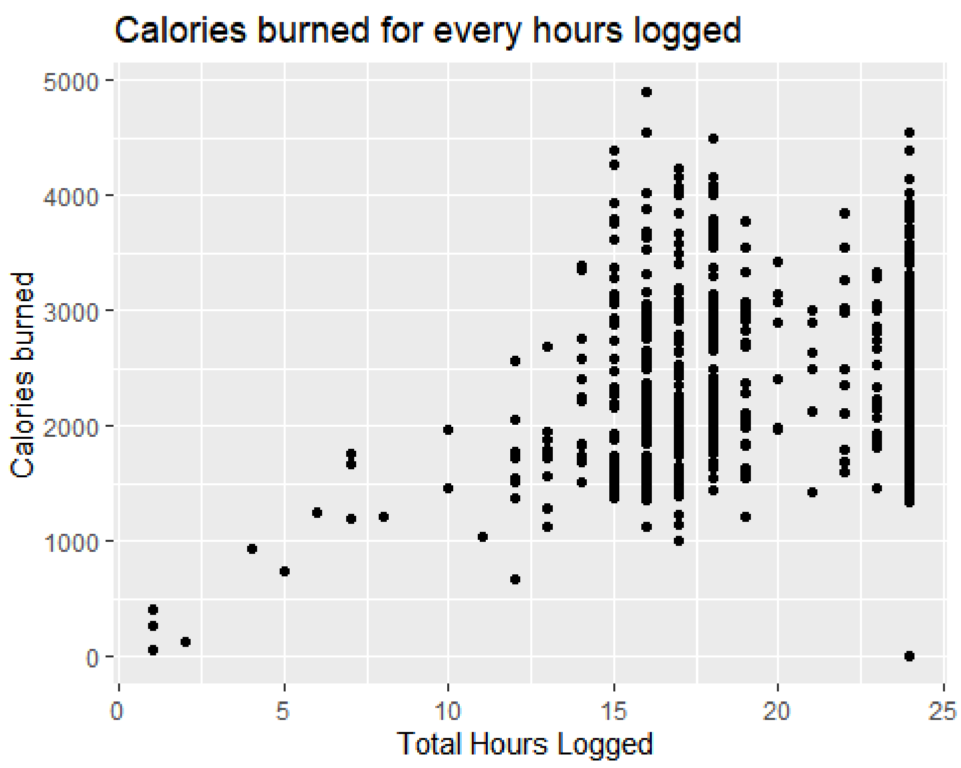
```
p<-ggplot(data=dataset)

p + geom_bar(mapping = aes(x=activity_Day ))+coord_flip() +theme_classic()+
  labs(title="Number of times users logged in app across the week", x =
"Activity Day",y="Frequency")
```



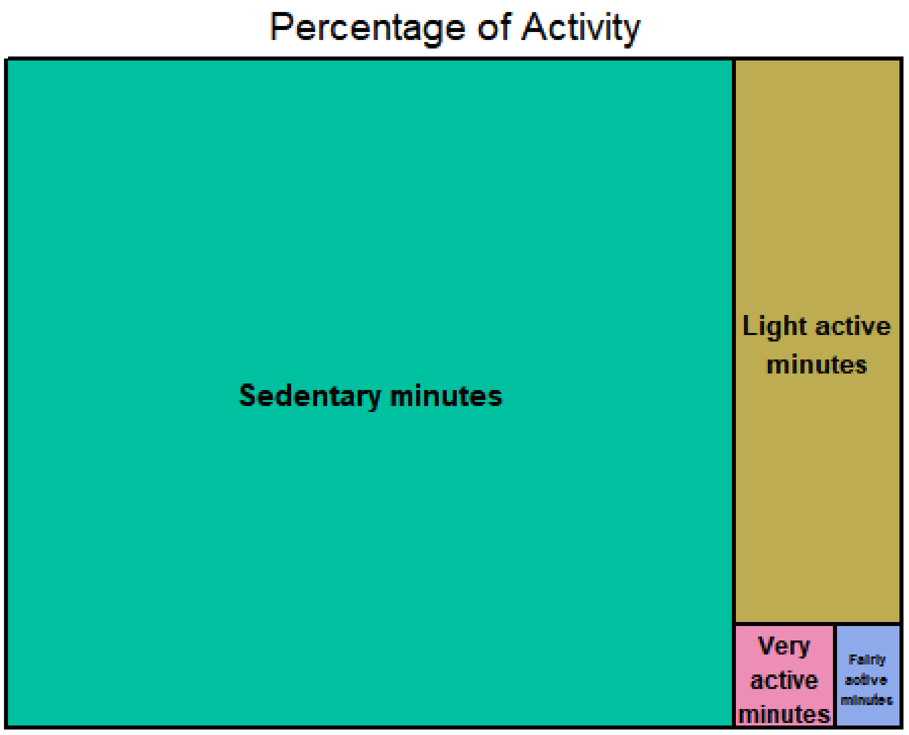
```
p + geom_point(mapping = aes(x= total_steps,y=calories)) +
  geom_smooth(mapping=aes(x=total_steps,y=calories))

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



```
  labs(title="Calories burned for every step taken", x = "Steps
taken",y="Calories burned")

## $x
## [1] "Steps taken"
##
## $y
## [1] "Calories burned"
##
## $title
## [1] "Calories burned for every step taken"
##
## attr(,"class")
## [1] "labels"

p + geom_point(mapping = aes(x = totactive_Hours,y=calories)) +
  labs(title="Calories burned for every hours logged", x = "Total Hours
Logged ", y="Calories burned")
```



```
value<-c(sum(dataset$lightly_active_minutes) ,
sum(dataset$fairly_active_minutes), sum(dataset$very_active_minutes),
sum(dataset$sedentary_minutes))
group <- c(" Light active minutes" ,"Fairly active minutes" , "Very active
minutes" , "Sedentary minutes")
activity_data <- data.frame(group,value)
treemap( activity_data,
        index="group",
        vSize = "value",
        type = "index",
        title="Percentage of Activity"
        )
```


Percentage of Activity
Sedentary minutes
Light active minutes
Very active minutes
Fairly active minutes