# Bellabeat Case Study

adetunji daniel

June 22, 2023

## How can a wellness Technology Company Play it Smart?

**INTRODUCTION**

Bellabeat is a high-tech manufacturer of health-focused products for women. Urska Srsen, co-founder of bellabeat,believes that analyzing smart device fitness data could help unlock new growth opportunities fot the company. i have been tasked to focus on one of bellabeat's products and analyze smart device data to gain insight into how consumers are using smart devices.

To complete this task, i will follow the steps of data analysis process:

**ask**, **prepare**, **process**, **analyze**, **share** and **act**

## ASK

### 1. Who are the stakeholders?

- Urska Srsen: Bellabeat's cofounder and Chief Creative Officer
- Sando Mur: Mathematician, Bellabeat's cofounder and key member of the Bellabeat executive team
- Bellabeat marketing analytics team: A team of data analysts guiding Bellabeat's marketing strategy.

### 2. What are the Business Objectives?

- What are the trends identified?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy?

## PREPARE

### Where is the data stored? or source of data

- The data is publicly available on Kaggle: FitBit Fitness Tracker Data and stored in 18 csv file
- 30 FitBit users who consented to the submission of personal tracker data

**Limitations of data**

1. Sampling Bias : The user information of just 30 Fitbit users is not representative of the entire female population

2. Data credibility : is the data?

- Reliable? NO - as it only has 30 respondents
- Original? NO - as it is gotten from a third party provider (Amazon Mechanical Turk)
- Comprehensive? YES - most parameters needed are given
- Current? NO - the data is outdated as it is 7 years old
- Cited? NO - as it is a third party data hence unknown.

## PROCESS

For this analysis R is used to clean, analyze and visualize the data

### Setting up the tool

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----------------------- tidyverse
2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts -----------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(skimr)
```

### Importing the data set

```
daily_activity <-
read.csv("file:///C:/Users/NEWGENERATION/Desktop/dailyActivity_merged.csv")
```

```
sleep_day <-
read.csv("file:///C:/Users/NEWGENERATION/Desktop/sleepDay_merged.csv")
```

*having a quick glance at the data*

```
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
```

```
head(sleep_day)
```

```
##           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
```

## Data cleaning and Manipulation

- Checking for null or missing values

```
sum(is.null(daily_activity))
```

```
## [1] 0
```

```
sum(is.null(sleep_day))
```

```
## [1] 0
```

The two dataframes contains no missing values

- checking for duplicates values

```
sum(duplicated(daily_activity))
```

```
## [1] 0
```

```
sum(duplicated(sleep_day))
```

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

the dailyactivity dataframe has no duplicates value, but there are 3 duplicates value in the sleepday dataframe, so our next step is to remove the duplicates value

```
sleep_day <- distinct(sleep_day)
```

the duplicates value has been removed, so let's confirm the removal of the duplicates value.

```
sum(duplicated(sleep_day))
```

```
## [1] 0
```

- checking for number of distinct Id's

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

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

```
n_distinct(sleep_day$Id)
```

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

there are 33 and 24 unique id's respectively indicating irregularity of the dataset as we are expected to have 30 distinct id's.

- taking a look at the data frame column names

```
colnames(daily_activity)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
```

```
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
colnames(sleep_day)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"
```

- checking the column names are of correct datatypes

```
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 ...
```

```
str(sleep_day)
```

```
## 'data.frame':    410 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" "4/16/2016 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 ...
```

The ActivityDate column under the daily_activity data frame and the SleepDay column under sleep_day data frame were stored as character, so there is need to change them to date type and datetime type respectively

- correcting the wrong data type

```
daily_activity <- daily_activity %>%
  mutate(ActivityDate = mdy(ActivityDate))
```

```
sleep_day <- sleep_day %>%
  mutate(SleepDay = mdy_hms(SleepDay))
```

- confirming the dataframes are of correct data types

```
str(daily_activity$ActivityDate)
```

```
##  Date[1:940], format: "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15"
"2016-04-16" ...
```

```
str(sleep_day$SleepDay)
```

```
##  POSIXct[1:410], format: "2016-04-12" "2016-04-13" "2016-04-15" "2016-04-
16" "2016-04-17" ...
```

- Quick statistical view of the users activities

```
summary(daily_activity)
```

```
##        Id              ActivityDate          TotalSteps    TotalDistance
##  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
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  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
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  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
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
```

```
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

```
 summary(sleep_day)
```

```
##        Id               SleepDay                    TotalSleepRecords
##  Min.   :1.504e+09   Min.   :2016-04-12 00:00:00   Min.   :1.00
##  1st Qu.:3.977e+09   1st Qu.:2016-04-19 00:00:00   1st Qu.:1.00
##  Median :4.703e+09   Median :2016-04-27 00:00:00   Median :1.00
##  Mean   :4.995e+09   Mean   :2016-04-26 11:38:55   Mean   :1.12
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04 00:00:00   3rd Qu.:1.00
##  Max.   :8.792e+09   Max.   :2016-05-12 00:00:00   Max.   :3.00
##  TotalMinutesAsleep TotalTimeInBed
##  Min.   : 58.0      Min.   : 61.0
##  1st Qu.:361.0      1st Qu.:403.8
##  Median :432.5      Median :463.0
##  Mean   :419.2      Mean   :458.5
##  3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :796.0      Max.   :961.0
```

### *OBSERVATIONS*

1. the average number of steps taken is 7638 in a day and an average distance of 5.5 km which is below the recommended 10,000 steps or 8km per day.
2. the users burnt an average of 2304 calories in a day
3. on average users sleep for about 419 minutes, approximately 7hours in a day
4. majority of the users prefers sedentary activities as the average sedentary minutes is 991 minutes approximately 20 hours.
5. It is not surprising that the total time on bed is greater than the total time asleep.

### *Transforming And Manipulating The DataFrames*

The following data Manipulation is performed :

1. create new column *weekdays* by extracting the day of the week from date column for further analysis
2. create new coulumn *totalminutes* by summing up : VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes and SedentaryMinutes.
3. create a new dataframe for the daily_activity containing just (Id,weekdays,TotalSteps,TotalDistance,totalminutes,Calories) columns and a new data frame for the sleep_day containg just (Id, weekdays, TotalMinutesAsleep,TotalTimeInBed)

```
 daily_activity <- daily_activity %>%
  mutate(weekdays= weekdays(ActivityDate))

 daily_activity <- daily_activity %>%
  mutate(totalminutes =
VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes+SedentaryMinutes)
```

```
daily_activity <- daily_activity %>%
  select(Id,weekdays,TotalSteps,TotalDistance,totalminutes,Calories)

sleep_day <- sleep_day %>%
  mutate(weekdays = weekdays(SleepDay))

sleep_day <- sleep_day %>%
  select(Id, weekdays, TotalMinutesAsleep,TotalTimeInBed)

head(daily_activity)
```

```
##           Id  weekdays TotalSteps TotalDistance totalminutes Calories
## 1 1503960366   Tuesday      13162          8.50         1094     1985
## 2 1503960366 Wednesday      10735          6.97         1033     1797
## 3 1503960366  Thursday      10460          6.74         1440     1776
## 4 1503960366    Friday       9762          6.28          998     1745
## 5 1503960366  Saturday      12669          8.16         1040     1863
## 6 1503960366    Sunday       9705          6.48          761     1728
```

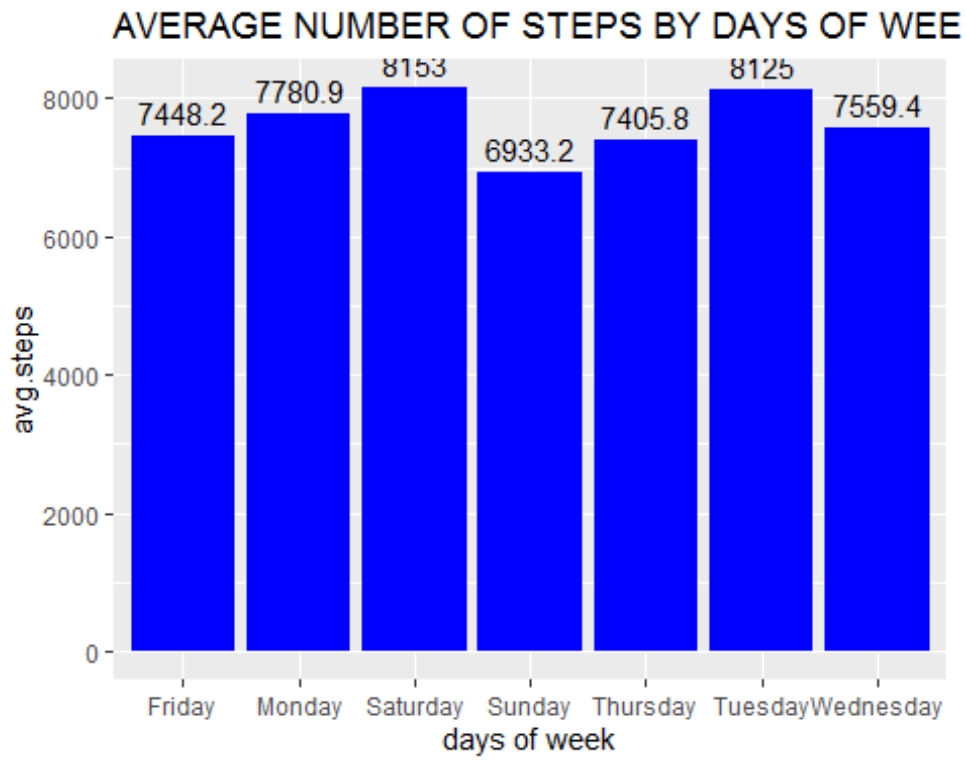
```
head(sleep_day)
```

```
##           Id  weekdays TotalMinutesAsleep TotalTimeInBed
## 1 1503960366   Tuesday                327            346
## 2 1503960366 Wednesday                384            407
## 3 1503960366    Friday                412            442
## 4 1503960366  Saturday                340            367
## 5 1503960366    Sunday                700            712
## 6 1503960366   Tuesday                304            320
```

## ANALYZE AND SHARE

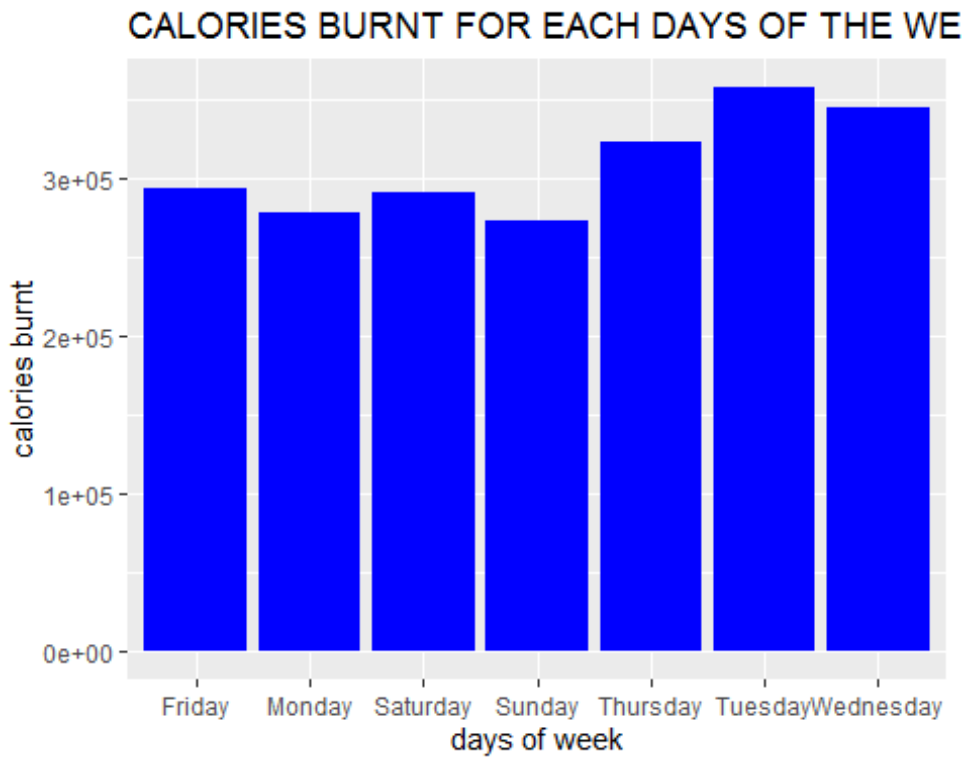
**In this phase i will be analyzing and visualizing the data to communicate my findings and observations.

```
averagesteps <- daily_activity %>%
  group_by(weekdays) %>%
  summarise(averagesteps =round(mean(TotalSteps),1)) %>%
  arrange(averagesteps)

ggplot(data = averagesteps) +
  geom_bar(mapping = aes(x = weekdays, y = averagesteps), fill = 'blue', stat
= "identity") +
  geom_text(aes(x = weekdays,y = averagesteps,label = averagesteps),vjust = -
0.5) +
  labs(x= "days of week", y = "avg.steps", title = "AVERAGE NUMBER OF STEPS
BY DAYS OF WEEKS")
```



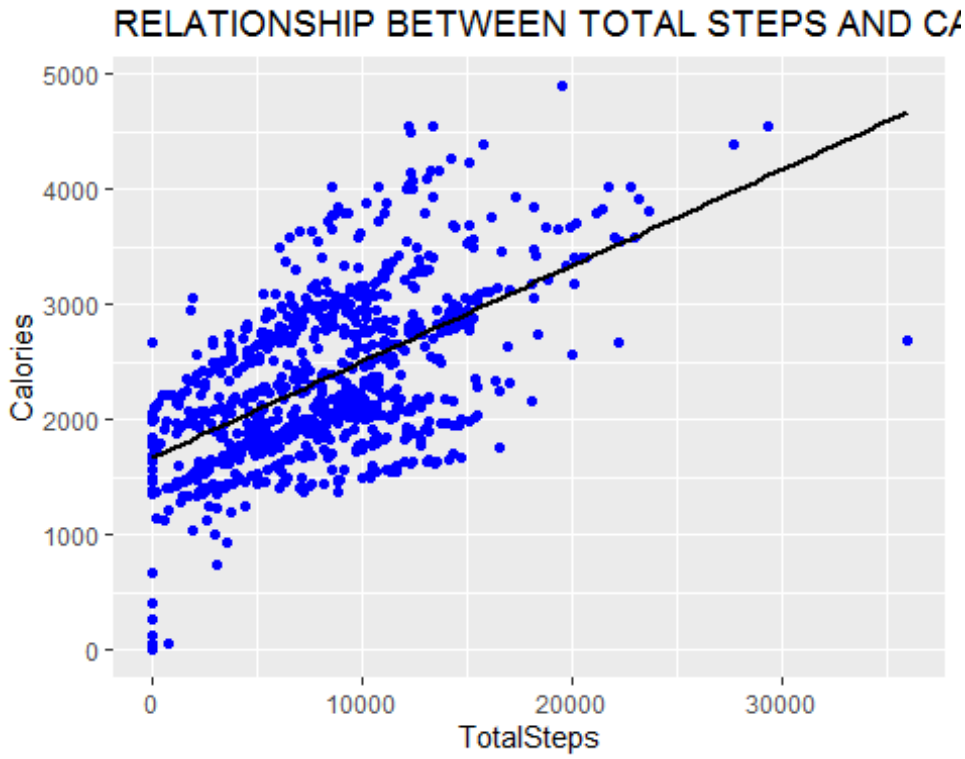

The above charts shows the average number of steps taken by users for each day of the week, and it is observed that users are more active on Tuesday and Saturday and less active on sunday.

```
caloriesburnt <- daily_activity %>%
   group_by(weekdays) %>%
   summarise(totalcalories = sum(Calories))

ggplot( data = caloriesburnt) +
  geom_bar(mapping = aes(x= weekdays, y = totalcalories),fill = 'blue', stat
="identity") +
  labs(x = "days of week" , y = "calories burnt", title = "CALORIES BURNT FOR
EACH DAYS OF THE WEEK")
```

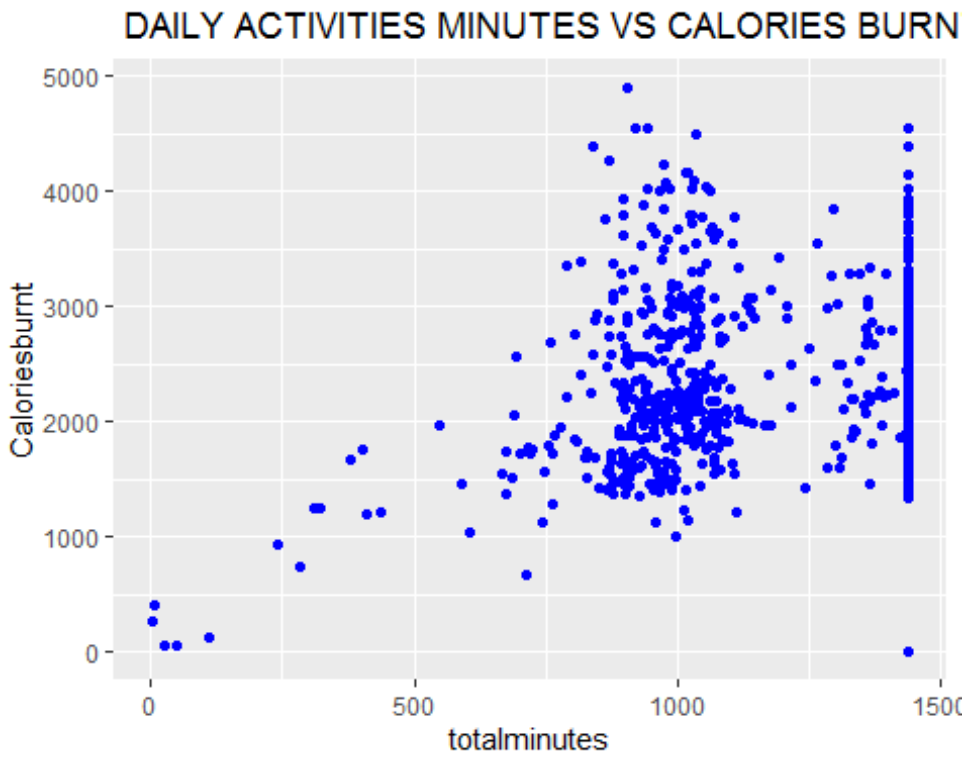



the chart above shows the total number of calories burnt for each day of the week and tuesday being the day with the highest number of burnt calories.

```
 ggplot(data = daily_activity) +
    geom_point(mapping = aes(x = TotalSteps, y = Calories),color = 'blue') +
    geom_smooth(aes(x =TotalSteps, y = Calories),color = 'black',method
="lm",se=FALSE)+
    labs(title = "RELATIONSHIP BETWEEN TOTAL STEPS AND CALORIES BURNT")

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

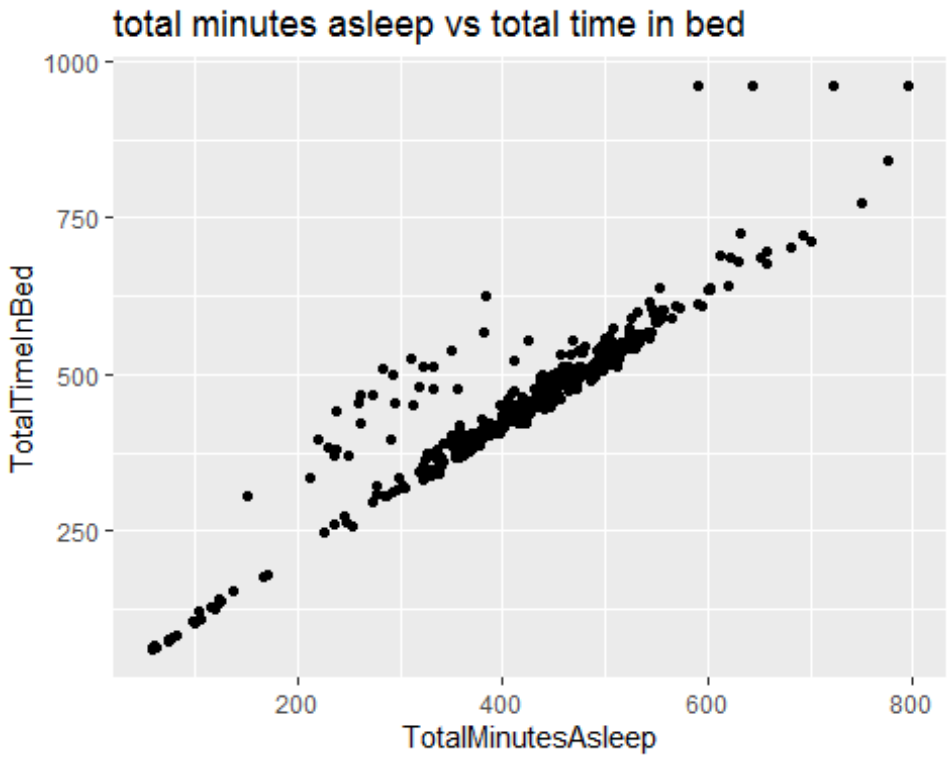



```
ggplot(data = daily_activity) +
    geom_point(mapping = aes(x = totalminutes, y = Calories), color = 'blue')
+
    labs(title = " DAILY ACTIVITIES MINUTES VS CALORIES BURNT", X =" daily
minutes",y = "Caloriesburnt")
```




The plots above shows there is a positive correlation between the user activities and calories burnt, the greater the steps taken the more calories users burn and the greater the time spent on activities the more calories they burn.

```
ggplot(data = sleep_day) +
   geom_point(mapping = aes(x = TotalMinutesAsleep, y = TotalTimeInBed)) +
   labs(title = "total minutes asleep vs total time in bed")
```




this scatterplots shows a positive correlation between the two variables indicating that users are usually in bed only when they are asleep. so they don't stay much time awake in bed before sleeping.

## ACT

Here we will be reviewing our business objectives and delivering recommendations based on the insight of our analysis.

1.**What are the trends identified?**

- with an average number of 7638steps (5.5km) taken per day users are yet to meet up with the recommended number of steps to be taken.
- majority of the users spend more time being sedentary
- Users burnt more calories on weekdays than on weekends, with sunday being the less active day.

2. **How could these trends apply to Bellabeat customers?**

- The trends can actually help bellabeat understand user behaviours and tailor their products and services to meet customers need effectively.

3. **How could these trends help influence Bellabeat marketing strategy?**

- Bellbeat app should include features like activity reminder to encourage users to spend less time being sedentary.

- Educate users on the relationship between the number of steps taken and calories burn, this is to encourage them to make use of the bellabeat app to track thier steps.

*NOTE :*

Due to the limitations of the data, bellabeat should obtain more data for an accurate analysis.