Case Study

Shanga Saadallah

2023-02-03

##Task statement Key stakeholders Urska Srsen Sando mur Finding trends in smart device usage, how could these apply to bellabeat and how they can help the marketing strategy for bellabeat products.

Prepare data

Data source: kaggle Data format: Long format data Are data ROCCC: Reliability: This dataset is under CC0: Public Domain license meaning the creator has waive his right to the work under the copyright law. ## ROCCC analysis Reliability: LOW – dataset was collected from 30 individuals whose gender is unknown. Originality : LOW – third party data collect using Amazon Mechanical Turk. Comprehensive : MEDIUM – dataset contains multiple fields on daily activity intensity, calories used, daily steps taken, daily sleep time and weight record. Current : MEDIUM – data is 5 years old but the habit of how people live does not change over a few years Cited : HIGH – data collector and source is well documented ## Data Selection The data needed are daily usage of the smart device therefore there will be only the following files selected DailyActivities_merged.csv DailyCalories_merged.csv DailyIntensities_merged.csv DailySteps_merged.csv SleepDaily_merged.scv WeighLogInfo_merged.csv ## Prepare the data The data used in this analysis is the Fitbit Fitness Tracker Data made available by Mobius stored on Kaggle. This dataset is under CC0: Public Domain license meaning the creator has waive his right to the work under the copyright law. The data 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. These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. The dataset has in total 18 files in .csv format organized in long format Process the data Google spreadsheet and Google BigQuery will be used to process the data as the tool functionality fits the purpose

Data cleaning

sleepDay_merged.csv and weightLogInfo_merged.csv are loaded into Google Sheets for data cleaning. The fields "SleepDay" and "Date" were not correctly formatted. The following steps have been done: The date column has been selected and formatted to 'Date' using the spreadsheet function Time in the column has been removed as time is irrelevant in this analysis AM/PM indicator is also removed. ## Data integrity The selected data has been loaded into Google BigQuery for analysis. The following queries have been run to check the number of unique Id in each table:

SELECT DISTINCT Id FROM `my-capstone-case-study.smart_device.daily_activities` SELECT DISTINCT Id FROM `my-capstone-case-study.smart_device.daily_calories` SELECT DISTINCT Id FROM`my-capstone-case-study.smart_device.daily_intensities' SELECT DISTINCT Id FROM`my-capstone-case-study.smart__device.daily__steps SELECT DISTINCT Id FROM `my-capstone-case-study.smart_device.sl SELECT DISTINCT Id FROM`my-capstone-case-study.smart__device.weight__log'

Result(distinct Id in each table): 33 33 33 33 24 8 The result shows the dataset is inconsistent as we expect 30 unique IDs on all tables. The sleepDay_merged table and the weigthLogInfo_merged table have the highest inconsistencies with 6 and 22 inputs missing. This would affect the result of the analysis.

Data analysis

The hypothesis has been made with the data available on activity, sleep time, and weight. 1-There is a relationship between activity level and calories burnt. 2-There is a relationship between activity level and sleep time 3-There is a relationship between activity level and weight In order to find out the relation and validate the hypothesis, four queries have been constructed to aggregate the data for analysis.

For finding activity level and calories burnt data in Bigquery

SELECT Id, ActivityDate, Calories, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDistance, (VeryActiveDistance+ModeratelyActiveDistance+LightActiveDistance) AS TotalActiveDistance, SedentaryActiveDistance, (VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes) AS TotalActiveMinutes, SedentaryMinutes

FROM `my-capstone-case-study.smart_device.daily_activity`

For finding relationship between activity level and sleep time

SELECT
Activity.Id, ActivityDate, TotalDistance, TotalMinutesAsleep, TotalTimeInBed, TotalSteps, VeryActiveDistance, ModeratelyActiveDistance, VeryActiveMinutes, FairlyActiveMinutes, LightActiveDistance, SedentaryActiveDistance, LightlyActiveMinutes, SedentaryMinutes FROM `my-capstone-case-study.smart_device.daily_acti`
As Activity INNER JOIN `my-capstone-case-study.smart_device.sleep_day` AS Sleep ON Activity.Id = Sleep.Id AND Activity.ActivityDate = Sleep.SleepDay

For finding relationship between activiy and weight/BMI

SELECT

Activity.Id, ActivityDate, TotalDistance, TotalSteps, BMI, VeryActiveDistance, ModeratelyActiveDistance, VeryActiveMinutes, FairlyActiveMinutes, LightActiveDistance, SedentaryActiveDistance, LightlyActiveMinutes, SedentaryMinutes

FROM `my-capstone-case-study.smart_device.daily_activity` As Activity INNER JOIN `my-capstone-case-study.sm`
AS Weight ON Activity.Id = Weight.Id AND Activity.ActivityDate = Weight.Date ### Analysis

Load the packages.

```
#install.packages("tidyverse", lib="C:/Users/Shange/AppData/Local/R/win-library/4.2")
#install.packages('skimr')
#install.packages('cowplot')
#install.packages("plotly")
#install.packages("ggplot2", lib="C:/Users/Shange/AppData/Local/R/win-library/4.2")
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## 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
```

```
library(tidyverse) #wrangle data
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks plotly::filter(), stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr) #clean data
library(lubridate)  #wrangle date attributes
```

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

```
library(skimr) #get summary data
library(ggplot2) #visualize data
library(cowplot) #grid the plot
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(readr) #save csv
```

Prepaire the data

```
setwd("D:/R/BellaBeat-Device")
daily_df <- read.csv("daily_activity.csv")
sleep_day <- read.csv("activity_sleep.csv")
weight <- read.csv("activity_weight.csv")
intensities <- read.csv("daily_intensities.csv")

daily_df <- daily_df%>%
  mutate( Weekday = weekdays(as.Date(ActivityDate, "%Y-%m-%d")))
sleep_day <- sleep_day%>%
  mutate( Weekday = weekdays(as.Date(ActivityDate, "%m/%d/%Y")))
merged_data <- merge(daily_df, weight, by = c("Id"), all=TRUE)
merged_data <- merged_data %>%
  mutate( Weekday = weekdays(as.Date(ActivityDate.x, "%m/%d/%Y")))
sleep <- read.csv("sleep.csv")
weight_log <- read.csv("weight_log.csv")
set1 <- merge(daily_df, weight_log, by = c("Id"), all=TRUE)
set2 <- merge(sleep, set1, by = c("Id"), all=TRUE)
awake <- mutate(set2, AwakeTime = TotalTimeInBed -TotalMinutesAsleep)
```

Analysis

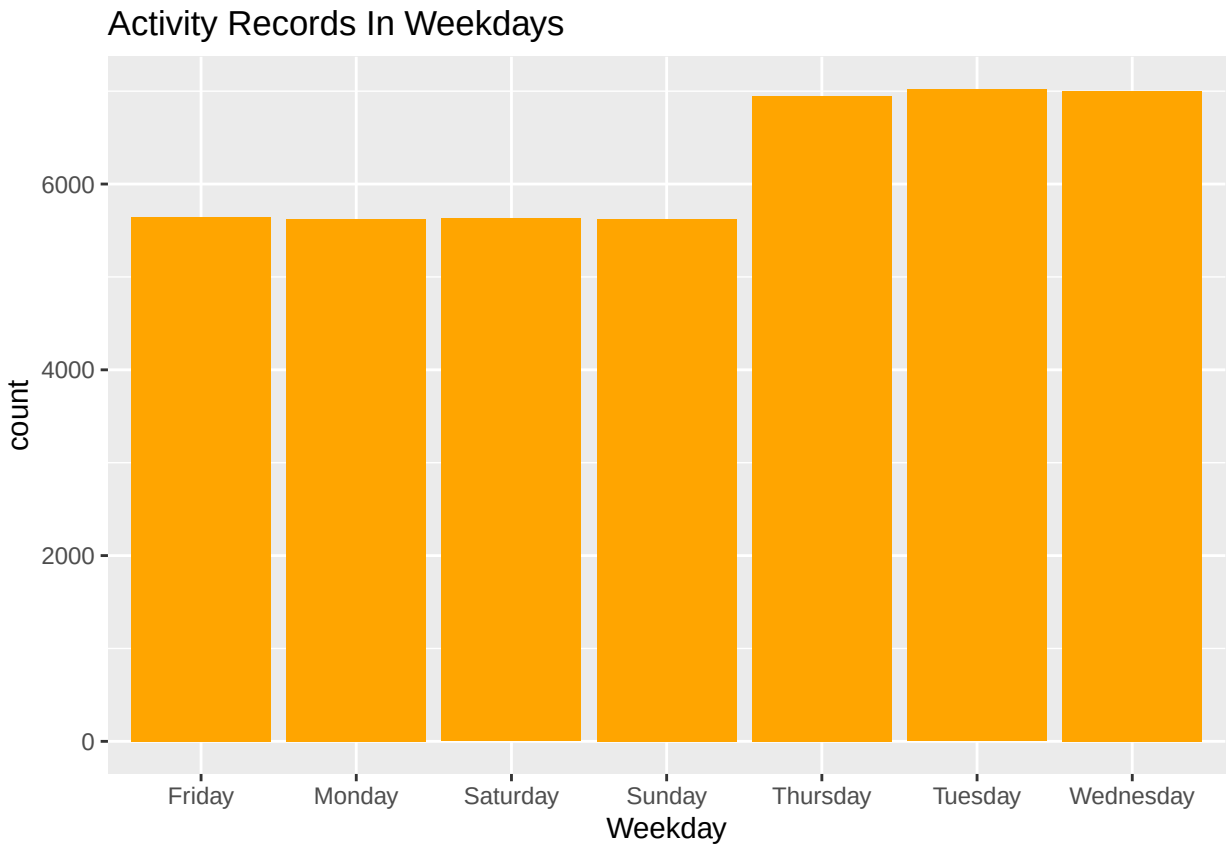
```
set2 %>%
  dplyr::select(Weekday,
                TotalSteps,
                TotalDistance,
                VeryActiveMinutes,
                FairlyActiveMinutes,
                LightlyActiveMinutes,
                SedentaryMinutes,
                Calories,
                TotalMinutesAsleep,
                TotalTimeInBed,
                WeightPounds,
                BMI
  ) %>%
  summary()
```

```
##    Weekday            TotalSteps    TotalDistance    VeryActiveMinutes
##  Length:43482       Min.   :    0   Min.   : 0.000   Min.   :  0.00
##  Class :character   1st Qu.: 5899   1st Qu.: 3.910   1st Qu.:  0.00
##  Mode  :character   Median :10199   Median : 6.820   Median : 15.00
##                     Mean   : 9373   Mean   : 6.417   Mean   : 23.59
##                     3rd Qu.:12109   3rd Qu.: 8.350   3rd Qu.: 38.00
##                     Max.   :36019   Max.   :28.030   Max.   :210.00
##
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:  3.00      1st Qu.:194.0        1st Qu.: 637.0   1st Qu.:1850
##  Median : 14.00      Median :238.0        Median : 697.0   Median :2046
##  Mean   : 17.82      Mean   :232.2        Mean   : 722.7   Mean   :2106
```

```
##  3rd Qu.: 31.00      3rd Qu.:288.0        3rd Qu.: 745.0   3rd Qu.:2185
##  Max.   :143.00      Max.   :518.0        Max.   :1440.0   Max.   :4900
##
##  TotalMinutesAsleep TotalTimeInBed   WeightPounds        BMI
##  Min.   : 58.0      Min.   : 61.0   Min.   :116.0   Min.   :21.45
##  1st Qu.:400.0      1st Qu.:421.0   1st Qu.:134.9   1st Qu.:23.89
##  Median :442.0      Median :457.0   Median :135.6   Median :24.00
##  Mean   :433.9      Mean   :458.3   Mean   :139.6   Mean   :24.42
##  3rd Qu.:477.0      3rd Qu.:510.0   3rd Qu.:136.7   3rd Qu.:24.21
##  Max.   :796.0      Max.   :961.0   Max.   :294.3   Max.   :47.54
##  NA's   :971        NA's   :971     NA's   :8974    NA's   :8974
```

The activities recorded were mostly concentrated in the days of tuesday, wednesday, and thursday. The records show that weekends are not the best activity days

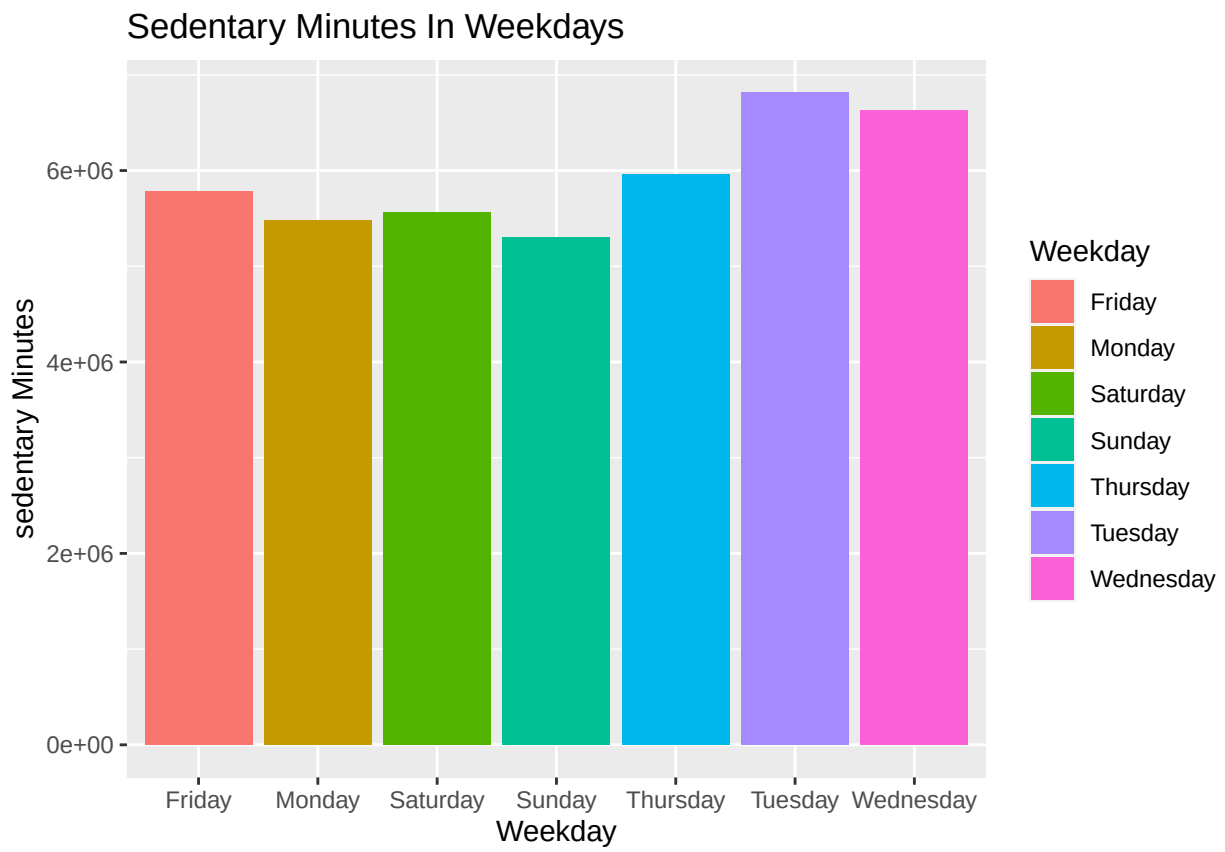
```
library(ggplot2)
library(tidyverse)
ggplot(data=awake, aes(x=Weekday))+
  geom_bar(fill="Orange") +
  labs(title="Activity Records In Weekdays")
```



The sedentary minutes are highest on days of Wednesday, Tuesday, and Thursday. The least sedentary minutes are recorded on Sunday and the highest on Tuesday.

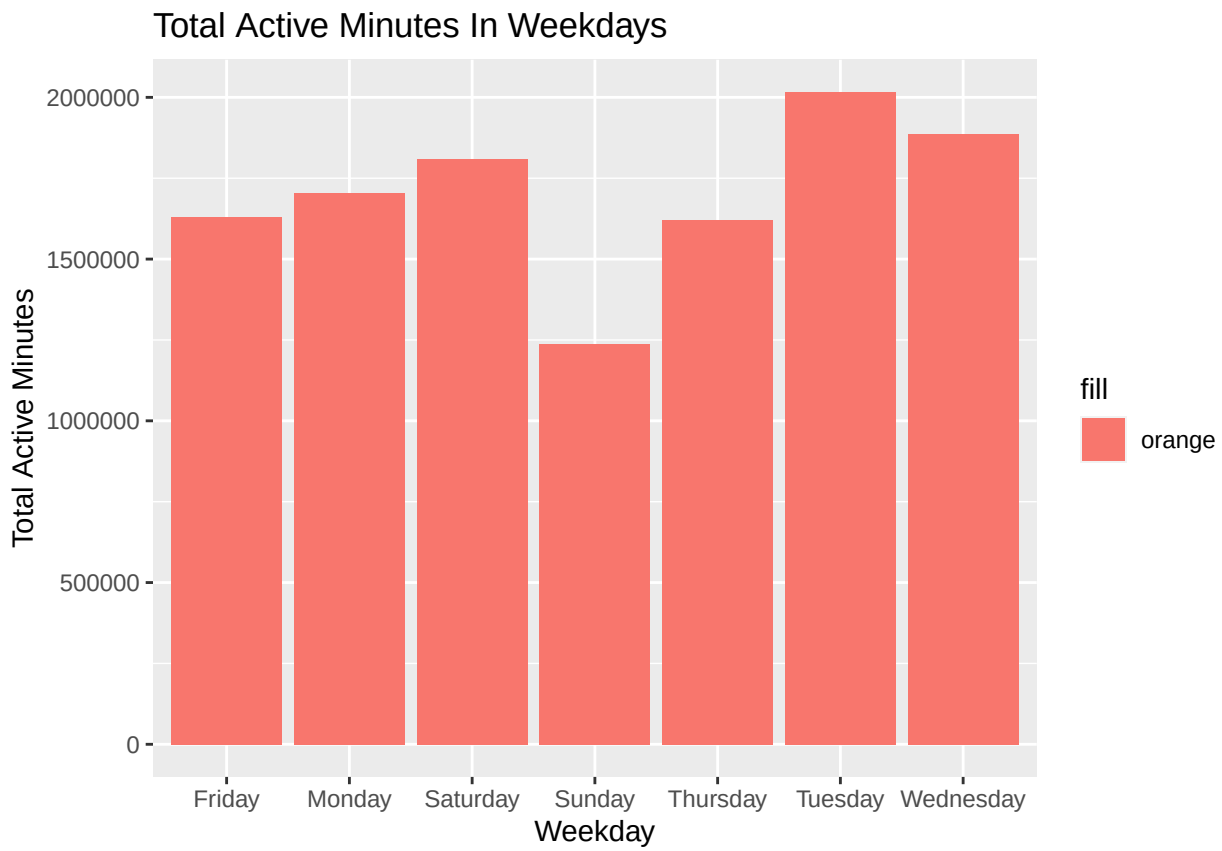
```
library(ggplot2)
ggplot(data=awake, aes(x=Weekday, y=SedentaryMinutes+LightlyActiveMinutes, fill=Weekday))+
```

```
  geom_bar(stat="identity")+
  ylab("sedentary Minutes")+
  labs(title="Sedentary Minutes In Weekdays")
```



The activity level varies from day to day, they tend to increase on Tuesday, Monday, and Wednesday. The activity level jumps down on Sunday and Friday.

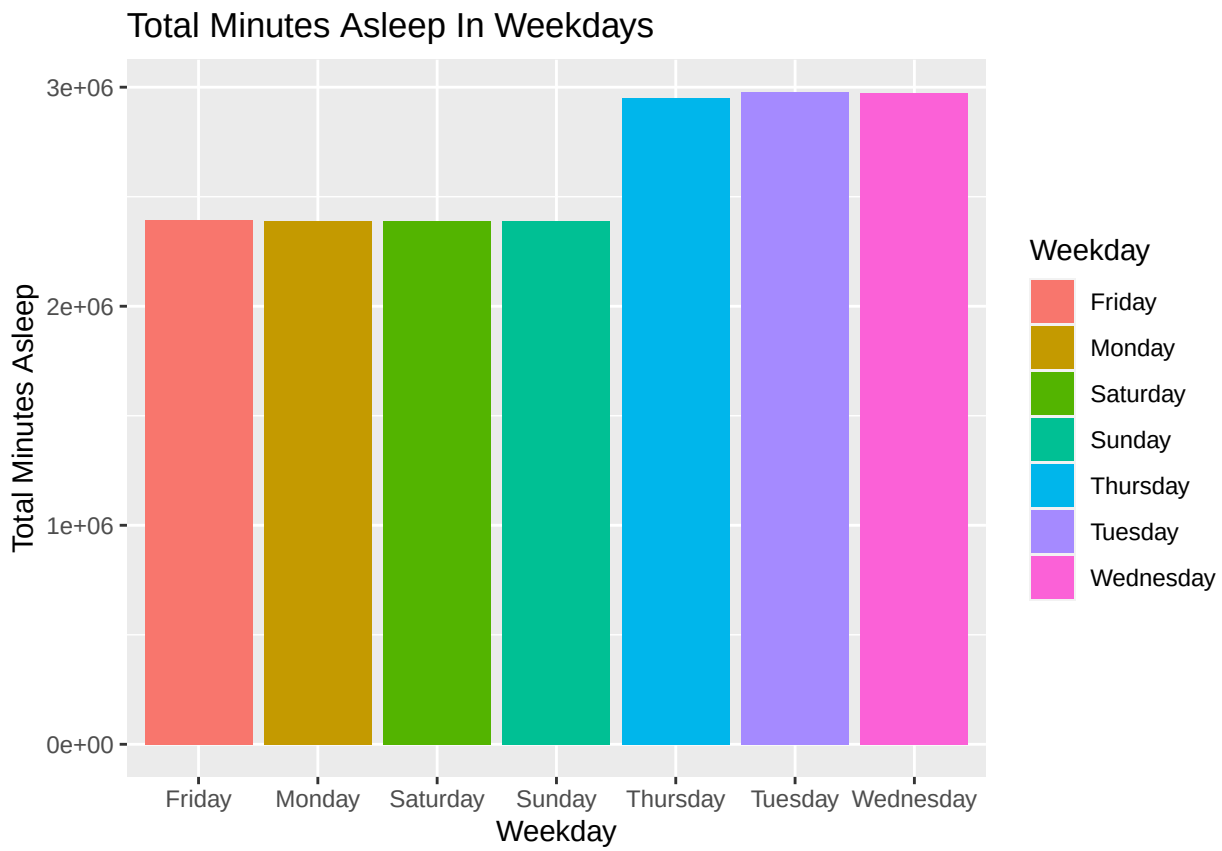
```
ggplot(data=awake, aes(x=Weekday, y=VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes, fill="o
  geom_bar(stat="identity")+
  ylab("Total Active Minutes")+
  labs(title="Total Active Minutes In Weekdays")
```



The user's sleep was best on Tuesday, Thursday, and Wednesday, which is similar to the data records and this raised the question "how comprehensive are the data to form an accurate analysis?"

```
ggplot(data=awake, aes(x=Weekday, y=TotalMinutesAsleep, fill=Weekday))+
  geom_bar(stat="identity")+
  ylab("Total Minutes Asleep")+
  labs(title="Total Minutes Asleep In Weekdays")
```

```
## Warning: Removed 971 rows containing missing values (`position_stack()`).
```



Searching more in the data and the relation between calories and total steps shows that users who recorded total steps of 5000 to 15000 are burning calories from 1000 to 3000 calories

```
#install.packages("ggpmisc")
library(ggpmisc)
```

```
## Loading required package: ggpp
```

```
##
## Attaching package: 'ggpp'
```

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

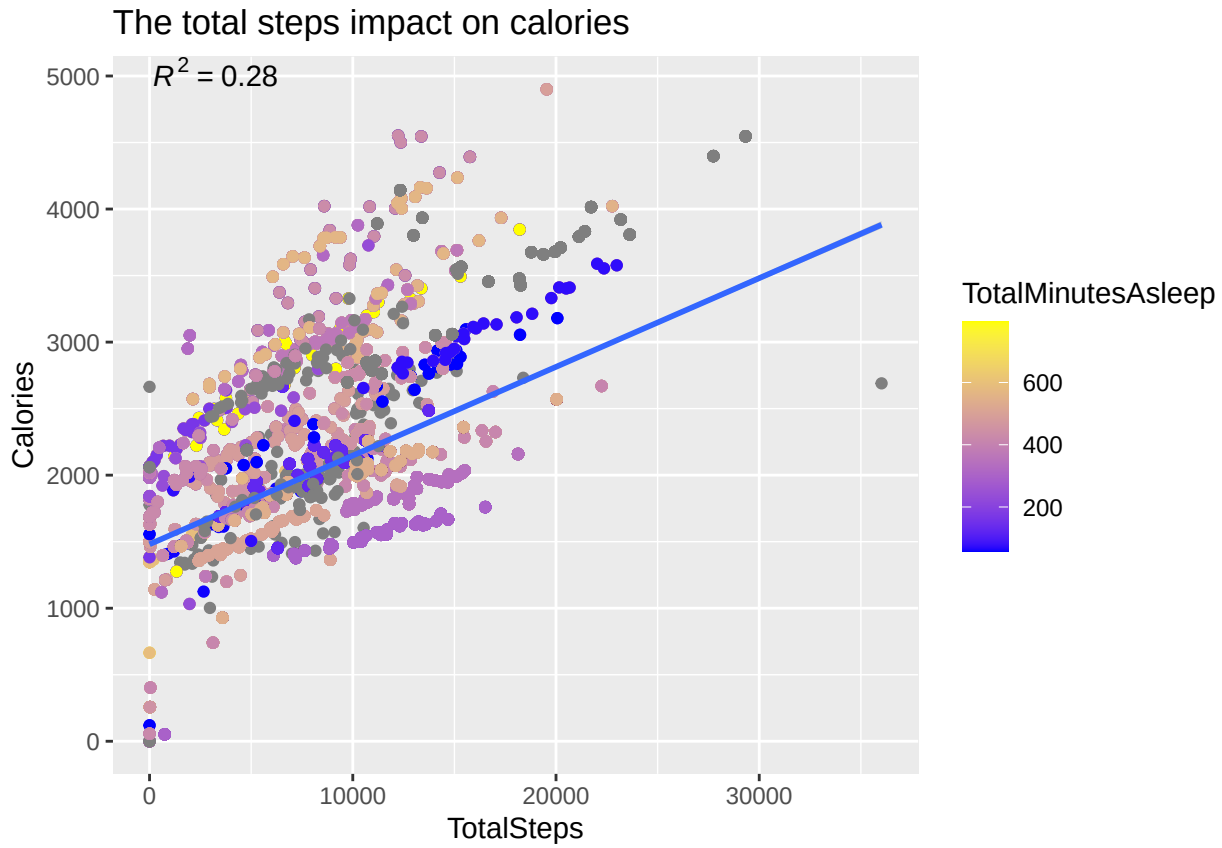
```
par(mfrow = c(2, 2))
ggplot(data=awake, aes(x=TotalSteps, y=Calories, color=TotalMinutesAsleep))+
  geom_point()+
  stat_smooth(method=lm)+
  scale_color_gradient(low="blue", high="yellow")+
  labs(title="The total steps impact on calories")+
  stat_poly_eq(label.y = 400, aes(label = ..rr.label..))
```

```
## Warning: The dot-dot notation (`..rr.label..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(rr.label)` instead.
```

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

## Warning: The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?

## Warning: The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



Comparing the three active levels to the total steps, we see most data is concentrated on users who burn about 1500 to 3500 calories have most recorded data. Despite that users who even have light active minutes do burn calories but The very active Minute users are increasingly burning more calories and that tends to increase.

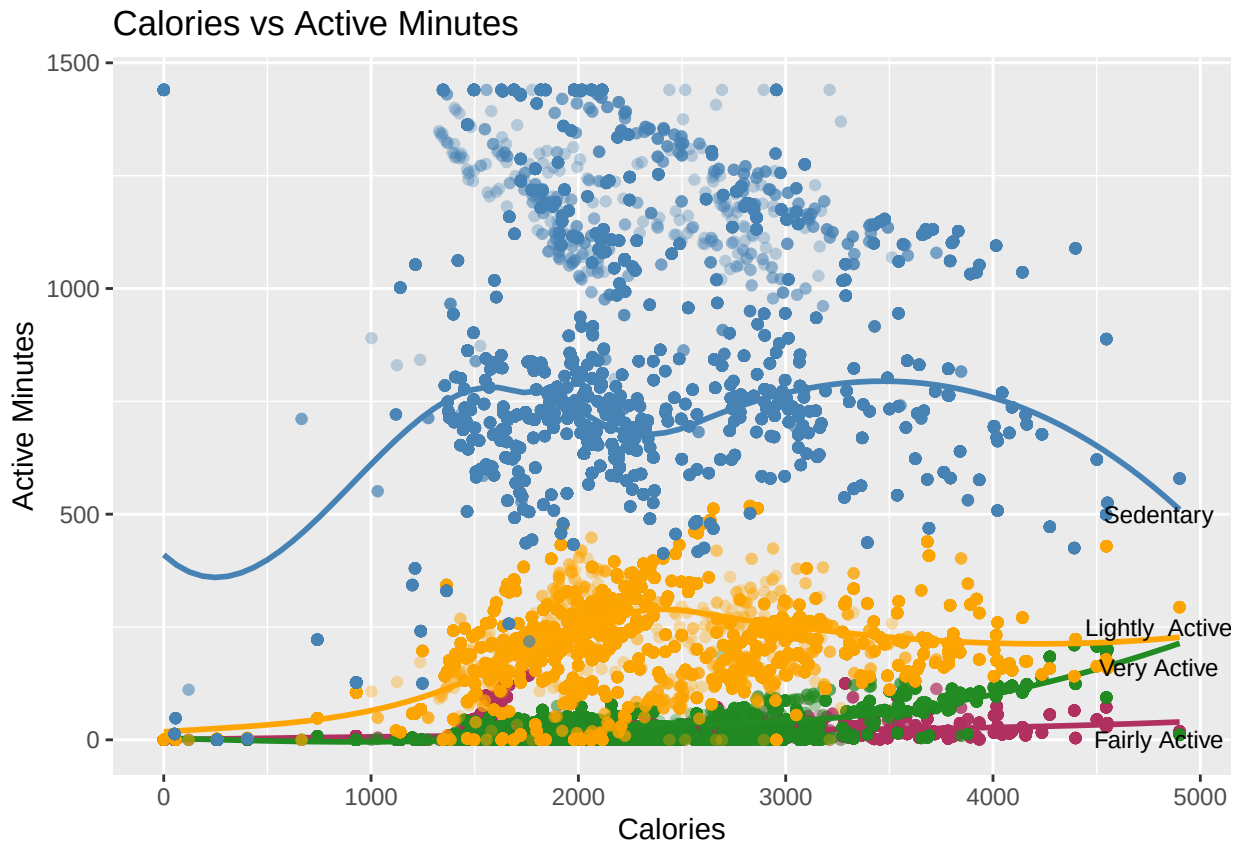
```
ggplot(data = set2) +
  geom_point(mapping=aes(x=Calories, y=FairlyActiveMinutes), color = "maroon", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x, mapping=aes(x=Calories, y=FairlyActiveMinutes, color=Fairly

  geom_point(mapping=aes(x=Calories, y=VeryActiveMinutes), color = "forestgreen", alpha = 1/3) +
  geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=VeryActiveMinutes, color=VeryActi
```

```
geom_point(mapping=aes(x=Calories, y=LightlyActiveMinutes), color = "orange", alpha = 1/3) +
geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=LightlyActiveMinutes, color=Light

geom_point(mapping=aes(x=Calories, y=SedentaryMinutes), color = "steelblue", alpha = 1/3) +
geom_smooth(method = loess,formula =y ~ x,mapping=aes(x=Calories, y=SedentaryMinutes, color=Sedentary

annotate("text", x=4800, y=160, label="Very Active", color="black", size=3)+
annotate("text", x=4800, y=0, label="Fairly Active", color="black", size=3)+
annotate("text", x=4800, y=500, label="Sedentary", color="black", size=3)+
annotate("text", x=4800, y=250, label="Lightly  Active", color="black", size=3)+
labs(x = "Calories", y = "Active Minutes", title="Calories vs Active Minutes")
```



##Recommendations The users are having more sedentary minutes Suggestion for the Bellabeat app *Average steps per day are 4738, which is quite lower than the healthy count of 8000 - 10000 steps given by CDC, thus the app can motivate users to achieve the daily target of 10000 steps.* The app can include a weight loss program where users are made aware of their calorie burn and active time.

*The data shows users with high sedentary time have lower sleep time which affects quality sleep that in turn has negative health effects, thus the app can remind users to take a walk or do the movement at regular intervals.

*More time in bed shows more sleep time, thus the app can notify users of the bedtime on a daily basis which can also improve their sleep cycle and overall mental and physical health.