Case study with R

Bellabeat Company

Sanusi

3/1/2023

Introduction and background

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Its aim is to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

Questions for Analysis

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy?

Objective

Identify trends in how consumers use non-Bellabeat smart device the insights discovered will then help guide marketing strategy for the company which could help unlock new growth opportunities for the company.

Loading common Packages and libraries

```r
library(tidyverse)
library(janitor)
library(ggplot2)
library(dplyr)
library(tidyr)
library(readr)
library(skimr)
library(scales)
```

Loading the csv files

```r
daily_activities = read_csv('dailyActivity_merged.csv')
sleep_day = read_csv('sleepDay_merged.csv')
weight = read_csv('weightLogInfo_merged.csv')
```

Exploring the daily_activities data

Taking a look at the data

```
# to display the first 6 rows
head(daily_activities)
```

```
## # A tibble: 6 x 15
##         Id Activ~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8 Seden~9
##      <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~   13162    8.5     8.5        0    1.88   0.550    6.06       0
## 2 1.50e9 4/13/2~   10735    6.97    6.97       0    1.57   0.690    4.71       0
## 3 1.50e9 4/14/2~   10460    6.74    6.74       0    2.44   0.400    3.91       0
## 4 1.50e9 4/15/2~    9762    6.28    6.28       0    2.14   1.26     2.83       0
## 5 1.50e9 4/16/2~   12669    8.16    8.16       0    2.71   0.410    5.04       0
## 6 1.50e9 4/17/2~    9705    6.48    6.48       0    3.19   0.780    2.51       0
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>, and abbreviated variable names
## #   1: ActivityDate, 2: TotalSteps, 3: TotalDistance, 4: TrackerDistance,
## #   5: LoggedActivitiesDistance, 6: VeryActiveDistance,
## #   7: ModeratelyActiveDistance, 8: LightActiveDistance,
## #   9: SedentaryActiveDistance
```

Identifying the column names and format

```
# To display the column names in the daily_activity dataframe
colnames(daily_activities)
```

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

```
# To display the format of the columns
str(daily_activities)
```

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Getting a summary of the daily_activities data. By displaying the number of missing values, the complete rates of each column

```r
#To display the the shape of the dataset and some descriptive statistics like the mean and standard dev
skim_without_charts(daily_activities)
```

Table 1: Data summary

Name	daily_activities
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
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
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

To confirm the number of distinct users

```
n_distinct(daily_activities$Id)
```

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

Checking for duplicate values

```
num_of_duplicate <- sum(duplicated(daily_activities))
cat("There are ",num_of_duplicate," number of duplicate in daily activity data")
```

```
## There are  0  number of duplicate in daily activity data
```

Exploring the sleep data

Taking a close look at the sleep_day data

```
head(sleep_day)
```

```
## # A tibble: 6 x 5
##           Id SleepDay              TotalSleepRecords TotalMinutesAsleep TotalT~1
##        <dbl> <chr>                             <dbl>              <dbl>    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327      346
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384      407
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412      442
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340      367
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700      712
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304      320
## # ... with abbreviated variable name 1: TotalTimeInBed
```

Identifying the column names and format

```
colnames(sleep_day)
```

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

Getting a summary of the daily_activities data. By displaying the number of missing values, the complete rates of each column, mean and so on

```
skim_without_charts(sleep_day)
```

Table 4: Data summary

Name	sleep_day
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
Id	0	1	5.000979e+09	2.060360e+09	1503960366	3977333714	4702921684	6962181067	8792009665
TotalSleepRecords	0	1	1.120000e+00	3.450000e-01	1	1	1	1	3
TotalMinutesAsleep	0	1	4.194700e+02	1.218340e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.227100e+02	61	403	463	526	961

To confirm the number of distinct users

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

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

To check the format of the columns in the dataset

```r
str(sleep_day)
```

```
## spc_tbl_ [413 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                : num [1:413] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr [1:413] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:0
##  $ TotalSleepRecords : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   SleepDay = col_character(),
##   ..   TotalSleepRecords = col_double(),
##   ..   TotalMinutesAsleep = col_double(),
##   ..   TotalTimeInBed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Checking for duplicate values

```r
num_of_duplicate_s <- sum(duplicated(sleep_day))
cat("There are ",num_of_duplicate_s," number of duplicate in sleep_day data")
```

```
## There are  3  number of duplicate in sleep_day data
```

Exploring the weight dataset

```r
head(weight)
```

```
## # A tibble: 6 x 8
##           Id Date                  WeightKg Weight~1   Fat   BMI IsMan~2   LogId
##        <dbl> <chr>                    <dbl>    <dbl> <dbl> <dbl> <lgl>     <dbl>
## 1 1503960366 5/2/2016 11:59:59 PM      52.6     116.    22  22.6 TRUE    1.46e12
## 2 1503960366 5/3/2016 11:59:59 PM      52.6     116.    NA  22.6 TRUE    1.46e12
## 3 1927972279 4/13/2016 1:08:52 AM     134.      294.    NA  47.5 FALSE   1.46e12
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.    NA  21.5 TRUE    1.46e12
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.    NA  21.7 TRUE    1.46e12
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     160.    25  27.5 TRUE    1.46e12
## # ... with abbreviated variable names 1: WeightPounds, 2: IsManualReport
```

Identifying the column names and checking the format

```r
# To display the column names in the METs dataframe
colnames(weight)
```

```
## [1] "Id"             "Date"           "WeightKg"       "WeightPounds"
## [5] "Fat"            "BMI"            "IsManualReport" "LogId"
```

```r
# To display the format of the columns
str(weight)
```

```
## spc_tbl_ [67 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id            : num [1:67] 1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr [1:67] "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "
##  $ WeightKg      : num [1:67] 52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num [1:67] 116 116 294 125 126 ...
##  $ Fat           : num [1:67] 22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num [1:67] 22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: logi [1:67] TRUE TRUE FALSE TRUE TRUE TRUE ...
##  $ LogId         : num [1:67] 1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   Date = col_character(),
##   ..   WeightKg = col_double(),
##   ..   WeightPounds = col_double(),
##   ..   Fat = col_double(),
##   ..   BMI = col_double(),
##   ..   IsManualReport = col_logical(),
##   ..   LogId = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Getting a summary of the daily_activities data

```r
#To check for the missing values and complete rate of the dataset and some descriptive statistics like
skim_without_charts(weight)
```

Table 7: Data summary

Name	weight
Number of rows	67
Number of columns	8
Column type frequency:	
character	1
logical	1
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

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
IsManualReport	0	1	0.61	TRU: 41, FAL: 26

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
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

To confirm the number of distinct users

```
n_distinct(weight$Id)
```

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

Checking for any duplicate

```
num_of_duplicate_W <- sum(duplicated(daily_activities))
cat("There are ",num_of_duplicate_W," number of duplicate in weight data")
```

```
## There are  0  number of duplicate in weight data
```

Cleaning and formating

While accessing the data, I discovered that the **sleepday** dataset has 3 duplicate rows, some data quality issues in the dataset. The cleaning steps are documented below

```
# Removing duplicate rows in the sleepday data
sleep_day_cleaned = distinct(sleep_day)
```

Checking to see if the changes have been made

```
sum(duplicated(sleep_day_cleaned))
```

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

Changing the date format to datetime datatype

```
# changing the ActivityDate, sleepday column to date format from charater

daily_activities$ActivityDate <- mdy(daily_activities$ActivityDate)
```

```r
# changing format for sleepDay column in sleepDay data

sleep_day_cleaned$SleepDay <- mdy_hms(sleep_day_cleaned$SleepDay)
```

```r
# changing format for date column in weight data

weight$Date <- mdy_hms(weight$Date)
```

to check if the changes were made

```r
daily_activities$ActivityDate[1]
```

```
## [1] "2016-04-12"
```

```r
sleep_day_cleaned$SleepDay[1]
```

```
## [1] "2016-04-12 UTC"
```

```r
weight$Date[1]
```

```
## [1] "2016-05-02 23:59:59 UTC"
```

Adding a new weekday column to the dailyactivity dataset

```r
# Extracting the day from the date column
new_daily_activities <- daily_activities %>%
  mutate(weekday = wday(ActivityDate,label = TRUE))

# to check if the changes were made
new_daily_activities$weekday[1:3]
```

```
## [1] Tue Wed Thu
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

Analysis

This phase will include merging of 3 datasets,grouping of average steps and average active minutes by users, calculating the total active minutes and categorizing the steps into how active the user is.

Merging the daily activity, sleepday datasets

```r
# Merging the daily activity and sleepday file with outer join because the number of observation for da
mergee <- left_join(new_daily_activities,sleep_day_cleaned, by = "Id", multiple = 'all')

# To check our merged file
head(mergee)
```

```
## # A tibble: 6 x 20
##           Id ActivityD~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8
##        <dbl> <date>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## 2 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## 3 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## 4 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## 5 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## 6 1503960366 2016-04-12    13162     8.5     8.5       0    1.88   0.550    6.06
## # ... with 11 more variables: SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>,
## #   weekday <ord>, SleepDay <dttm>, TotalSleepRecords <dbl>,
## #   TotalMinutesAsleep <dbl>, TotalTimeInBed <dbl>, and abbreviated variable
## #   names 1: ActivityDate, 2: TotalSteps, 3: TotalDistance, 4: TrackerDistance,
## #   5: LoggedActivitiesDistance, 6: VeryActiveDistance, ...
```

```
nrow(mergee)
```

```
## [1] 12575
```

```
# to view the coloumn name in our merged file
colnames(mergee)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"                 "weekday"
## [17] "SleepDay"                 "TotalSleepRecords"
## [19] "TotalMinutesAsleep"       "TotalTimeInBed"
```

Grouping the average steps and average minutes by user Id

```
group_daily_average <- mergee %>%
  rowwise() %>%
  mutate(Total = sum(across(ends_with("Minutes")))) %>%
  group_by(Id) %>%
  summarize(mean_steps = mean(TotalSteps),mean_distance = mean(TotalDistance),Avg_very_active_min = mea
            Avg_fairly_active_min = mean(FairlyActiveMinutes),Avg_lightly_active_min = mean(LightlyActi
            Avg_sedentary_min = mean(SedentaryMinutes),total_active_minutes = mean(Total),mean_calories
            )
# to check the changes made
head(group_daily_average)
```

```
## # A tibble: 6 x 10
##       Id mean_~1 mean_~2 Avg_v~3 Avg_f~4 Avg_l~5 Avg_s~6 total~7 mean_~8 Avg_s~9
##    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9  12117.   7.81   38.7    19.2     220.     848.   1126.   1816.    360.
## 2 1.62e9   5744.   3.91    8.68    5.81    153.    1258.   1426.   1483.     NA
## 3 1.64e9   7283.   5.30    9.57   21.4     178.    1162.   1371.   2811.    294
## 4 1.84e9   2580.   1.71    0.129   1.29    115.    1207.   1323.   1573.    652
## 5 1.93e9    916.   0.635   1.32    0.774    38.6   1317.   1358.   2173.    417
## 6 2.02e9  11371.   8.08   36.3    19.4     257.    1113.   1426.   2510.     NA
## # ... with abbreviated variable names 1: mean_steps, 2: mean_distance,
## #   3: Avg_very_active_min, 4: Avg_fairly_active_min,
## #   5: Avg_lightly_active_min, 6: Avg_sedentary_min, 7: total_active_minutes,
## #   8: mean_calories, 9: Avg_sleep_time
```

```
# to confirm the number of users
nrow(group_daily_average)
```

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

Creating a new User Active level column

This column will group the users into how active they. The grouping will be splited into these four categories:

- Sedentary : users with less than 5000 steps per day
- lightly active: users within the range of 5000 to 7500 steps per day
- Fairly active: users within the range of 7500 to 10000 steps per day
- Very active users: users with more than 10000 steps per day

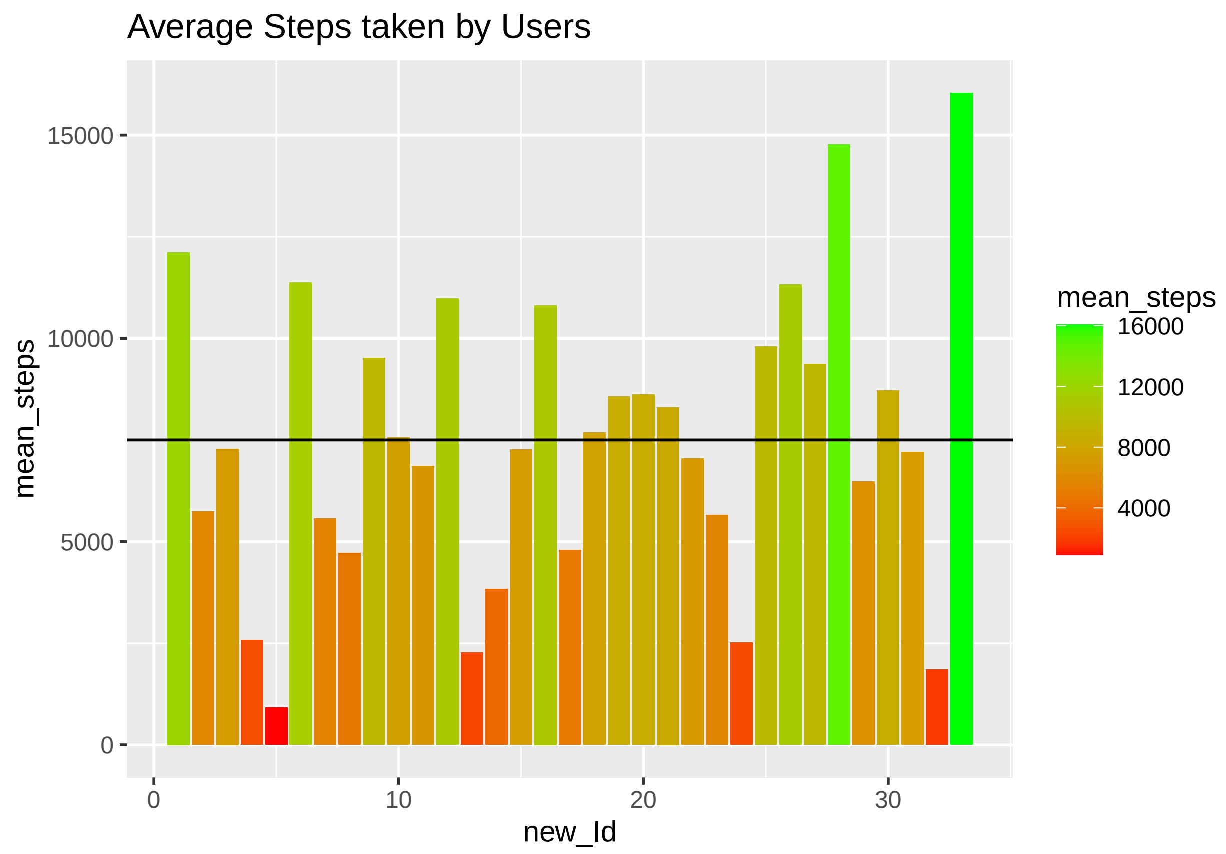
```
user_activity_level <- group_daily_average %>%
  mutate(user_active_level = case_when(
  mean_steps < 5000 ~ "Sedentary",
  mean_steps >= 5000 & mean_steps < 7500 ~ "Lightly user",
  mean_steps >= 7500 & mean_steps < 10000 ~ "Fairly user",
  mean_steps >= 10000 ~ "Active user"
))

# to check if the user active level has been added
head(user_activity_level)
```

```
## # A tibble: 6 x 11
##        Id mean_~1 mean_~2 Avg_v~3 Avg_f~4 Avg_l~5 Avg_s~6 total~7 mean_~8 Avg_s~9
##    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9  12117.   7.81   38.7    19.2     220.     848.   1126.   1816.    360.
## 2 1.62e9   5744.   3.91    8.68    5.81    153.    1258.   1426.   1483.     NA
## 3 1.64e9   7283.   5.30    9.57   21.4     178.    1162.   1371.   2811.    294
## 4 1.84e9   2580.   1.71    0.129   1.29    115.    1207.   1323.   1573.    652
## 5 1.93e9    916.   0.635   1.32    0.774    38.6   1317.   1358.   2173.    417
## 6 2.02e9  11371.   8.08   36.3    19.4     257.    1113.   1426.   2510.     NA
## # ... with 1 more variable: user_active_level <chr>, and abbreviated variable
## #   names 1: mean_steps, 2: mean_distance, 3: Avg_very_active_min,
## #   4: Avg_fairly_active_min, 5: Avg_lightly_active_min, 6: Avg_sedentary_min,
## #   7: total_active_minutes, 8: mean_calories, 9: Avg_sleep_time
```

Visualization

Average steps per user

```
group_daily_average %>%
  mutate("new_Id" = c(1:33)) %>%
  ggplot(aes(x= new_Id,y= mean_steps)) + geom_col(aes(fill = mean_steps)) + geom_hline(yintercept = 750
```



```
# To check the percentage of users who meet the recommended steps per day
average_requirement <- group_daily_average %>%
  mutate(req = case_when(
    mean_steps >= 7500 ~ "good",
    mean_steps < 7500 ~ "bad"
    ))

req_percent <- average_requirement %>%
  group_by(req) %>%
  summarise(per = n()) %>%
  mutate(total = sum(per)) %>%
  group_by(req) %>%
  summarize(total_percen = per/total) %>%
  mutate(labelss = percent(total_percen))

req_percent
```

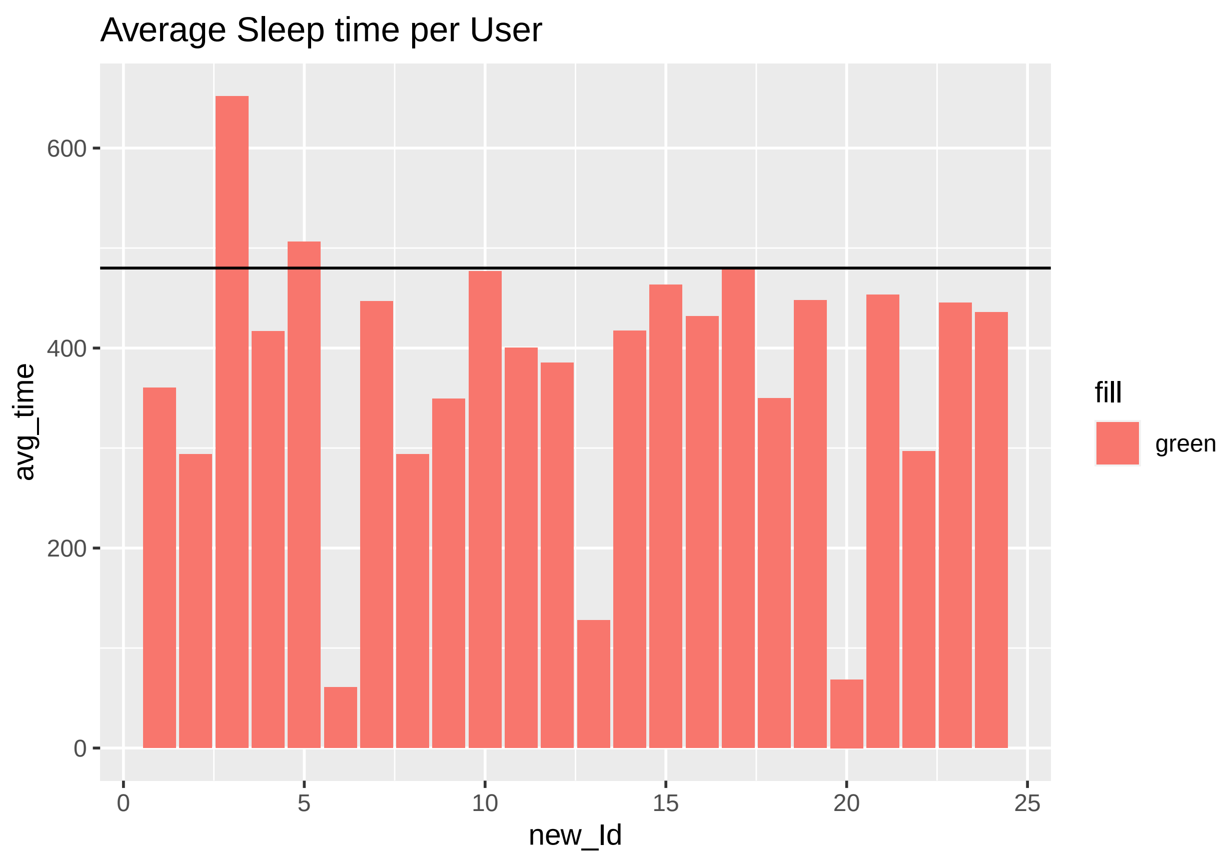
```
## # A tibble: 2 x 3
##   req   total_percen labelss
##   <chr>        <dbl> <chr>
## 1 bad          0.515 51.5%
## 2 good         0.485 48.5%
```

Observation

From observing the above chart, 51.5% of the users take less than the required amount of steps(7500) recommended for general health and fitness

Average time slept per day

```
sleep_day_cleaned %>% group_by(Id) %>%
  summarise(avg_time = mean(TotalMinutesAsleep)) %>%
  mutate("new_Id" = c(1:24)) %>%
  ggplot(aes(x= new_Id, y= avg_time)) + geom_col(aes(fill = "green")) + geom_hline(yintercept = 480,col
```

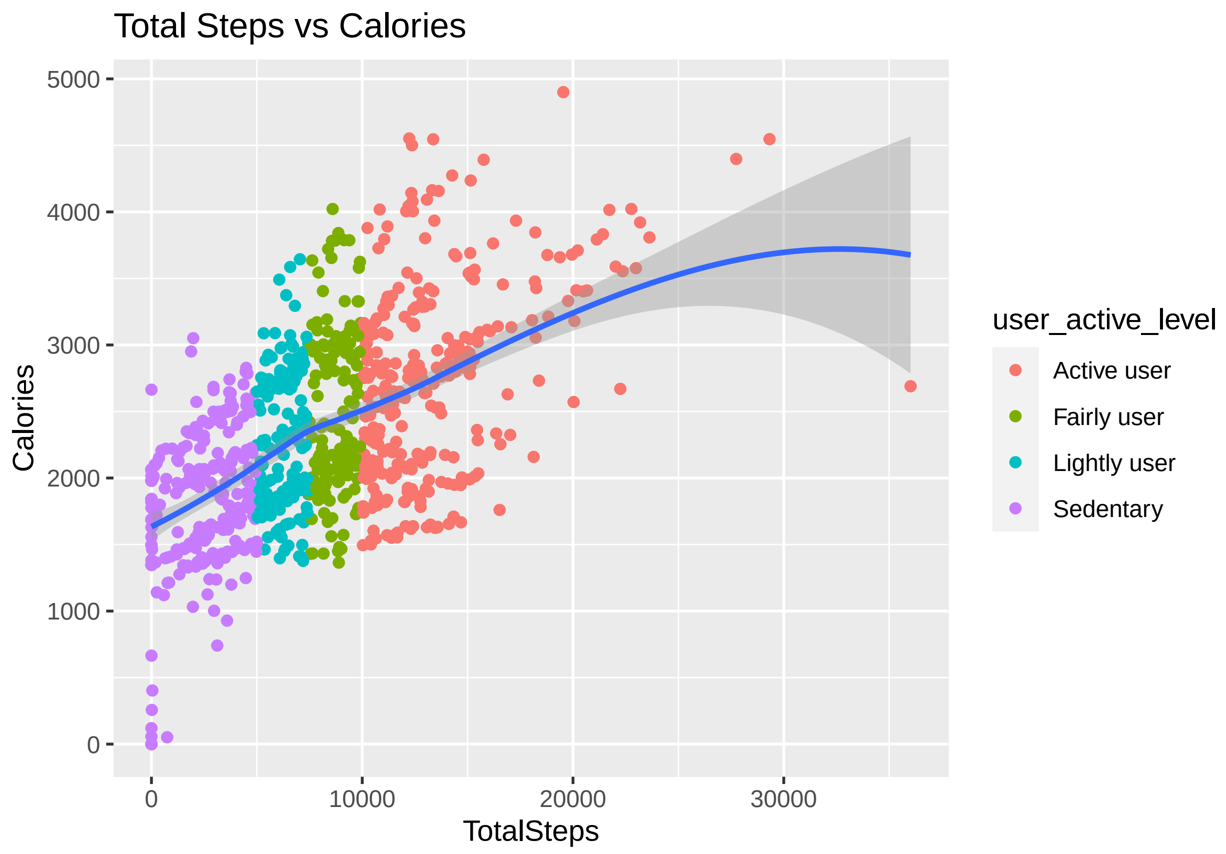


Observation: Most users do not meet the recommended sleep time which is 8hours(480min).

Total steps and Calories

```
# Creating a new column to include how active each user is per date
nne <- new_daily_activities %>%
  mutate(user_active_level = case_when(
  TotalSteps < 5000 ~ "Sedentary",
  TotalSteps >= 5000 & TotalSteps < 7500 ~ "Lightly user",
  TotalSteps >= 7500 & TotalSteps < 10000 ~ "Fairly user",
  TotalSteps >= 10000 ~ "Active user"))

ggplot(data = nne, aes(x= TotalSteps, y= Calories)) + geom_point(aes(color = user_active_level)) + geom_
```

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



Observation Active users tends to take in more calories. The plot also shows that users that take in more calories tends to take more steps

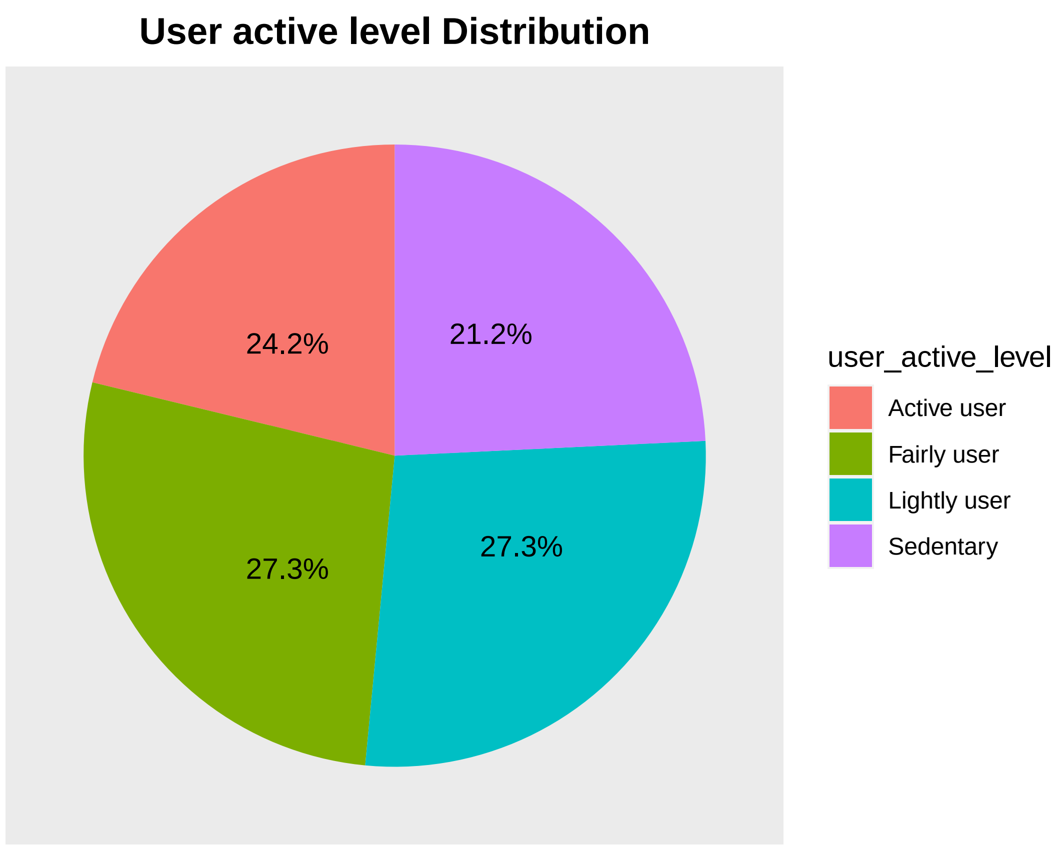
Ploting the User active level Distribution

```
# Calculating the percentage of each active level
user_percent <- user_activity_level %>%
  group_by(user_active_level) %>%
  summarise(num = n()) %>%
  mutate(total = sum(num)) %>%
  group_by(user_active_level) %>%
```

```r
  summarize(total_percent = num/total) %>%
  mutate(labels = percent(total_percent))

# checking the data
head(user_percent)
```

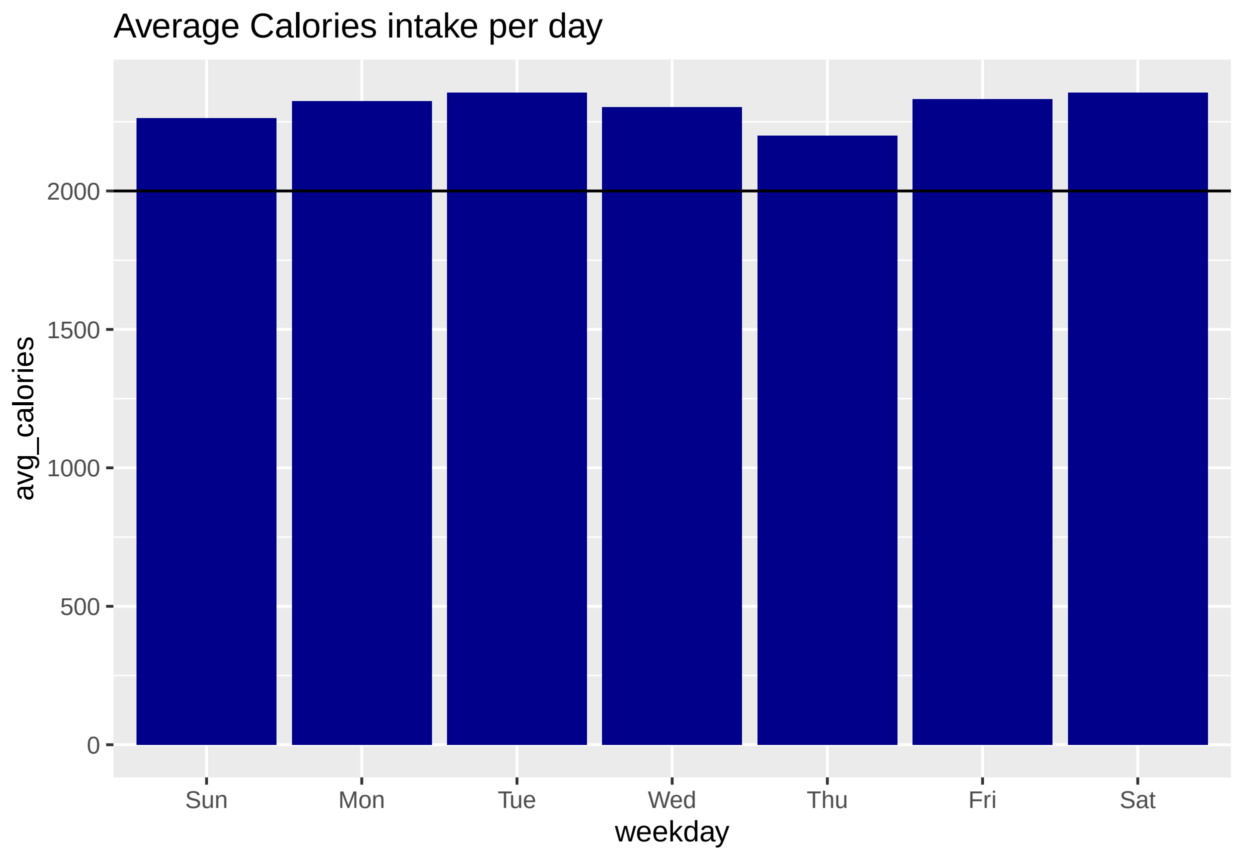
```
## # A tibble: 4 x 3
##   user_active_level total_percent labels
##   <chr>                     <dbl> <chr>
## 1 Active user               0.212 21.2%
## 2 Fairly user               0.273 27.3%
## 3 Lightly user              0.273 27.3%
## 4 Sedentary                 0.242 24.2%
```

```r
# Visualizing
user_percent %>%
  ggplot(aes(x="", y= total_percent)) + geom_bar(aes(fill = user_active_level),
                                                stat = "identity",width = 1 ) +
  coord_polar("y",start = 0) +
  theme(plot.title = element_text(hjust = 0.5,size = 14, face = "bold"),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.text.x  = element_blank(),
        axis.ticks = element_blank()) +
  geom_text(aes(label = labels),position = position_stack(vjust = 0.5)) +
  labs(title = "User active level Distribution")
```



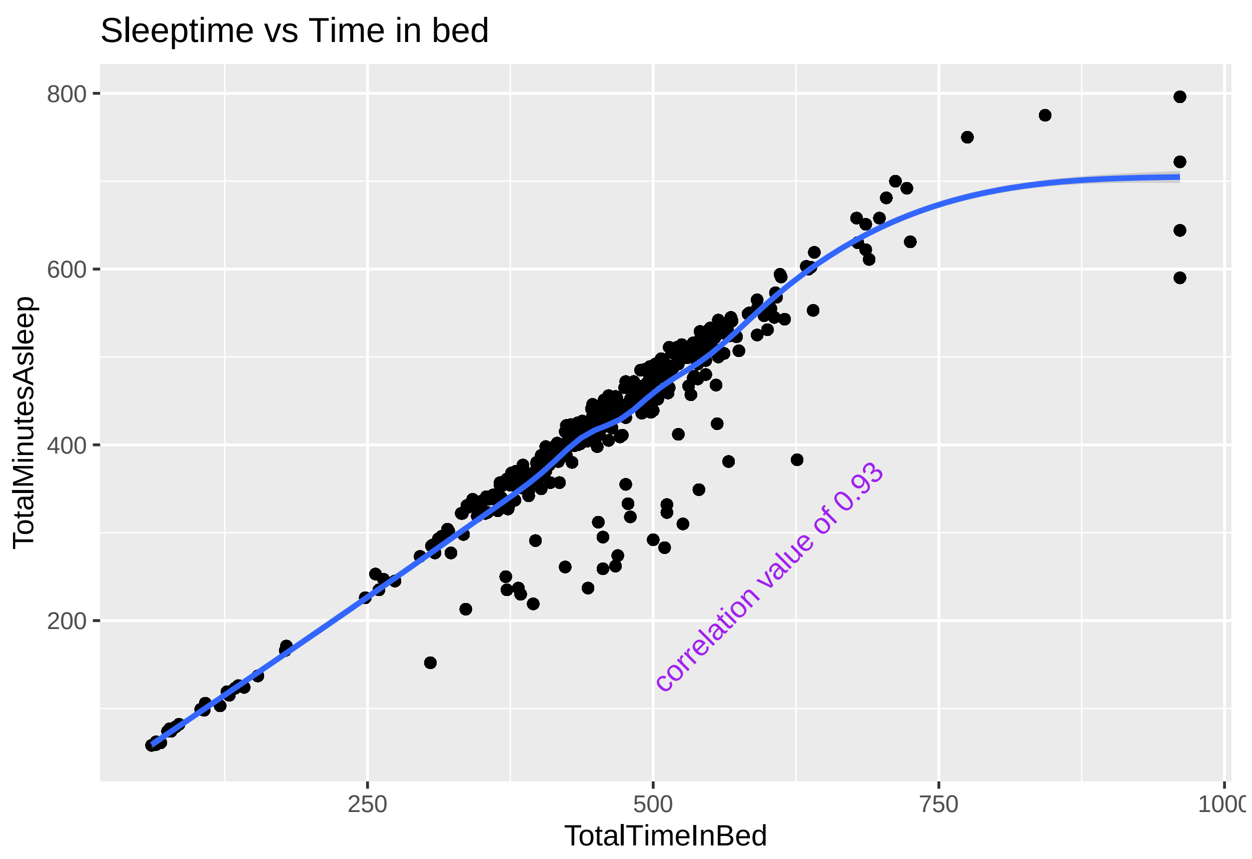
## Average Calories intake per day

```
new_daily_activities %>%
  group_by(weekday) %>%
  summarize(avg_calories = mean(Calories) ) %>%
  ggplot(aes(x= weekday, y = avg_calories)) + geom_col(fill = "darkblue") + geom_hline(yintercept = 200
```



Observation: The users meet the recommended calories intake per day

```
res <- cor.test(mergee$TotalTimeInBed,mergee$TotalMinutesAsleep, method = "pearson")
ggplot(data = mergee, aes(x= TotalTimeInBed , y = TotalMinutesAsleep )) + geom_point() + geom_smooth() 
```



Observation It is observed that there is a strong positive relationship between the time spent on bed and sleep time.

Categorizing the users BMI into normal, overweight and obese

- Normal BMI fall within 18.5 to 24.9 kg/m^2
- Overweight fall within 25 to 29.9 kg/m^2
- Obese 30 kg/m^2 above

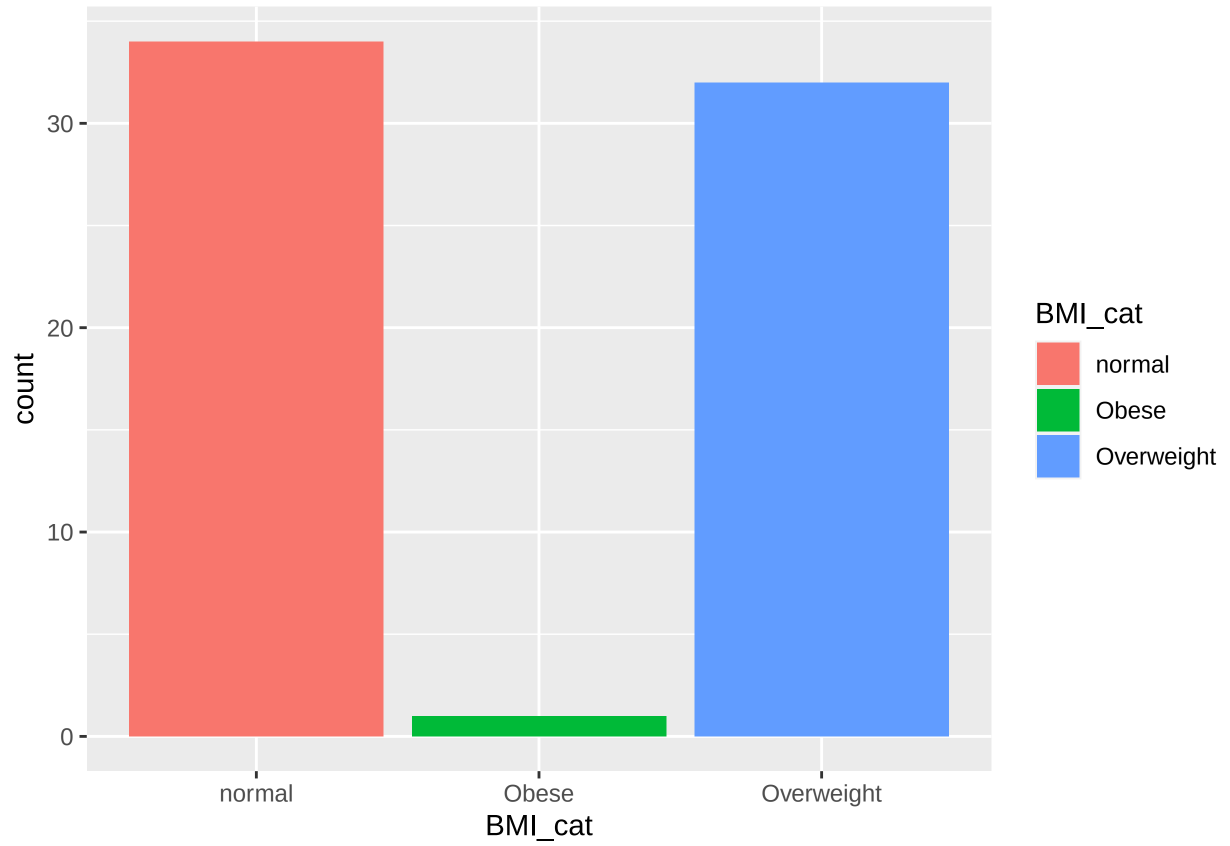
```
new_weight <- weight %>%
  mutate(BMI_cat = case_when(
    BMI >= 18.5 & BMI < 24.9 ~ "normal",
    BMI >=25 & BMI < 30 ~ "Overweight",
    BMI >= 30 ~ "Obese"
  )) %>%
  select(Id,WeightKg,BMI,BMI_cat)

# checking the changes we made
head(new_weight)
```

```
## # A tibble: 6 x 4
##             Id WeightKg   BMI BMI_cat
##          <dbl>    <dbl> <dbl> <chr>
## 1 1503960366     52.6  22.6 normal
## 2 1503960366     52.6  22.6 normal
## 3 1927972279    134.   47.5 Obese
## 4 2873212765     56.7  21.5 normal
## 5 2873212765     57.3  21.7 normal
## 6 4319703577     72.4  27.5 Overweight
```

```r
# grouping the users into the 3 categories of user BMI
new_weight %>%
  group_by(BMI_cat) %>%
  summarise(per = n()) %>%
  mutate(total = sum(per)) %>%
  group_by(BMI_cat) %>%
  summarize(total_percen = per/total) %>%
  mutate(labelss = percent(total_percen))
```

```
## # A tibble: 3 x 3
##   BMI_cat     total_percen labelss
##   <chr>              <dbl> <chr>
## 1 Obese             0.0149 1.5%
## 2 Overweight        0.478  47.8%
## 3 normal            0.507  50.7%
```

```r
# Visualizing the user weight distribution
ggplot(data = new_weight, aes(x= BMI_cat)) + geom_bar(aes(fill = BMI_cat))
```



Observation From the chart above, we observed that although 50.7% of users have a normal BMI, a significant amount of users(47.8%) are overweight and small set of users are obese.

Conclusion

The main aim of this analysis is to discover trends that could help guide the marketing strategy for Bellabeat company which could help unlock new growth opportunities. After exploring the datasets and doing some research on this case study, I discover the following insights

- It was discovered that most of the users do not take the recommended average step by day which is 7500.
- After observing the relationship between the sleep time and time in bed, it is observed that the higher the time spent on bed the higher the sleep time
- It was also discovered that most of the users fall under the recommended sleep time to saty healthy which is 8hrs/480mins.
- From the relationship between the total steps taken and the total calories per day, it was observed that the more calories users take, the more steps taken per day. The relationship also shows that the more active the users are the more calories they take. Active users take more that 10,000 steps per day
- after observing the weight dataset, it was discovered that 49.3% of the users are either overweight or obese

Limitations Some of the limitation of this analysis are:

- Datasets used have a small sample and can be biased and since we didn't have the demographic details of users, we wouldn't know if our sample is referring to young or adult women. Further analysis would be needed to help the marketing strategy know where or who to focus on.
- Dataset does not include categories of physical activities that could be used to by users to improve their fitness goals or understand the type of physical activities that could help burn the most calories.
- The weight dataset used is a very small sample which can be biased

Recommendation

- Since most users failed to meet the recommended sleep time. Bellabeat could improve on their app by including an alarm that remind when sleeping time is getting close. Research as shown that going to sleep at the same time everyday could make it easier to sleep every night and it could also help the body run more efficiently
- The marketing team could use the analysis on the users BMI to help help market bellabeat product to those with overweight and obese, either by reducing the calories intake per day or by increasing the physical activities

- To help market Bellabeat products, the marketing team could advertise the benefit of using their product focusing on the overweight and obese users
- Users with less than an average step of 5000 per day could benefit from Bellabeat app which could help monitor their average steps per day