Project Bellabeat

Tannu verma

2022-03-02

This case study analyzes smart device fitness data to help unlock new growth opportunities for the Bellabeat company. The insights discovered will help guide a marketing strategy for the company.

About the Company

Bellabeat (https://bellabeat.com/) is a high-tech company that manufactures health-focused smart products including, the Bellabeat app, Leaf, Time, and Spring. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their health and habits. Bellabeat also offers a subscription-based membership program for users, giving them access to personalized guidance on having a healthy lifestyle. All of this has positioned itself as a tech-driven wellness company for women.

Characters and Products

Characters

- Urška Sršen: Bellabeat's co founder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat co-founder
- Marketing Analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy

Products

- Bellabeat app: The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products. Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
- Time: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide insights into your daily wellness.
- Spring: This water bottle tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.
- Bellabeat membership: Bellabeat also offers a subscription-based membership program for users. Membership gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health, beauty, and mindfulness-based on their lifestyle and goals.

Business Task

Analyze smart device usage of an existing competitor to identify how consumers use non-Bellabeat devices. Then identify potential opportunities for growth and recommendations for the Bellabeat marketing strategy improvement based on the insights of that analysis.

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?

Prepare - Data Collection

FitBit Fitness Tracker Data (https://www.kaggle.com/arashnic/fitbit) (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.

The dataset has the following 18 files in .csv format.

1.dailyActivity_merged.csv 2.dailyCalories_merged.csv 3.dailyIntensities_merged.csv 4.dailySteps_merged.csv 5.heartrate_seconds_merged.csv 6.hourlyCalories_merged.csv 7.hourlyIntensities_merged.csv 8.hourlySteps_merged.csv 9.minuteCaloriesNarrow_merged.csv 10.minuteCaloriesWide_merged.csv 11.minuteIntensitiesNarrow_merged.csv 12.minuteIntensitiesWide_merged.csv 13.minuteMETsNarrow_merged.csv 14.minuteSleep_merged.csv 15.minuteStepsNarrow_merged.csv 16.minuteStepsWide_merged.csv 17.sleepDay_merged.csv 18.weightLogInfo_merged

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("purrr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library("tidyverse")
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.9
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library("readr")
library("tidyr")
library("skimr")
library("purrr")
library("dplyr")
library("lubridate")
```

```
##
## Attaching package: 'lubridate'
```

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

```
library("ggplot2")
```

Importing Datasets

In order to understand user behavior on device usage and apply to daily and weekly habit, I will choose the following datdsets:

```
getwd()
```

```
## [1] "/cloud/project"
```

```
daily_activity<-read_csv("dailyActivity.csv")
```

```
## Rows: 940 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dailyCalories<-read_csv("dailyCalories_merged.csv")
```

```
## Rows: 940 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, Calories
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dailySteps<-read_csv("dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
hourlyCalories<-read_csv("hourlyCalories_merged.csv")
```

```
## Rows: 22099 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, Calories
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
minuteSleep<-read_csv("minuteSleep_merged.csv")
```

```
## Rows: 188521 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): date
## dbl (3): Id, value, logId
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
SleepDay<-read_csv("sleepDay_merged.csv")
```

```
## Rows: 413 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
weightLoginfo<-read_csv("weightLogInfo_merged.csv")
```

```
## Rows: 67 Columns: 8
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Processing and cleaning data

For processing and cleaning data I used library “skimr” to get summary of each datasets,library “purrr” to find and replace missing values,and I format the data and time type for further analysis using “parse_date_time”. Overall, the data cleaning process involves: (1) check if there are any missing data. (2) converting DateTime data with string types to formal date types. (3) ensuring data integrity by checking some logical sense behind datasets.

getting summary of each data

```
#daily_activity

head(daily_activity)
```

```
## # A tibble: 6 × 15
##        Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie…
##     <dbl> <chr>             <dbl>         <dbl>           <dbl>            <dbl>
## 1  1.50e9 04-12-2016        13162          8.5             8.5                0
## 2  1.50e9 4/13/2016         10735          6.97            6.97               0
## 3  1.50e9 4/14/2016         10460          6.74            6.74               0
## 4  1.50e9 4/15/2016          9762          6.28            6.28               0
## 5  1.50e9 4/16/2016         12669          8.16            8.16               0
## 6  1.50e9 4/17/2016          9705          6.48            6.48               0
## # … with 9 more variables: VeryActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>
```

```
skim_without_charts(daily_activity)
```

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	9	10	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	
Id	0	1	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900

```
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"
```

```
#dailycalories

head(dailyCalories)
```

```
## # A tibble: 6 × 3
##           Id ActivityDay Calories
##        <dbl> <chr>          <dbl>
## 1 1503960366 04-12-2016      1985
## 2 1503960366 4/13/2016       1797
## 3 1503960366 4/14/2016       1776
## 4 1503960366 4/15/2016       1745
## 5 1503960366 4/16/2016       1863
## 6 1503960366 4/17/2016       1728
```

```
skim_without_charts(dailyCalories)
```

Data summary

Name	dailyCalories
Number of rows	940
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDay	0	1	9	10	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	2320127002.0	4445114986	6.962181e+09	8877689391
Calories	0	1	2.303610e+03	7.181700e+02	0	1828.5	2134	2.793250e+03	4900

```
colnames(dailyCalories)
```

```
## [1] "Id"          "ActivityDay" "Calories"
```

```
#dailysteps

head(dailySteps)
```

```
## # A tibble: 6 × 3
##           Id ActivityDay StepTotal
##        <dbl> <chr>           <dbl>
## 1 1624580081 05-01-2016      36019
## 2 8877689391 4/16/2016       29326
## 3 8877689391 4/30/2016       27745
## 4 8877689391 4/27/2016       23629
## 5 8877689391 04-12-2016      23186
## 6 8053475328 4/24/2016       22988
```

```
skim_without_charts(dailySteps)
```

Data summary

Name	dailySteps
Number of rows	940
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDay	0	1	9	10	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	4445114986.0	6962181067	8877689391
StepTotal	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7405.5	10727	36019

```
colnames(dailySteps)
```

```
## [1] "Id"          "ActivityDay" "StepTotal"
```

```
#hourlyCalories

head(hourlyCalories)
```

```
## # A tibble: 6 × 3
##           Id ActivityHour     Calories
##        <dbl> <chr>               <dbl>
## 1 1503960366 04-12-2016 00:00       81
## 2 1503960366 04-12-2016 01:00       61
## 3 1503960366 04-12-2016 02:00       59
## 4 1503960366 04-12-2016 03:00       47
## 5 1503960366 04-12-2016 04:00       48
## 6 1503960366 04-12-2016 05:00       48
```

```
skim_without_charts(hourlyCalories)
```

Data summary

Name	hourlyCalories
Number of rows	22099
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	16	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.4225e+09	1503960366	2320127002	4445114986	6962181067	8877689391
Calories	0	1	9.739000e+01	6.0700e+01	42	63	83	108	948

```
colnames(hourlyCalories)
```

```
## [1] "Id"           "ActivityHour" "Calories"
```

```
#minutesleep

head(minuteSleep)
```

```
## # A tibble: 6 × 4
##           Id date                 value       logId
##        <dbl> <chr>                <dbl>       <dbl>
## 1 1503960366 4/12/2016 2:47:30 AM     3 11380564589
## 2 1503960366 4/12/2016 2:48:30 AM     2 11380564589
## 3 1503960366 4/12/2016 2:49:30 AM     1 11380564589
## 4 1503960366 4/12/2016 2:50:30 AM     1 11380564589
## 5 1503960366 4/12/2016 2:51:30 AM     1 11380564589
## 6 1503960366 4/12/2016 2:52:30 AM     1 11380564589
```

```
skim_without_charts(minuteSleep)
```

Data summary

Name	minuteSleep
Number of rows	188521
Number of columns	4
Column type frequency:	
character	1
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
date	0	1	19	21	0	49773	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.996595e+09	2.066950e+09	1503960366	3977333714	4702921684	6962181067	8792009665
value	0	1	1.100000e+00	3.300000e-01	1	1	1	1	3
logId	0	1	1.149611e+10	6.822863e+07	11372227280	11439308639	11501142214	11552534115	11616251768

```
colnames(minuteSleep)
```

```
## [1] "Id"    "date"  "value" "logId"
```

```
#sleepday

head(SleepDay)
```

```
## # A tibble: 6 × 5
##           Id SleepDay           TotalSleepRecor… TotalMinutesAsl… TotalTimeInBed
##        <dbl> <chr>                         <dbl>            <dbl>          <dbl>
## 1 1503960366 04-12-2016 00:00                  1              327            346
## 2 1503960366 4/13/2016 12:00:0…                2              384            407
## 3 1503960366 4/15/2016 12:00:0…                1              412            442
## 4 1503960366 4/16/2016 12:00:0…                2              340            367
## 5 1503960366 4/17/2016 12:00:0…                1              700            712
## 6 1503960366 4/19/2016 12:00:0…                1              304            320
```

```
skim_without_charts(SleepDay)
```

Data summary

Name	SleepDay
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	16	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	2.06036e+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

```
colnames(SleepDay)
```

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

```
#weightLoginfo

head(weightLoginfo)
```

```
## # A tibble: 6 × 8
##           Id Date       WeightKg WeightPounds   Fat   BMI IsManualReport   LogId
##        <dbl> <chr>         <dbl>        <dbl> <dbl> <dbl> <lgl>            <dbl>
## 1 1503960366 05-02-201…     52.6         116.    22  22.6 TRUE           1.46e12
## 2 1503960366 05-03-201…     52.6         116.    NA  22.6 TRUE           1.46e12
## 3 1927972279 4/13/2016…    134.          294.    NA  47.5 FALSE          1.46e12
## 4 2873212765 4/21/2016…     56.7         125.    NA  21.5 TRUE           1.46e12
## 5 2873212765 05-12-201…     57.3         126.    NA  21.7 TRUE           1.46e12
## 6 4319703577 4/17/2016…     72.4         160.    25  27.5 TRUE           1.46e12
```

```
skim_without_charts(weightLoginfo)
```

Data summary

Name	weightLoginfo
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	16	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	1.50396e+09	6.962181e+09	6.962181e+09	8.877689e+09	8.877689e+09
WeightKg	0	1.00	7.204000e+01	1.392000e+01	5.26000e+01	6.140000e+01	6.250000e+01	8.505000e+01	1.335000e+02
WeightPounds	0	1.00	1.588100e+02	3.070000e+01	1.15960e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02
Fat	65	0.03	2.350000e+01	2.120000e+00	2.20000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+01
BMI	0	1.00	2.519000e+01	3.070000e+00	2.14500e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01
LogId	0	1.00	1.461771e+12	7.835136e+08	1.46044e+12	1.461080e+12	1.461800e+12	1.462375e+12	1.463100e+12

```
colnames(weightLoginfo)
```

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

Understanding some summary statistics

Here I used “n_distinct” to find if there is any duplicate in datasets. There are 33 subject IDs in daily_activity,daily Calories,hourly calories data and only 24 IDs in sleepDay,minute sleep data, which means not everyone tracks their sleep patterns. And only 8 IDs in weight data.

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

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

```
n_distinct(SleepDay$Id)
```

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

```
n_distinct(dailyCalories$Id)
```

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

```
n_distinct(hourlyCalories$Id)
```

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

```
n_distinct(minuteSleep$Id)
```

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

```
n_distinct(weightLoginfo$Id)
```

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

finding and replacing missing Values

I found that only dataset “weight_log_info” have 65 missing values in “Fat” column.I replaced these misssing values to “NA”.

```
daily_activity %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $ActivityDate
## [1] 0
##
## $TotalSteps
## [1] 0
##
## $TotalDistance
## [1] 0
##
## $TrackerDistance
## [1] 0
##
## $LoggedActivitiesDistance
## [1] 0
##
## $VeryActiveDistance
## [1] 0
##
## $ModeratelyActiveDistance
## [1] 0
##
## $LightActiveDistance
## [1] 0
##
## $SedentaryActiveDistance
## [1] 0
##
## $VeryActiveMinutes
## [1] 0
##
## $FairlyActiveMinutes
## [1] 0
##
## $LightlyActiveMinutes
## [1] 0
##
## $SedentaryMinutes
## [1] 0
##
## $Calories
## [1] 0
```

```
dailyCalories %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $ActivityDay
## [1] 0
##
## $Calories
## [1] 0
```

```
dailySteps %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $ActivityDay
## [1] 0
##
## $StepTotal
## [1] 0
```

```
hourlyCalories %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $ActivityHour
## [1] 0
##
## $Calories
## [1] 0
```

```
minuteSleep %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $date
## [1] 0
##
## $value
## [1] 0
##
## $logId
## [1] 0
```

```
SleepDay %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $SleepDay
## [1] 0
##
## $TotalSleepRecords
## [1] 0
##
## $TotalMinutesAsleep
## [1] 0
##
## $TotalTimeInBed
## [1] 0
```

```
weightLoginfo %>% map(~sum(is.na(.)))
```

```
## $Id
## [1] 0
##
## $Date
## [1] 0
##
## $WeightKg
## [1] 0
##
## $WeightPounds
## [1] 0
##
## $Fat
## [1] 65
##
## $BMI
## [1] 0
##
## $IsManualReport
## [1] 0
##
## $LogId
## [1] 0
```

```
# replacing missing values
na.strings=c(" ","NA")
head(weightLoginfo)
```

```
## # A tibble: 6 × 8
##           Id Date       WeightKg WeightPounds   Fat   BMI IsManualReport   LogId
##        <dbl> <chr>          <dbl>        <dbl> <dbl> <dbl> <lgl>            <dbl>
## 1 1503960366 05-02-201…     52.6         116.    22  22.6 TRUE           1.46e12
## 2 1503960366 05-03-201…     52.6         116.    NA  22.6 TRUE           1.46e12
## 3 1927972279 4/13/2016…    134.          294.    NA  47.5 FALSE          1.46e12
## 4 2873212765 4/21/2016…     56.7         125.    NA  21.5 TRUE           1.46e12
## 5 2873212765 05-12-201…     57.3         126.    NA  21.7 TRUE           1.46e12
## 6 4319703577 4/17/2016…     72.4         160.    25  27.5 TRUE           1.46e12
```

convert string data type to datetime data type

From the above result, I found that some datasets have redundant information to merge into 1 dataset for further analysis. For example, ActivityDate also represents SleepDay but with a different datetime format. The datetime format is more manageable for further analysis and dataset merge. Therefore, I can first convert the datatype of the date column to datetime format to process the dataset merge.

```
daily_activity$Activity_Date<- parse_date_time(daily_activity$ActivityDate, "%m/%d/%y")

#OR
#daily_activity <- daily_activity %>%
#  mutate(ActivityDate=as.Date(ActivityDate, format = "%m/%d/%y"))
#View(daily_activity)

dailyCalories$Activity_Date<- parse_date_time(dailyCalories$ActivityDay, "%m/%d/%y")

dailySteps$Activity_Date<-parse_date_time(dailySteps$ActivityDay, "%m/%d/%y")

hourlyCalories$Activity_Hour<-parse_date_time(hourlyCalories$ActivityHour, "%m/%d/%y %H:%M:%S")

SleepDay$SleepDate<-parse_date_time(SleepDay$SleepDay,"%m/%d/%y %H:%M:%S")

weightLoginfo$WeightDate<-parse_date_time(weightLoginfo$Date,"%m/%d/%y %H:%M:%S")
```

Ensuring data vaildation to check integrity in date types.

```
#data validation to check data types

class(hourlyCalories$Activity_Hour)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(SleepDay$SleepDate)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(weightLoginfo$WeightDate)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(daily_activity$Activity_Date)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(dailyCalories$Activity_Date)
```

```
## [1] "POSIXct" "POSIXt"
```

```
class(dailySteps$Activity_Date)
```

```
## [1] "POSIXct" "POSIXt"
```

Analyse and Visualisation Phase

Transform sleep_day table

```
SleepDay <- SleepDay %>%
  separate(col = SleepDay, c("date", "time"), sep = " ") %>%
  mutate(date = mdy(date),
         week_day = weekdays(date)) %>%
  select(-time) # all are 12:00:00 AM
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 252 rows [2, 3, 4, 5,
## 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 26, 27, 30, 31, 34, 35, ...].
```

```
head(SleepDay)
```

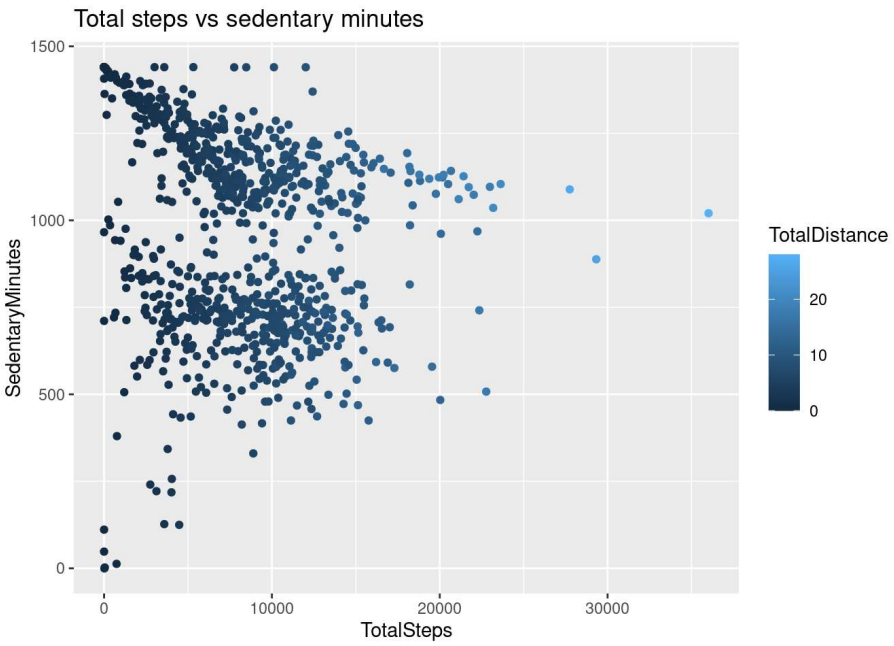
```
## # A tibble: 6 × 7
##           Id date       TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <date>                 <dbl>              <dbl>          <dbl>
## 1 1503960366 2016-04-12                 1                327            346
## 2 1503960366 2016-04-13                 2                384            407
## 3 1503960366 2016-04-15                 1                412            442
## 4 1503960366 2016-04-16                 2                340            367
## 5 1503960366 2016-04-17                 1                700            712
## 6 1503960366 2016-04-19                 1                304            320
## # … with 2 more variables: SleepDate <dttm>, week_day <chr>
```

Plotting some exploration

Data visualization will be presented with topic focus in the following order:

Steps vs sedentary minutes Time in bed vs minute asleep

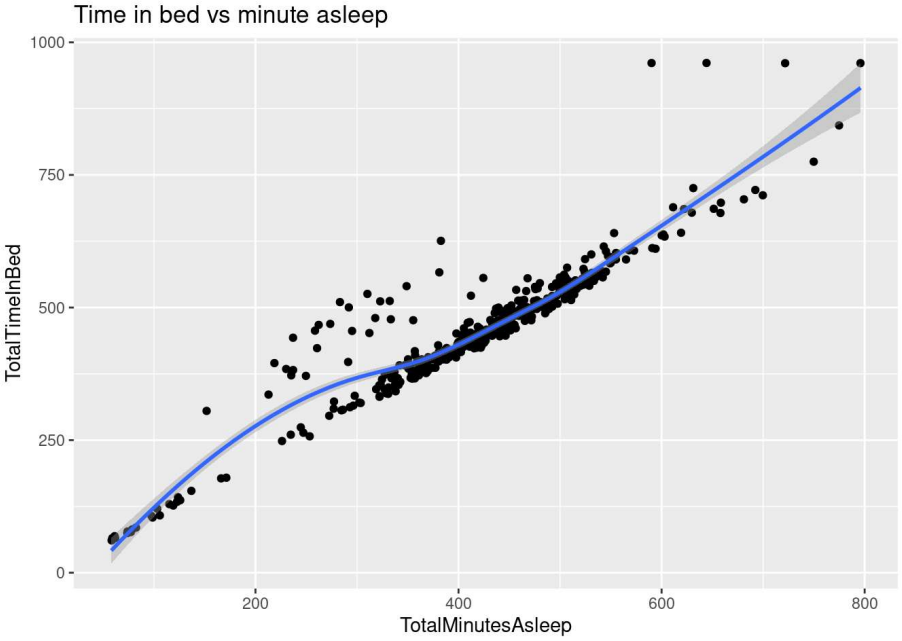
```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes,color=TotalDistance)) + geom_point(position = position_jitt
er())+labs(title="Total steps vs sedentary minutes")
```




```
#We can reduce over-plotting by adding some jitter:
```

The output graphs goes downwards.This shows users who are more sedentary walk less.Therefore they achieve less steps.

```
#Total time in bed vs total minute asleep
ggplot(data=SleepDay, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) + geom_point(position = position_jitter())+geom_smooth(me
thod='loess' ,formula='y ~ x')+labs(title="Time in bed vs minute asleep")
```



There is a 'strong positive correlation' between the amount of time in bed and the total minutes asleep, by FitBit users. The visualisation shows something that was highlighted during the summarisation of the data, that half of the time users aren't getting enough sleep. Bellabeat could use this information to send reminders to their user base from the app telling them the best time to get sleep based on previous nights rest and daily activity.

Merging these two datasets together

```
combined_data <- merge(SleepDay, daily_activity, by= c ("Id"),all=TRUE)
n_distinct(combined_data$Id)
```

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

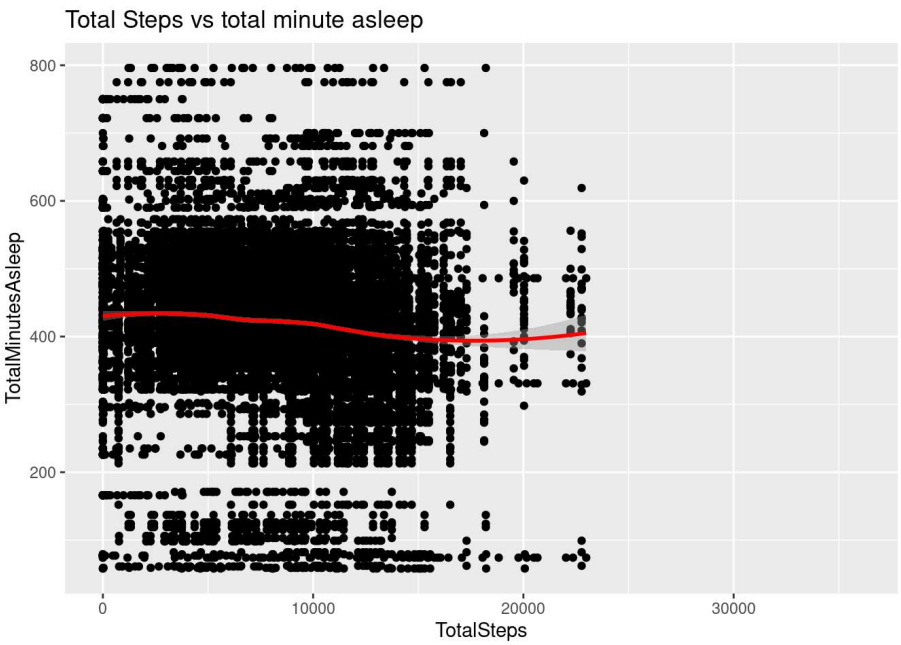
```
head(combined_data)
```

```
##           Id       date TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1 1503960366 2016-04-12                 1                327            346
## 2 1503960366 2016-04-12                 1                327            346
## 3 1503960366 2016-04-12                 1                327            346
## 4 1503960366 2016-04-12                 1                327            346
## 5 1503960366 2016-04-12                 1                327            346
## 6 1503960366 2016-04-12                 1                327            346
##             SleepDate week_day ActivityDate TotalSteps TotalDistance
## 1 2020-04-12 16:00:00  Tuesday   05-07-2016      11992          7.71
## 2 2020-04-12 16:00:00  Tuesday   05-06-2016      12159          8.03
## 3 2020-04-12 16:00:00  Tuesday   05-01-2016      10602          6.81
## 4 2020-04-12 16:00:00  Tuesday    4/30/2016      14673          9.25
## 5 2020-04-12 16:00:00  Tuesday   04-12-2016      13162          8.50
## 6 2020-04-12 16:00:00  Tuesday    4/13/2016      10735          6.97
##   TrackerDistance LoggedActivitiesDistance VeryActiveDistance
## 1            7.71                       0              2.46
## 2            8.03                       0              1.97
## 3            6.81                       0              2.29
## 4            9.25                       0              3.56
## 5            8.50                       0              1.88
## 6            6.97                       0              1.57
##   ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
## 1                     2.12                3.13                       0
## 2                     0.25                5.81                       0
## 3                     1.60                2.92                       0
## 4                     1.42                4.27                       0
## 5                     0.55                6.06                       0
## 6                     0.69                4.71                       0
##   VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
## 1                37                  46                  175              833
## 2                24                   6                  289              754
## 3                33                  35                  246              730
## 4                52                  34                  217              712
## 5                25                  13                  328              728
## 6                21                  19                  217              776
##   Calories Activity_Date
## 1     1821    2016-05-07
## 2     1896    2016-05-06
## 3     1820    2016-05-01
## 4     1947    2016-04-30
## 5     1985    2016-04-12
## 6     1797    2016-04-13
```

```
#Total steps vs total minute sleep
ggplot(data=combined_data, aes(x=TotalSteps, y=TotalMinutesAsleep)) +geom_point() +geom_smooth(color = "red",formula = y ~
 x, method = "loess")+labs(title="Total Steps vs total minute asleep")
```

```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 227 rows containing missing values (geom_point).
```

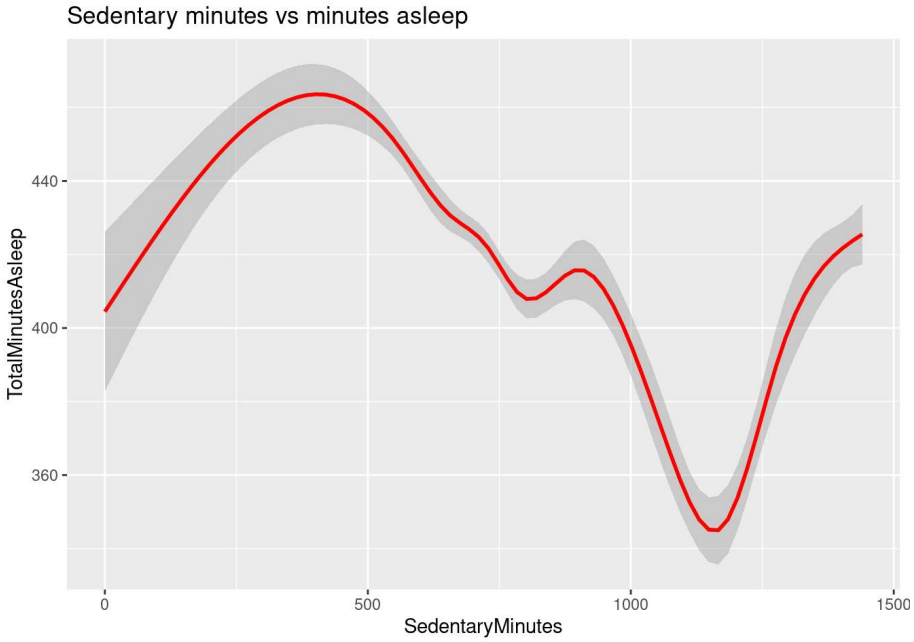


There appears to be no relationship between total daily steps taken across the dataset and minutes slept.

```
#Total minuts asleep vs sedentary minutes
ggplot(data=combined_data, aes(x=SedentaryMinutes,y=TotalMinutesAsleep))+
  geom_smooth(color = "red") +
  labs(title = "Sedentary minutes vs minutes asleep")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

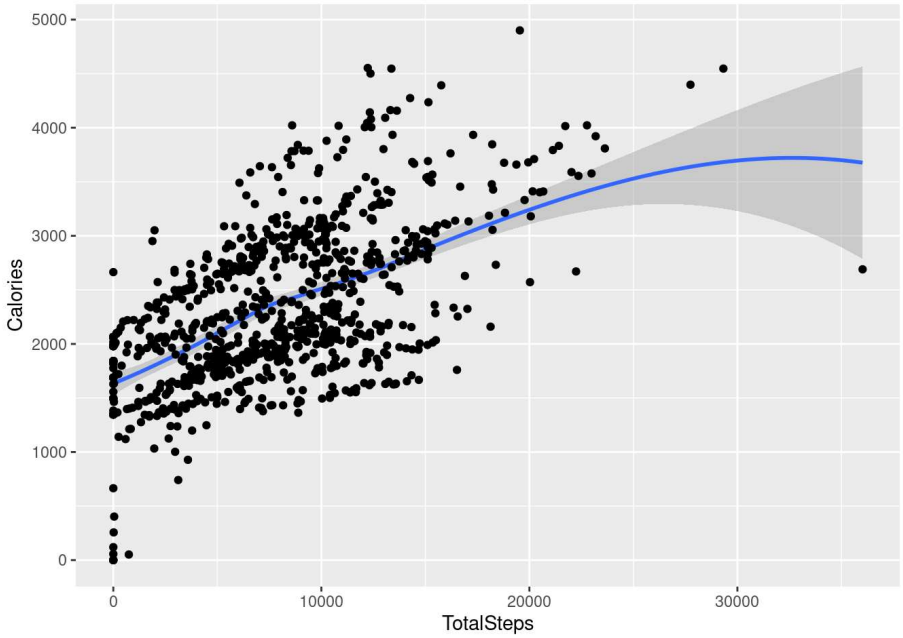
```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```



There is negative relationship

between sedentary minutes and minutes asleep. So,sedentary time can be one of the reasons for poor sleep.

```
ggplot(data=daily_activity, aes(x=TotalSteps, y=Calories)) +geom_smooth(formula = y ~ x, method = "loess")+ geom_point(posit
ion = position_jitter())
```



There is a positive relationship

between total steps and calories burn.More steps taken , more calories burn.

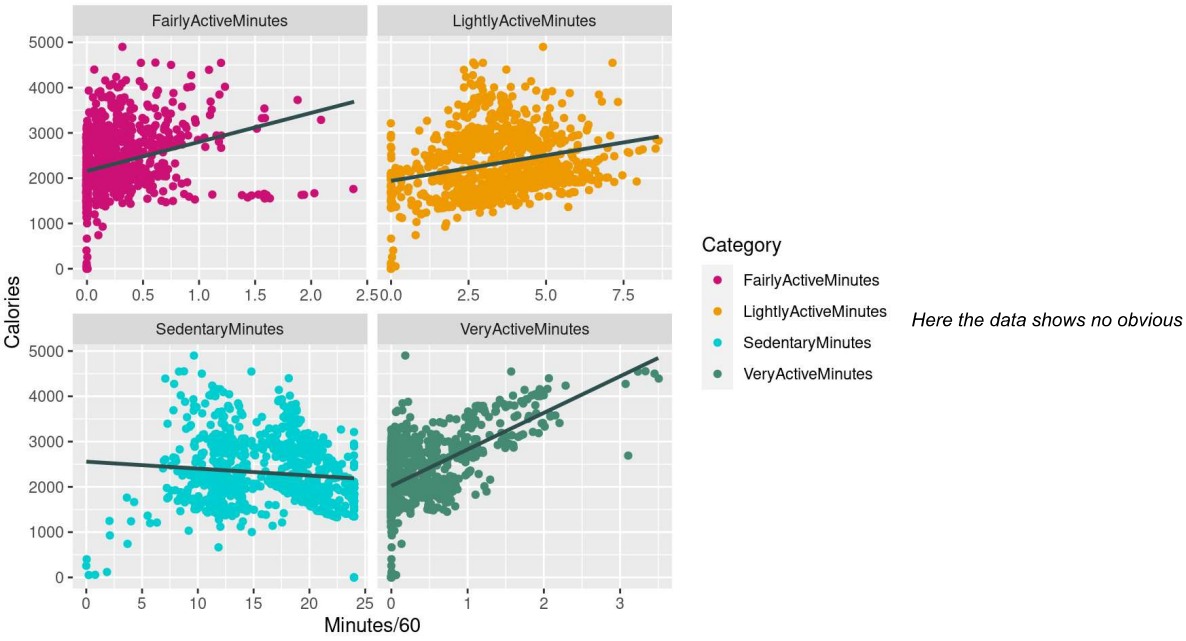
```
activity_min<-daily_activity %>% group_by(Id,ActivityDate)%>%
  select(TotalSteps,SedentaryMinutes,LightlyActiveMinutes,FairlyActiveMinutes,VeryActiveMinutes,Calories) %>%
  pivot_longer(cols=4:7,names_to ="Category",values_to = "Minutes")
```

```
## Adding missing grouping variables: `Id`, `ActivityDate`
```

```
head(activity_min)
```

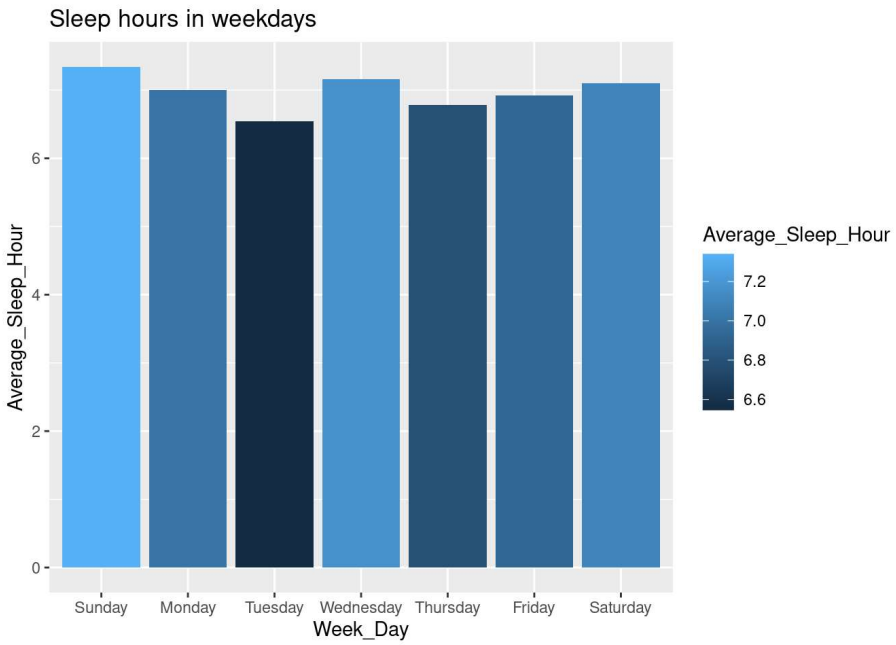
```
## # A tibble: 6 × 6
## # Groups:   Id, ActivityDate [2]
##           Id ActivityDate TotalSteps Calories Category             Minutes
##        <dbl> <chr>             <dbl>    <dbl> <chr>                  <dbl>
## 1 1503960366 04-12-2016        13162     1985 SedentaryMinutes         728
## 2 1503960366 04-12-2016        13162     1985 LightlyActiveMinutes     328
## 3 1503960366 04-12-2016        13162     1985 FairlyActiveMinutes       13
## 4 1503960366 04-12-2016        13162     1985 VeryActiveMinutes         25
## 5 1503960366 4/13/2016         10735     1797 SedentaryMinutes         776
## 6 1503960366 4/13/2016         10735     1797 LightlyActiveMinutes     217
```

```
ggplot(activity_min,aes(x=Minutes/60,y=Calories,color=Category))+
  geom_jitter(stat="identity")+
  scale_color_manual(values=c("deeppink3","orange2","darkturquoise","aquamarine4"))+
  geom_smooth(formula=y~x,color="darkslategrey",method="lm",se=F)+
  facet_wrap(~ Category,scales="free_x")
```



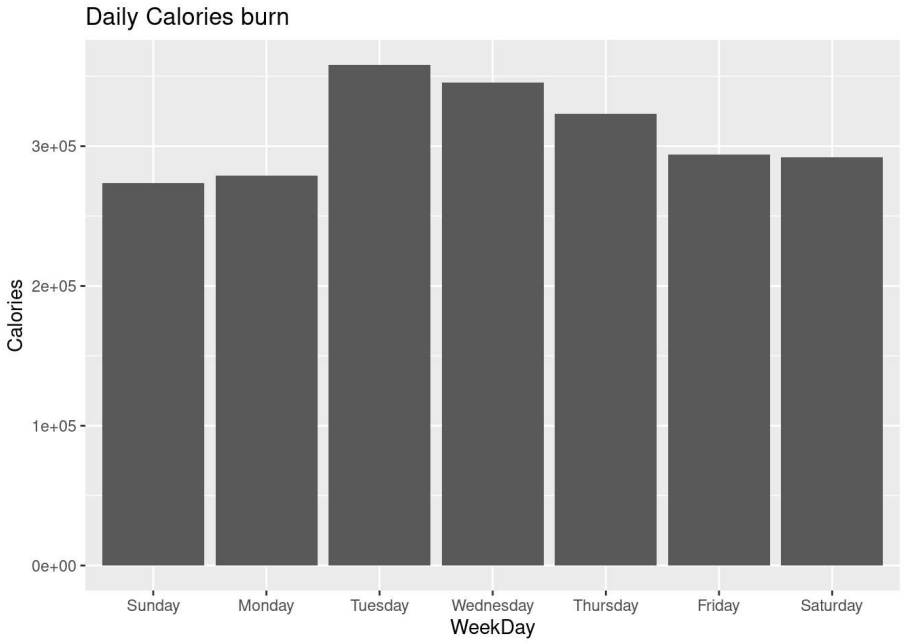
Here the data shows no obvious relationship between fairly active minutes or Lightly active minutes logged per day across the dataset and calories burned. The negligible correlation coefficient strength confirms this. The data shows a positive relationship between 'very active minutes' and calories burned. The correlation coefficient strength is also moderate, but slightly stronger than that between total steps and calories. *The data shows a negative relationship between 'sedentary minutes' and calories burned.

```
# sleep days vs sleep hours
#Calculate total hours
SleepDay$TotalHourSleep <- SleepDay$TotalMinutesAsleep/60

#Add weekday column
SleepDay$sleepWeekday <- weekdays(as.Date(SleepDay$SleepDate))

#Group by weekday
weeklySleepHour<- SleepDay %>% group_by(sleepWeekday)%>%
  summarise(Average_Sleep_Hour = mean(TotalHourSleep),.groups='drop')

#Reorder the plot by sequence of the weekday
Week_Day <- factor(weeklySleepHour$sleepWeekday,level = c('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday'))
ggplot(weeklySleepHour, aes(x = Week_Day, y = Average_Sleep_Hour,fill=Average_Sleep_Hour)) +
  geom_col()+labs(title="Sleep hours in weekdays")
```



We'd like to know the user's sleep pattern during the weekday. We need to add weekday column to the dailyactivity_sleep table. From the plot, we notice that customers sleep longer on Sunday and less on tuesday.

```
#calories burn in weekdays
dailyCalories$Calories_weekday<-weekdays(as.Date(dailyCalories$Activity_Date))
weeklyCalories<-dailyCalories %>% group_by(Calories_weekday)
WeekDay <- factor(weeklyCalories$Calories_weekday,level = c('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'
, 'Saturday'))


ggplot(weeklyCalories, aes(x = WeekDay, y = Calories))+
  geom_col()+labs(title="Daily Calories burn")
```



We'd like to know the user's calories burn pattern during the weekday. We need to add weekday column to the dailyactivity_daily calories. From the plot, we notice that customers daily calories nurn is more on tuesday and less on sunday.It shows user are more active on tuesday and less active on sunday.

#Final Conclusion

By analysing some of the trends from the data provided ,

- People who are more sedentary are walk less(less steps) and they burn less Calories and sleep less as well. Hence they are less active.
- People are more likely to spend their time in bed rather than they fell asleep.
- People who walk more are burning their calories more.
- People who sleep more are comparatively burning their calories more.
- Most people are less active during their weekends(saturday and sunday). They achieve more steps on weekdays especially Tuesday.
- Most people sleep more equally on all days.

#Recommendations

- Bellabeat can implement some step goal rewards for the people who achieved more steps thoughtout the week.
- Most of the customers are less active on weekends.Bellabeat could try to implement some features to motivate people on weekends.

- More importantly Bellabeat mainly focusses on women health ,Bellabeat could try to implement some features based on women health on their app(Here the data is not provided about women health).