Bellabeat Case Study

Abhishek Salaria

2022-05-23

Introduction

Bellabeat is a high-tech manufacturer of health-focused products for women. It is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company.

Business Requirement

In this case study we'll analyze smart device data to gain insight into how consumers are using their smart devices. The insights will then help guide marketing strategy for the company. Analysis will be shared to the Bellabeat executive team along with high-level recommendations for Bellabeat's marketing strategy.

Ask Phase

We are asked to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices.

Questions

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

Prepare

In this bussiness task, we will use FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): that 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.

Data Organization

The Dataset contains 18 csv files (15 long-format + 3 wide-format). Each file contains different information such as sleep, calories, steps, distance, heart rate, Intensities on different timelines such as second, min, hour and day. Also, daily activity file contains all the data contained in all daily-type files apart from sleep and weight data.

To make things easier we'll be focusing on hour and day time frame.

Data Limitations

1. It contains only 33 user samples.
2. It lacks demographic data of costumers.

Process

Loading Packages

```
library(tidyverse)
library(readr)
library(reshape2)
```

Loading Datasets

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

## Rows: 940 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────

## Delimiter: ","
```

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

```
hourlyIntensities <- read_csv("hourlyIntensities_merged.csv")
```

```
## Rows: 22099 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
##     Use `spec()` to retrieve the full column specification for this data.
##     Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Preprocessing hourlyCalories

Preprocessing hourlyCalories to seperate date, day and hour from activity date.

```
hourlyCalories$ActivityDate <- strptime(hourlyCalories$ActivityHour,
                                        "%m/%d/%Y %I:%M:%S %p") %>%
  format("%m/%d/%Y") %>%
  as.Date(format = "%m/%d/%Y")

hourlyCalories$ActivityDay <- weekdays(as.Date(hourlyCalories$ActivityDate))

hourlyCalories$ActivityHour <- strptime(hourlyCalories$ActivityHour,
                                        "%m/%d/%Y %I:%M:%S %p") %>%
  format("%H:%M:%S")

hourlyCalories <- hourlyCalories %>%
  relocate(Calories, .after = ActivityDay)
```

Preprocessing sleepDay

Preprocessing SleepDay to convert the date to weekday format.

```
sleepDay$SleepDay <- strptime(sleepDay$SleepDay,
                                        "%m/%d/%Y %I:%M:%S %p") %>%
  format("%m/%d/%Y") %>%
  as.Date(format = "%m/%d/%Y")


sleepDay$SleepDay <- weekdays(as.Date(sleepDay$SleepDay))
```

Preprocessing hourlyIntensities

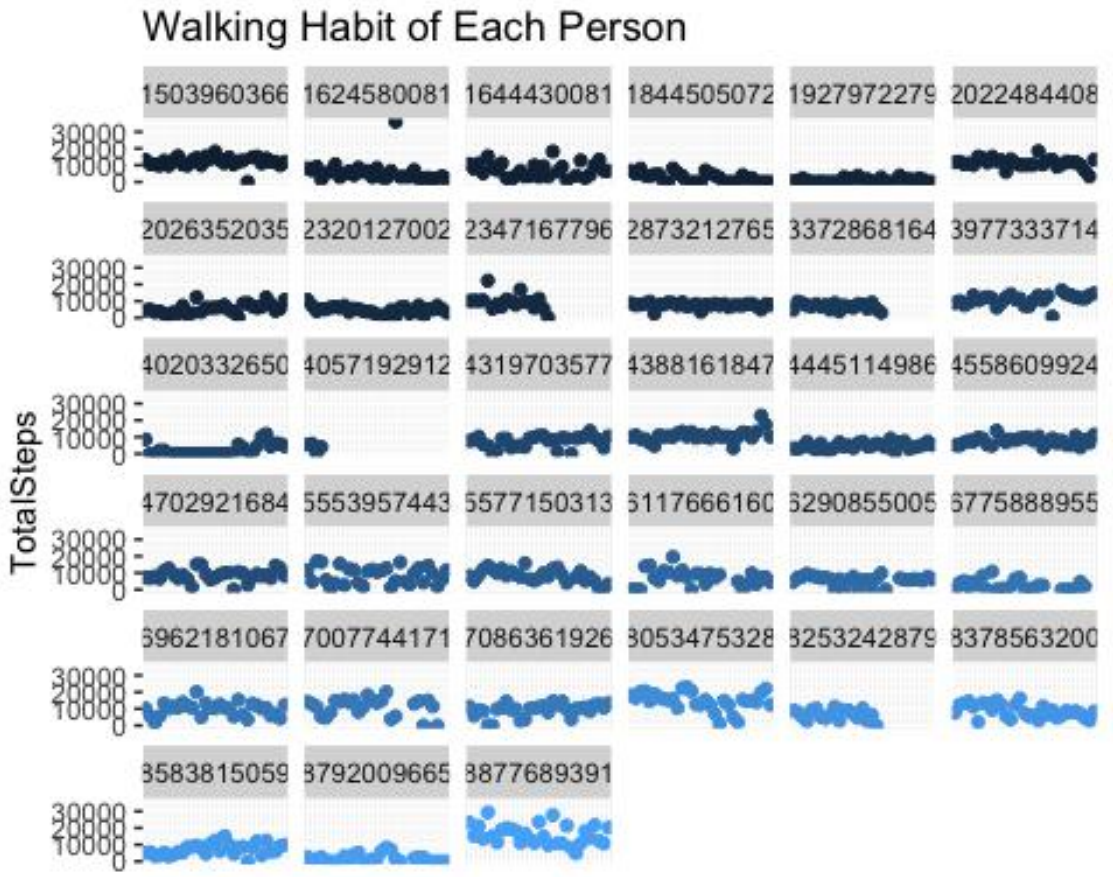
Preprocessing hourlyIntensities to convert ActivityHour to hms format.

```
hourlyIntensities$ActivityHour <- strptime(hourlyIntensities$ActivityHour,
                                        "%m/%d/%Y %I:%M:%S %p") %>%
  format("%H:%M:%S")
```

Analysis and Share

Walking Habit of Each Person.

```
ggplot(dailyActivity,aes(x=ActivityDate,y=TotalSteps,color=Id)) + geom_point() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position="none") +
  facet_wrap(~Id) + theme(axis.title.x=element_blank(),
                          axis.text.x=element_blank(),
                          axis.ticks.x=element_blank()) +
  ggtitle("Walking Habit of Each Person")
```



Summary of Daily Activities

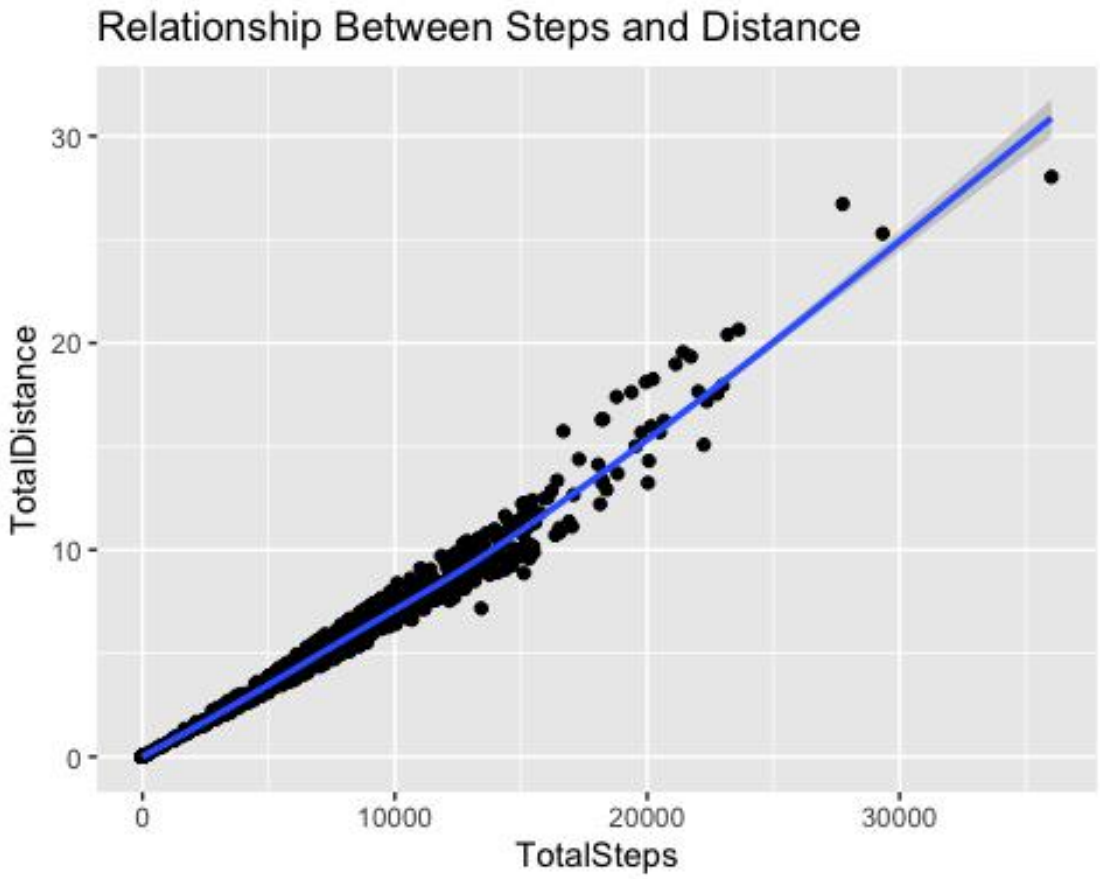
```
dailyActivity %>%
  select(TotalSteps,TrackerDistance,VeryActiveDistance,
         ModeratelyActiveDistance,LightActiveDistance, SedentaryActiveDistance,
         VeryActiveMinutes,FairlyActiveMinutes,LightlyActiveMinutes,
         SedentaryMinutes,Calories) %>%
  summary()
```

```
##      TotalSteps      TrackerDistance   VeryActiveDistance  ModeratelyActiveDistan
ce
##   Min.   :    0   Min.   : 0.000   Min.   : 0.000       Min.   :0.0000

##   1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 0.000       1st Qu.:0.0000

##   Median : 7406   Median : 5.245   Median : 0.210       Median :0.2400
```

```
##  Mean   : 7638   Mean   : 5.475   Mean   : 1.503     Mean   :0.5675

##  3rd Qu.:10727   3rd Qu.: 7.710   3rd Qu.: 2.053     3rd Qu.:0.8000

##  Max.   :36019   Max.   :28.030   Max.   :21.920     Max.   :6.4800

##  LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
##  Min.   : 0.000      Min.   :0.000000        Min.   :  0.00
##  1st Qu.: 1.945      1st Qu.:0.000000        1st Qu.:  0.00
##  Median : 3.365      Median :0.000000        Median :  4.00
##  Mean   : 3.341      Mean   :0.001606        Mean   : 21.16
##  3rd Qu.: 4.782      3rd Qu.:0.000000        3rd Qu.: 32.00
##  Max.   :10.710      Max.   :0.110000        Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0.0   Min.   :   0
##  1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8   1st Qu.:1828
##  Median :  6.00      Median :199.0        Median :1057.5   Median :2134
##  Mean   : 13.56      Mean   :192.8        Mean   : 991.2   Mean   :2304

##  3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5   3rd Qu.:2793
##  Max.   :143.00      Max.   :518.0        Max.   :1440.0   Max.   :4900
```

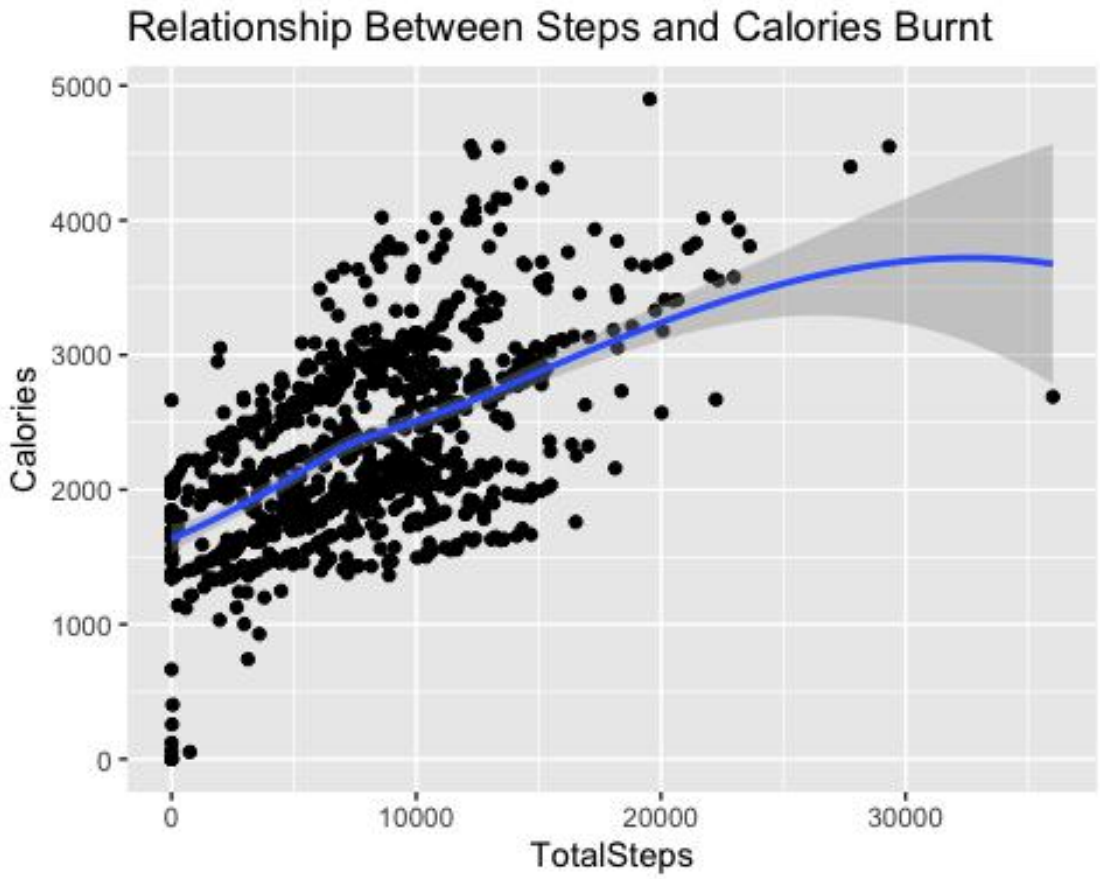
Positive Relationship between Total Steps & Total Distance

```
ggplot(dailyActivity,aes(x=TotalSteps,y=TotalDistance)) + geom_point() + geom_s
mooth() +
  ggtitle("Relationship Between Steps and Distance")

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



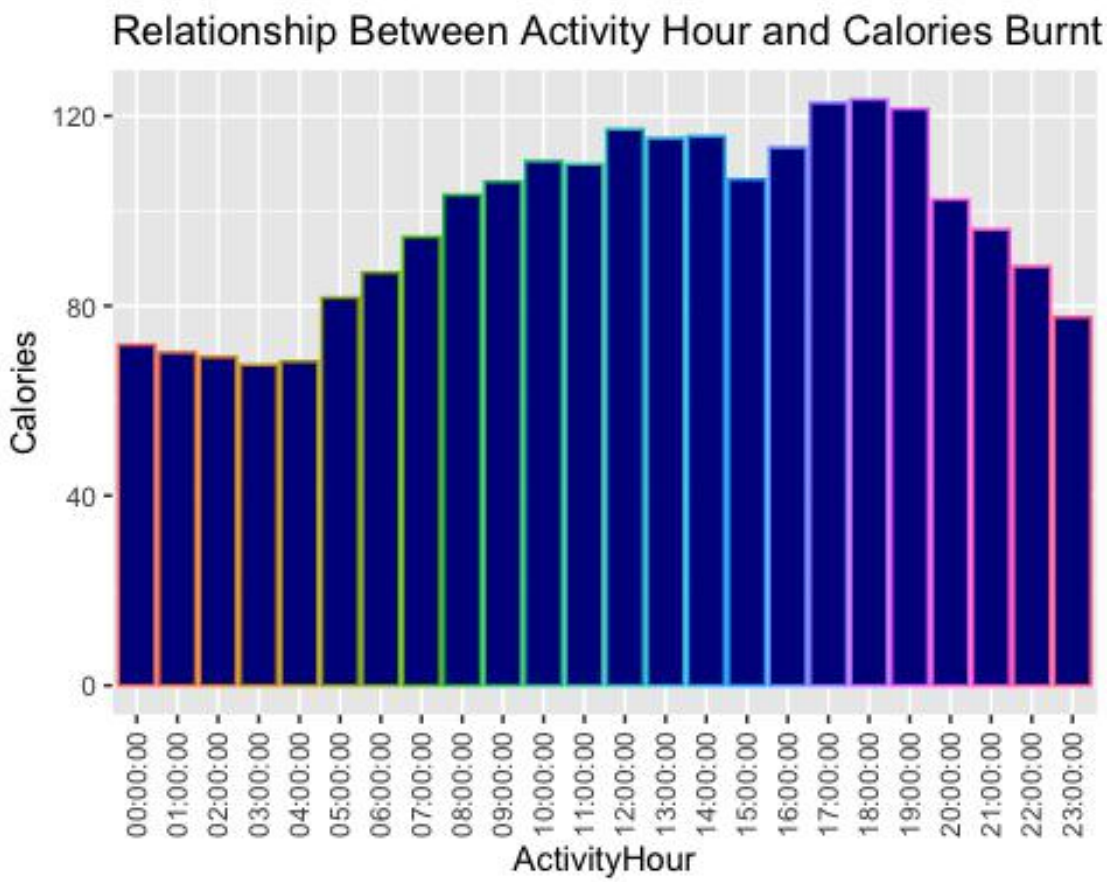
Positive Relationship between Total Steps & Calories Burnt

```
ggplot(dailyActivity,aes(x=TotalSteps,y=Calories)) + geom_point() + geom_smooth()
 +
  ggtitle("Relationship Between Steps and Calories Burnt")

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



Hourly Average Calories Burnt

```
average_hourly_Data <- hourlyCalories %>%
  select(ActivityHour,Calories) %>%
  group_by(ActivityHour) %>%
  summarise_all(.funs = mean)


ggplot(average_hourly_Data,aes(x=ActivityHour,y=Calories,color=ActivityHour)) +
  geom_bar(stat = "identity", fill='#00008B') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position="none") +
  ggtitle("Relationship Between Activity Hour and Calories Burnt")
```



Summary of Calories Burnt Hourly

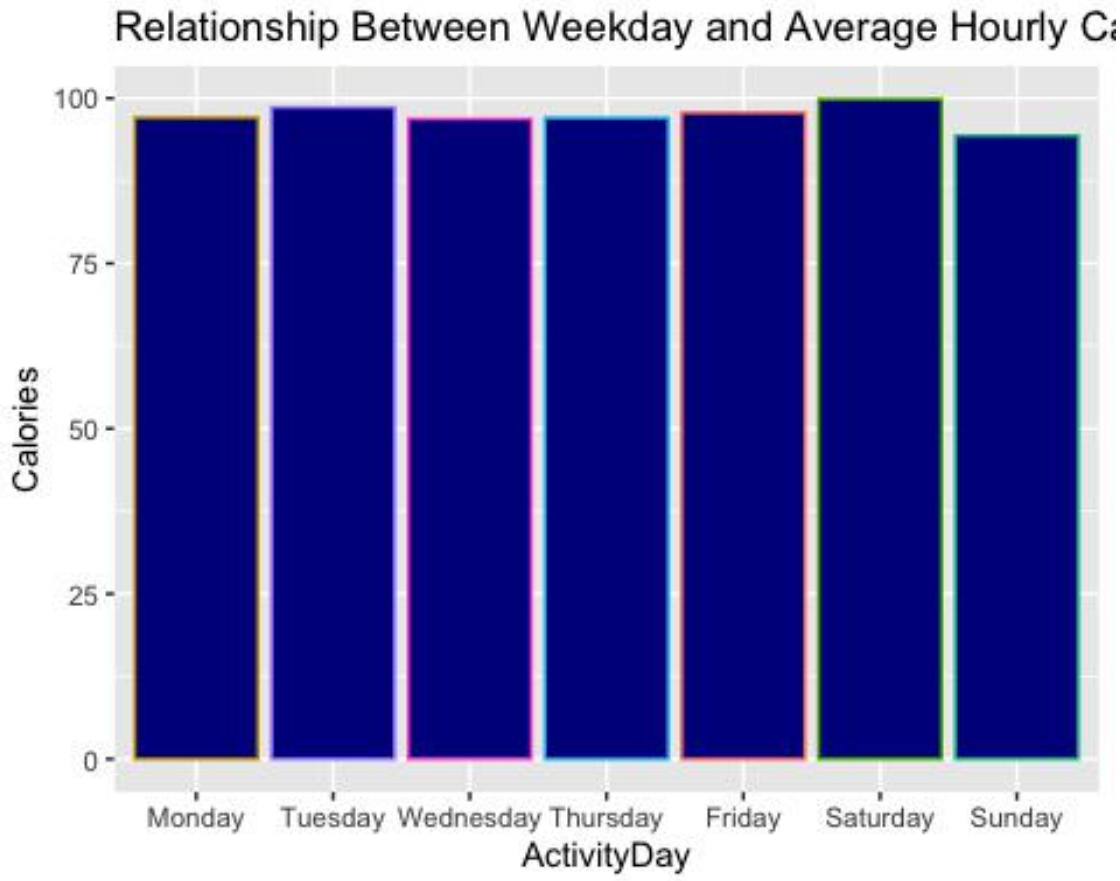
```
average_hourly_Data %>%
  select(Calories) %>%
  summary()
```

```
##     Calories
##  Min.   : 67.54
##  1st Qu.: 80.68
##  Median :102.85
##  Mean   : 97.50
##  3rd Qu.:113.82
##  Max.   :123.49
```

Daily Average Calories Burnt

```
average_daily_Data <- hourlyCalories %>%
  select(ActivityDay,Calories) %>%
```

```
  group_by(ActivityDay) %>%
  summarise_all(.funs = mean)


ggplot(average_daily_Data,aes(x=ordered(ActivityDay,
                                        levels=c("Monday", "Tuesday", "Wednes
day", "Thursday",
                                                 "Friday", "Saturday", "Sunday
")),
                              y=Calories,color=ActivityDay)) +
  geom_bar(stat = "identity", fill='#00008B') + labs(x = "ActivityDay") +
  theme(legend.position="none") +
  ggtitle("Relationship Between Weekday and Average Hourly Calories Burnt")
```



Summary of Calories Burnt Daily

```
average_daily_Data %>%
  select(Calories) %>%
  summary()
```

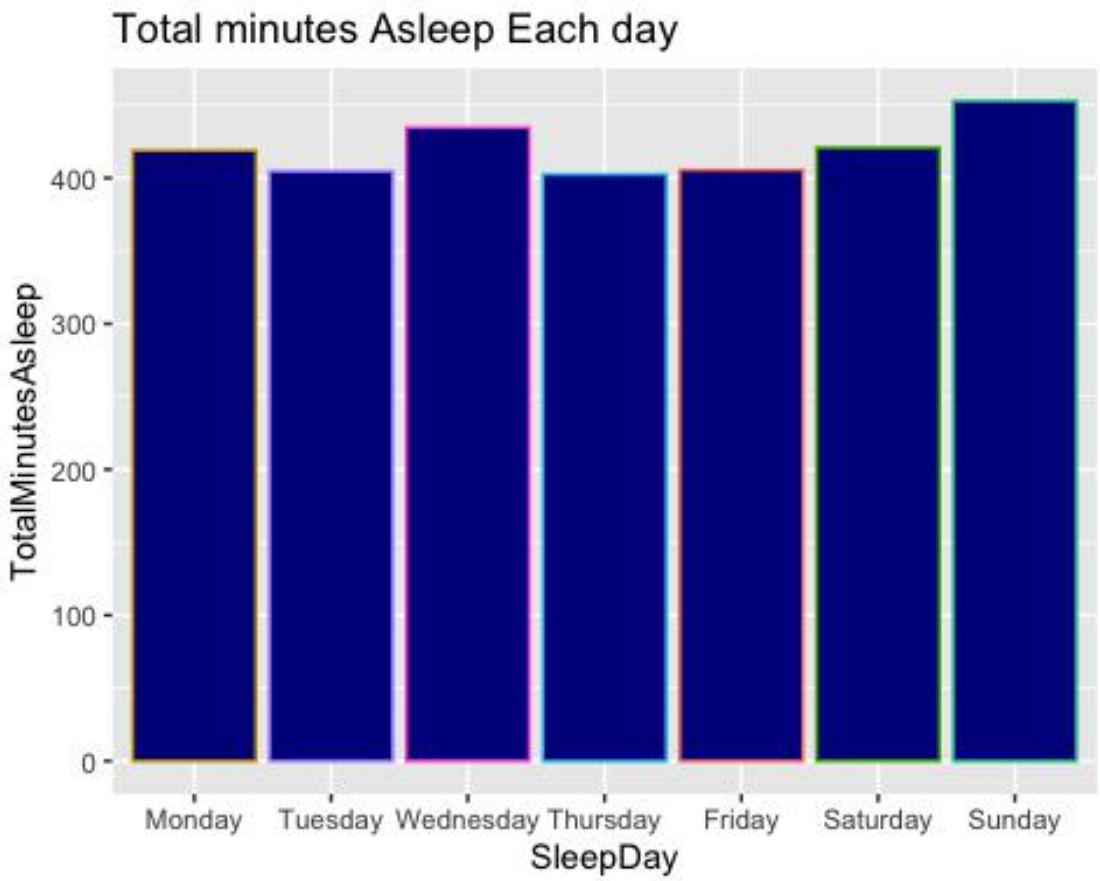
```
##        Calories
##    Min.    :94.34
##    1st Qu.:96.94
##    Median :97.05
##    Mean    :97.36
##    3rd Qu.:98.20
##    Max.    :99.87
```

Daily Sleeping Habit

```
average_sleep_Data <- sleepDay %>%
  select(SleepDay,TotalMinutesAsleep,TotalTimeInBed) %>%
  group_by(SleepDay) %>%
  summarise_all(.funs = mean)
average_sleep_Data$TimeTakenToSleep <- average_sleep_Data$TotalTimeInBed -
 average_sleep_Data$TotalMinutesAsleep
```

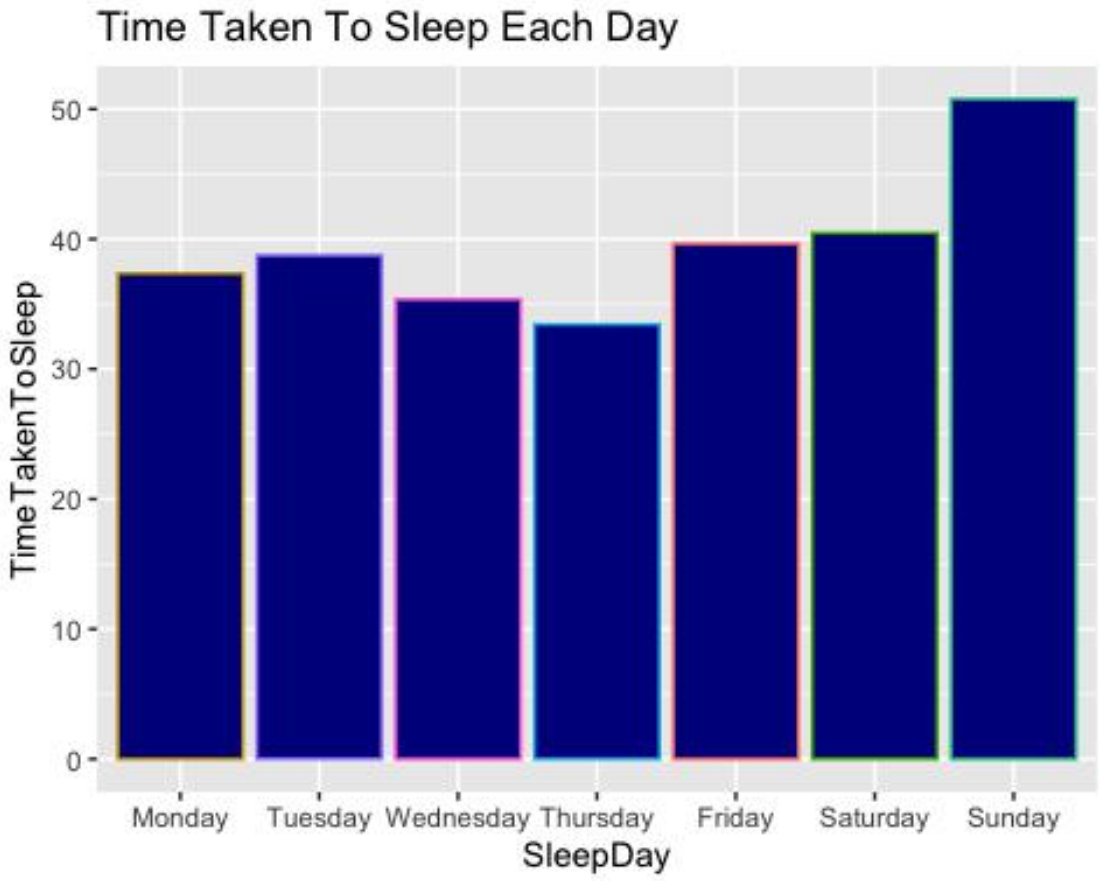
Total Minutes Slept Each day

```
ggplot(average_sleep_Data,aes(x=ordered(SleepDay,
                                        levels=c("Monday", "Tuesday", "Wednes
day", "Thursday",
                                                 "Friday", "Saturday", "Sunday
")),
                              y=TotalMinutesAsleep,color=SleepDay)) +
  geom_bar(stat = "identity", fill='#00008B') + labs(x = "SleepDay") +
  theme(legend.position="none") +
  ggtitle("Total minutes Asleep Each day")
```



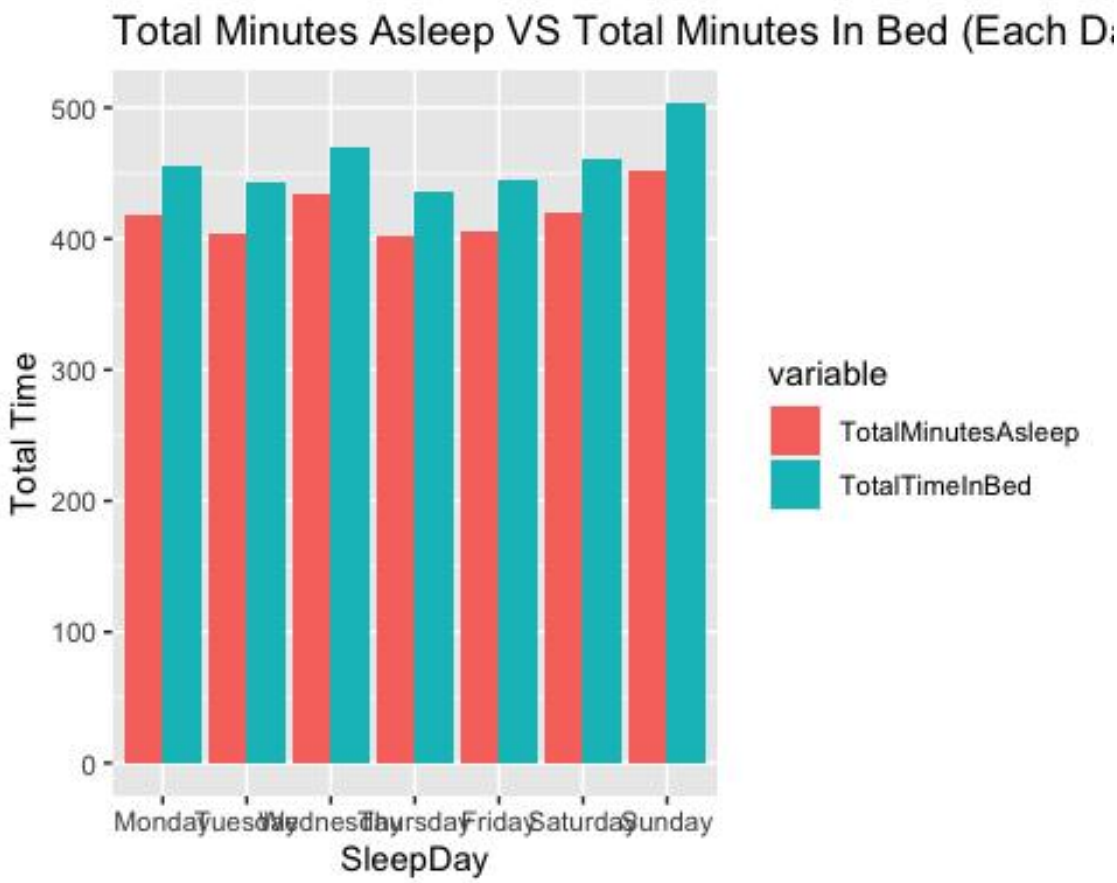
Total Time Taken to Sleep Each day

```
ggplot(average_sleep_Data,aes(x=ordered(SleepDay,
                                        levels=c("Monday", "Tuesday", "Wednes
day", "Thursday",
                                                 "Friday", "Saturday", "Sunday
")),
                              y=TimeTakenToSleep,color=SleepDay)) +
  geom_bar(stat = "identity", fill='#00008B') + labs(x = "SleepDay") +
  theme(legend.position="none") +
  ggtitle("Time Taken To Sleep Each Day")
```



Total Minutes Slept vs Total Time in Bed Each day

```
ggplot(melt(average_sleep_Data[,c('SleepDay','TotalMinutesAsleep','TotalTimeInBed
')]
               ,id.vars = 1),aes(x = ordered(SleepDay,
                                             levels=c("Monday", "Tuesday", "Wednesday",
 "Thursday",
                                                      "Friday", "Saturday", "Sunday"))
                                 ,y = value)) +
  geom_bar(aes(fill = variable),stat = "identity",position = "dodge") +
  labs(x = "SleepDay", y = "Total Time") +
  ggtitle("Total Minutes Asleep VS Total Minutes In Bed (Each Day)")
```



Summary of Sleeping Habits

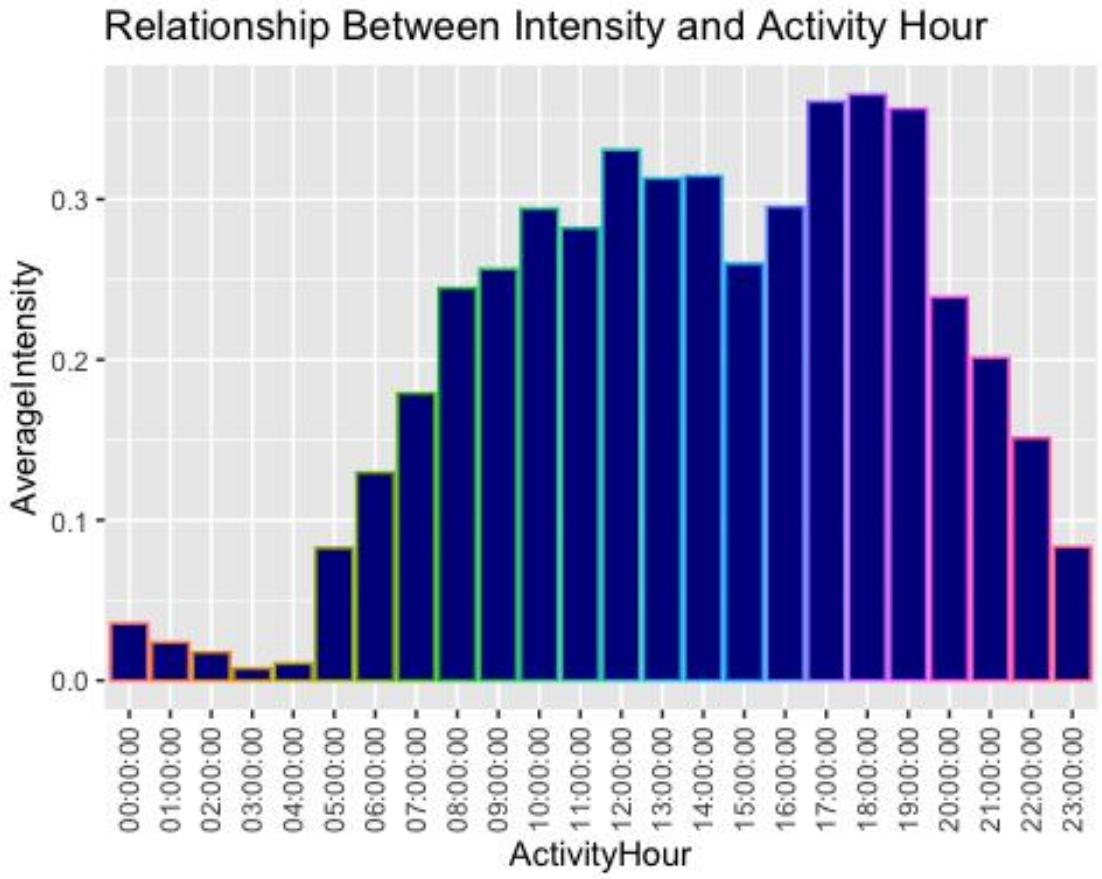
```
average_sleep_Data %>%
  select(TotalMinutesAsleep,TotalTimeInBed,TimeTakenToSleep) %>%
  summary()
```

```
##  TotalMinutesAsleep TotalTimeInBed  TimeTakenToSleep
##  Min.   :402.4      Min.   :435.8   Min.   :33.43
##  1st Qu.:405.0      1st Qu.:444.2   1st Qu.:36.34
##  Median :418.8      Median :456.2   Median :38.75
##  Mean   :419.9      Mean   :459.3   Mean   :39.39
##  3rd Qu.:427.7      3rd Qu.:465.7   3rd Qu.:40.05
##  Max.   :452.7      Max.   :503.5   Max.   :50.76
```

Hourly Intensities

```
average_Intensities_Data <- hourlyIntensities %>%
  select(ActivityHour,TotalIntensity,AverageIntensity) %>%
```

```
  group_by(ActivityHour) %>%
  summarise_all(.funs = mean)


ggplot(average_Intensities_Data,aes(x=ActivityHour,y=AverageIntensity, color=Activi
tyHour)) +
  geom_bar(stat = "identity", fill='#00008B') +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position="none") +
  ggtitle("Relationship Between Intensity and Activity Hour")
```



Conclusion

1) On Average People walks 7638 steps Everyday.

2) On Average People burns 2304 calories Everyday.

3) People Spend most time on bed on Sundays.

4) People Sleep for most time on Sundays.

5) People Burns Highest number of Calories on Saturdays.

6) People are highly active from 5:00–7:00 pm whereas they are least active from 2:00–4:00 am.

7) On an average people burns 97.50 calories each hour.

8) On an average people sleeps for 419.9 mins or 6.998 hrs (7 hrs Approximately) with minimum of 402.4 mins or 6.70 hrs and maximum of 452.7 mins or 7.545 hrs.

9) People are highly intensive from 5:00–7:00 pm whereas they are least intensive from 2:00–4:00 am (Similar to activity hours).

10) Approximately People takes 39.9 mins to sleep which is increased to 50.76 mins on Sundays.

Act

Recommendations

1) People walks 7638 steps everyday which can be increased to 10000 Steps as per the recommendations of CDC to lower mortality rate and improve health. So, through bellabeat app, peoples can be notified about the number of steps that is required to be completed and also the benefits associated with it.

2) On an average people are sleeping around 7 hours as per our analysis that is the minimum time an adult should sleep as per CDC, so anyone sleeping less than that should be sent notification about improving sleeping time and also some posts showing benefits of a good sleep.

3) Peoples taking more time to sleep can be suggested with some useful tips such as yoga or healthy eating habits so as to improve their sleep time and overall health.
4) People can be given inapp rewards such as titles, increasing levels etc., based on different milestones they achieve everyday such as completing the number of required steps each day so as to encourage them.

5) In addition to all above recommendations, age related information of users can be collected so that recommendation according to age can be made, like number of steps, sleep required for teenage, adult and old people.