Bellabeat Case Study

Rounak Saha

2024-06-03

Introduction

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist 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.

My report will include the following deliverables:

- A clear summary of the business task
- A description of all data sources used
- Documentation of any cleaning or manipulation of data
- A summary of your analysis
- Supporting visualizations and key findings
- Your top high-level content recommendations based on your analysis

Step 1: Ask

Business Task

To analyze one of Bellabeat's products and 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.

1.1 Key stakeholders:

- *Urška Sršen*: Bellabeat's cofounder and Chief Creative Officer
- *Sando Mur*: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
- *Bellabeat marketing analytics team*: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

1.2 Questions to explore for the 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?

Step 2: Prepare

The data being used in this case study can be found here: FitBit Fitness Tracker Data CC0: Public Domain, dataset made available through Mobius

The data is stored and uploaded in R Studio. 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 data set contains 18 CSV files organized in long format. Below is a breakdown of the data using the ROCCC approach:

- Reliability - **LOW**: The data comes from 30 fitbit users with unknown demographics who consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring.
- Original - **LOW**: Third party data collected using Amazon Mechanical Turk.
- Comprehensive - **MED**: The dataset contains multiple fields on daily activity intensity, calories used, daily steps taken, daily sleep time and weight record.
- Current - **LOW**: This data is from March 2016 through May 2016. The data is not current, meaning that user habits may have changed over the years.
- Cited - **LOW**: Data was collected from a third party, therefore unknown.

Step 3: Process

We will be installing and loading all necessary packages for data wrangling

```
install.packages("tidyverse", repos="https://cloud.r-project.org/")
install.packages("readr", repos="https://cloud.r-project.org/")
install.packages("dplyr", repos="https://cloud.r-project.org/")
install.packages("tidyr", repos="https://cloud.r-project.org/")
install.packages("ggplot2", repos="https://cloud.r-project.org/")
install.packages("lubridate",repos="https://cloud.r-project.org/")
install.packages("sqldf",repos="https://cloud.r-project.org/")
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)
```

```r
library(utils)
library(sqldf)
```

```
## Loading required package: gsubfn
## Loading required package: proto
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared object '/Library
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 0x0006): Library not loaded: /
##   Referenced from: <B3716E5A-BF4D-3CA3-B8EB-89643DB72A04> /Library/Frameworks/R.framework/Versions/4
##   Reason: tried: '/opt/X11/lib/libSM.6.dylib' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/
```

```
## tcltk DLL is linked to '/opt/X11/lib/libX11.6.dylib'
## Could not load tcltk.  Will use slower R code instead.
## Loading required package: RSQLite
```

```
## Warning: package 'RSQLite' was built under R version 4.3.3
```

```r
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

We will now load the dataframes using `read.csv()` function

```r
activity <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mturki

calories <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mturki

intensities <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mtu

steps <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mturkfitb

heart_rate <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mtur

sleep <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mturkfitb

weight <- read.csv("~/Desktop/DA prac/Tableau + SQL +EXCEL project/Bellabeat case study/archive/mturkfi
```

Lets take a look at the dataframes we loaded using `glimpse()` function

```r
glimpse(activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <int> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
```

```
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
glimpse(calories)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4/16/~
## $ Calories    <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 1786, 1775~
```

```
glimpse(heart_rate)
```

```
## Rows: 2,483,658
## Columns: 3
## $ Id    <dbl> 2022484408, 2022484408, 2022484408, 2022484408, 2022484408, 2022~
## $ Time  <chr> "4/12/2016 7:21:00 AM", "4/12/2016 7:21:05 AM", "4/12/2016 7:21:~
## $ Value <int> 97, 102, 105, 103, 101, 95, 91, 93, 94, 93, 92, 89, 83, 61, 60, ~
```

```
glimpse(intensities)
```

```
## Rows: 940
## Columns: 10
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDay              <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ VeryActiveMinutes        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
```

```
glimpse(sleep)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <int> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <int> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <int> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```r
glimpse(steps)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4/16/~
## $ StepTotal   <int> 13162, 10735, 10460, 9762, 12669, 9705, 13019, 15506, 1054~
```

```r
glimpse(weight)
```

```
## Rows: 67
## Columns: 8
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212~
## $ Date           <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, ~
## $ WeightPounds   <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6~
## $ Fat            <int> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,~
## $ IsManualReport <chr> "True", "True", "False", "True", "True", "True", "True"~
## $ LogId          <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,~
```

According to observation we can see that *calories*, *intensities* and *steps* are a subset of *calories* dataframe. In order to be sure, we can check using `sqldf()` function

```r
# check for calories

check_calories <- sqldf("select Id,ActivityDate,Calories
               from activity
               intersect
               select Id,ActivityDay,Calories
               from calories")
head(check_calories)
```

```
##           Id ActivityDate Calories
## 1 1503960366    4/12/2016     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
```

```r
nrow(check_calories)
```

```
## [1] 940
```

```r
# for steps
check_steps <- sqldf("select Id,ActivityDate, totalsteps
                     from activity
                     intersect
                     select Id,ActivityDay,Steptotal
                     from steps")
head(check_steps)
```

```
##           Id ActivityDate TotalSteps
## 1 1503960366    4/12/2016      13162
## 2 1503960366    4/13/2016      10735
## 3 1503960366    4/14/2016      10460
## 4 1503960366    4/15/2016       9762
## 5 1503960366    4/16/2016      12669
## 6 1503960366    4/17/2016       9705
```

```
nrow(check_steps)
```

```
## [1] 940
```

```
# for intensities
check_intensities <- sqldf("select Id, ActivityDay, SedentaryMinutes, LightlyActiveMinutes, FairlyActiv
                           from intensities
                           intersect
                           select Id, ActivityDate, SedentaryMinutes, LightlyActiveMinutes, FairlyActiv
                           from activity")
head(check_intensities)
```

```
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   4/12/2016              728                  328
## 2 1503960366   4/13/2016              776                  217
## 3 1503960366   4/14/2016             1218                  181
## 4 1503960366   4/15/2016              726                  209
## 5 1503960366   4/16/2016              773                  221
## 6 1503960366   4/17/2016              539                  164
##   FairlyActiveMinutes VeryActiveMinutes SedentaryActiveDistance
## 1                  13                25                       0
## 2                  19                21                       0
## 3                  11                30                       0
## 4                  34                29                       0
## 5                  10                36                       0
## 6                  20                38                       0
##   LightActiveDistance ModeratelyActiveDistance VeryActiveDistance
## 1                6.06                     0.55               1.88
## 2                4.71                     0.69               1.57
## 3                3.91                     0.40               2.44
## 4                2.83                     1.26               2.14
## 5                5.04                     0.41               2.71
## 6                2.51                     0.78               3.19
```

```
nrow(check_intensities)
```

```
## [1] 940
```

All of the 3 checks returned 940 row, so we can conclude, these are the subsets of *activity* data frame and can remove them for simplicity

```
rm(check_calories,check_intensities,check_steps, calories, intensities, steps)
```

Step 4: Analyse

To beging the analysis phase, we will first see how many participants there are in each category.

Checking the sample size of our dataframe

```
n_distinct(activity$Id)
```

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

```
n_distinct(heart_rate$Id)
```

```
## [1] 14
```

```
n_distinct(sleep$Id)
```

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

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

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

Checking significant changes in weight and BMI

```
weight %>% group_by(Id) %>% summarise(min(WeightKg), max(WeightKg), min(BMI), max(BMI))
```

```
## # A tibble: 8 x 5
##           Id `min(WeightKg)` `max(WeightKg)` `min(BMI)` `max(BMI)`
##        <dbl>           <dbl>           <dbl>      <dbl>      <dbl>
## 1 1503960366            52.6            52.6       22.6       22.6
## 2 1927972279           134.            134.        47.5       47.5
## 3 2873212765            56.7            57.3       21.5       21.7
## 4 4319703577            72.3            72.4       27.4       27.5
## 5 4558609924            69.1            70.3       27         27.5
## 6 5577150313            90.7            90.7       28         28
## 7 6962181067            61              62.5       23.8       24.4
## 8 8877689391            84              85.8       25.1       25.7
```

We will not include *heart_rate* and *weight* data frames in our analysis since they have very less sample size from our observation above using `n_distinct()` function and there is hardly any changes in weight for the participants. 8 and 14 participants are not enough to draw strong conclusion based on this data. Rather, sample size for *sleep* dataframe is also small but we will keep it for reference.

```
# Removing heart_rate and weight data frames
rm(heart_rate, weight)
```

```
# Checking for duplicates
sum(duplicated(activity))
```

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

```r
sum(duplicated(sleep))
```

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

```r
#Remove duplicates
sleep <- unique(sleep)
sum(duplicated(sleep))
```

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

```r
# Checking NA's
sum(is.na(activity))
```

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

```r
sum(is.na(sleep))
```

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

Since R is a case-sensitive, for simplicity we can change the variable names from camel-case to lowecase.

```r
# Formatting variable names
activity <- rename_with(activity, tolower)

sleep <- rename_with(sleep, tolower)
```

We need to change the data type of date column since it is in character using `as.date()` function.

```r
# Changing date format
activity$date <-as.Date(activity$activitydate, format = "%m/%d/%Y")

sleep$date<- as.Date(sleep$sleepday, format = "%m/%d/%Y")
```

```r
# Lets look at summary statistics of the activity dataset
activity %>%
select(totalsteps,calories,veryactiveminutes,fairlyactiveminutes,lightlyactiveminutes,sedentaryminutes)
summary()
```

```
##    totalsteps       calories    veryactiveminutes fairlyactiveminutes
##  Min.   :    0   Min.   :   0   Min.   :  0.00    Min.   :  0.00
##  1st Qu.: 3790   1st Qu.:1828   1st Qu.:  0.00    1st Qu.:  0.00
##  Median : 7406   Median :2134   Median :  4.00    Median :  6.00
##  Mean   : 7638   Mean   :2304   Mean   : 21.16    Mean   : 13.56
##  3rd Qu.:10727   3rd Qu.:2793   3rd Qu.: 32.00    3rd Qu.: 19.00
##  Max.   :36019   Max.   :4900   Max.   :210.00    Max.   :143.00
##  lightlyactiveminutes sedentaryminutes
##  Min.   :  0.0        Min.   :   0.0
##  1st Qu.:127.0        1st Qu.: 729.8
##  Median :199.0        Median :1057.5
##  Mean   :192.8        Mean   : 991.2
##  3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :518.0        Max.   :1440.0
```

```
# Lets look at summary statistics of the sleep dataset
sleep %>%
select(totalsleeprecords,totalminutesasleep) %>%
summary()
```

```
##  totalsleeprecords totalminutesasleep
##  Min.   :1.00      Min.   : 58.0
##  1st Qu.:1.00      1st Qu.:361.0
##  Median :1.00      Median :432.5
##  Mean   :1.12      Mean   :419.2
##  3rd Qu.:1.00      3rd Qu.:490.0
##  Max.   :3.00      Max.   :796.0
```

Some imporrtant discoveries were made from the summary:

- Average Sedentary time is 991 minutes
- Most of the population is lightly active
- Average participants sleep one time a day for 7 hours
- Average person burns 96kcal in an hour
- Daily average steps taken is 7638. The CDC recommends people to take 10000 steps a day

It is important to check the co-relation between sleep and activity because most of the times one is affected by other. In order to check that, we are going to merge the two data frames by the columns *Id* and *date*.

```
# Merging the dataframes
daily_activity_sleep <- merge(activity, sleep, by=c("id", "date"))
```

```
daily_activity_sleep %>%
group_by(id) %>%
count(id) %>%
  head()
```

```
## # A tibble: 6 x 2
## # Groups:   id [6]
##           id     n
##        <dbl> <int>
## 1 1503960366    25
## 2 1644430081     4
## 3 1844505072     3
## 4 1927972279     5
## 5 2026352035    28
## 6 2320127002     1
```

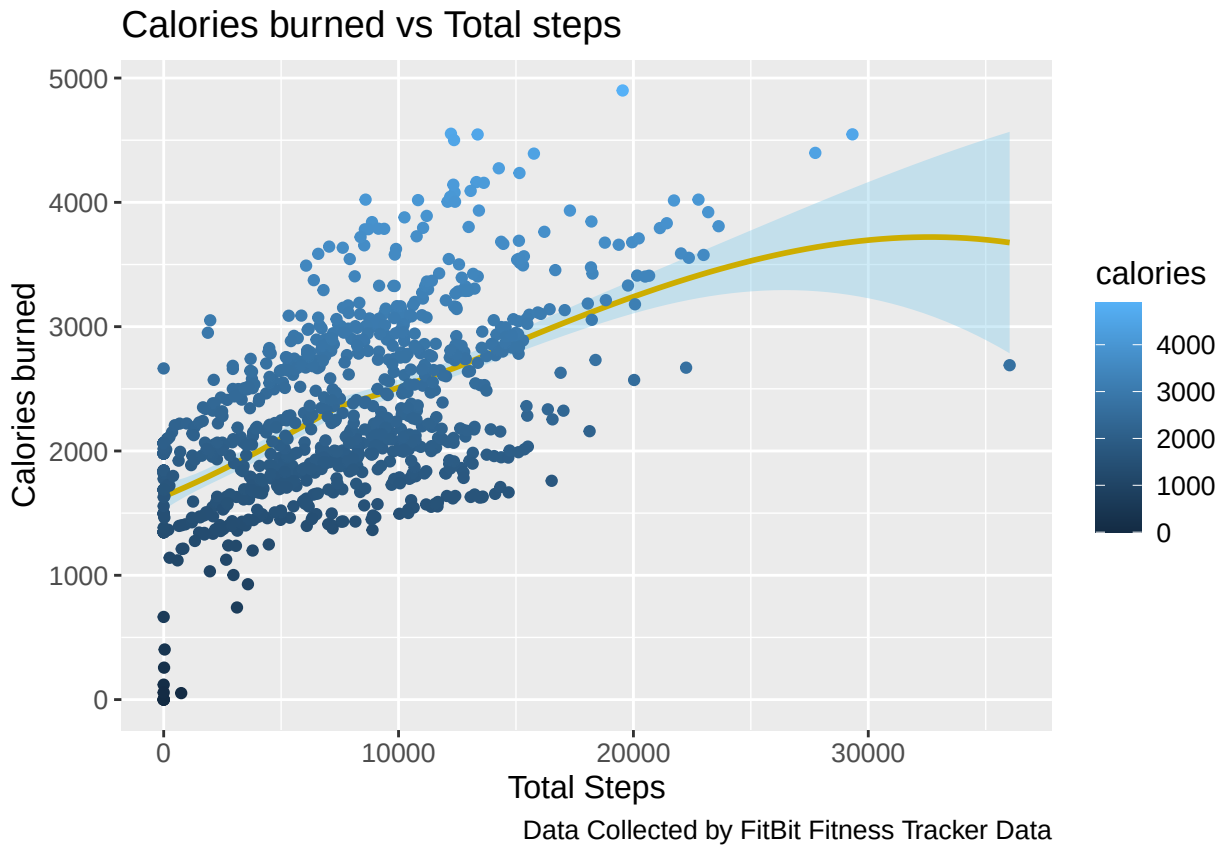
Step 5: Share

In this step, we will try to find trends and relationship among the participants by finding correlations between different variables.

Firstly, we will check the correlation between calories burned vs total steps taken

```
ggplot(activity) +
  aes(x= totalsteps, y= calories) +
  geom_smooth(color = 'gold3', fill= "skyblue")+
  geom_point(aes(color= calories)) +
  labs(x= 'Total Steps', y= 'Calories burned', title = "Calories burned vs Total steps",caption = "Data
  theme(text = element_text(size = 12), legend.position = 'right')
```

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

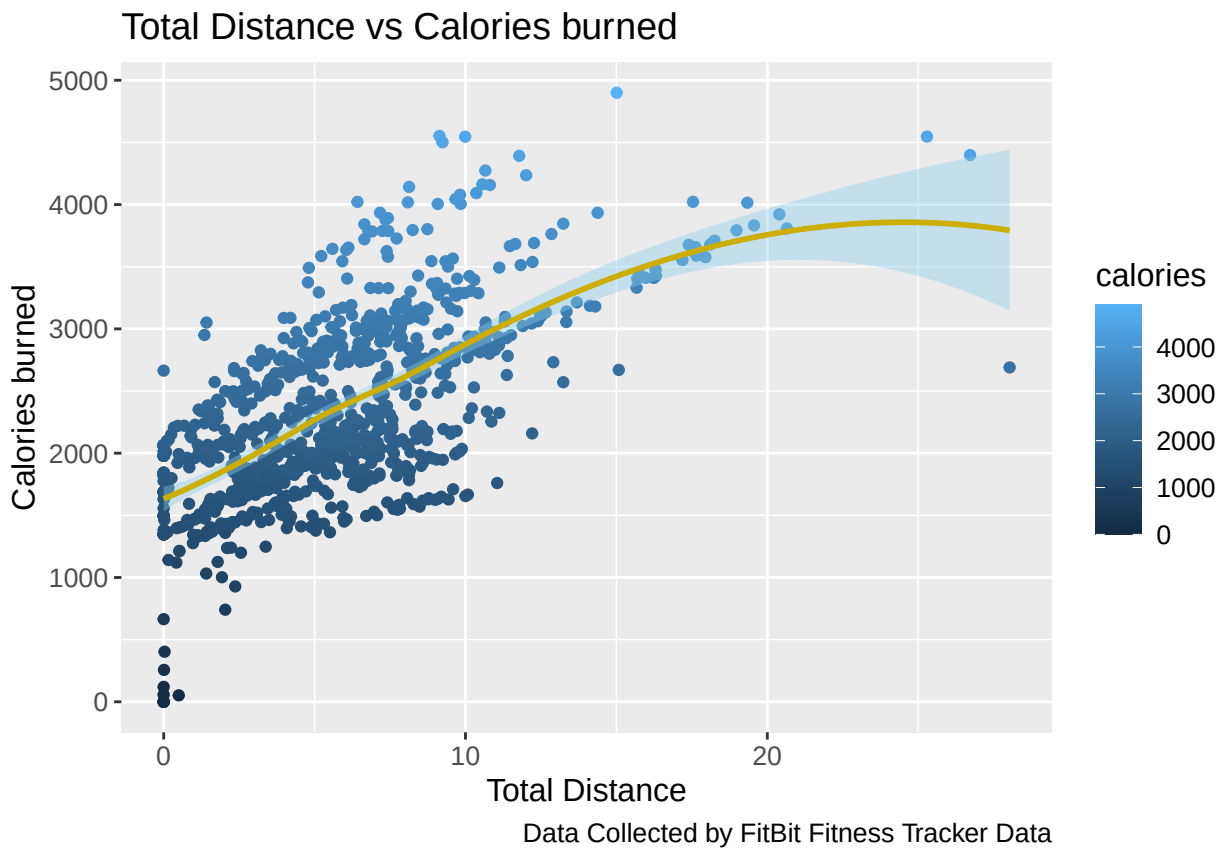


We can see a positive correlation between the two, which is obvious- the more active we are, the more calorie we burn.

Next, we will check the trends among total distance vs calories burned

```
ggplot(activity)+
  aes(x= totaldistance, y= calories)+
  geom_point(aes(colour = calories))+
  geom_smooth(color = 'gold3', fill='skyblue')+
  labs(x= 'Total Distance', y= 'Calories burned', title = 'Total Distance vs Calories burned', caption =
   theme(text = element_text(size = 12), legend.position = 'right')
```

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

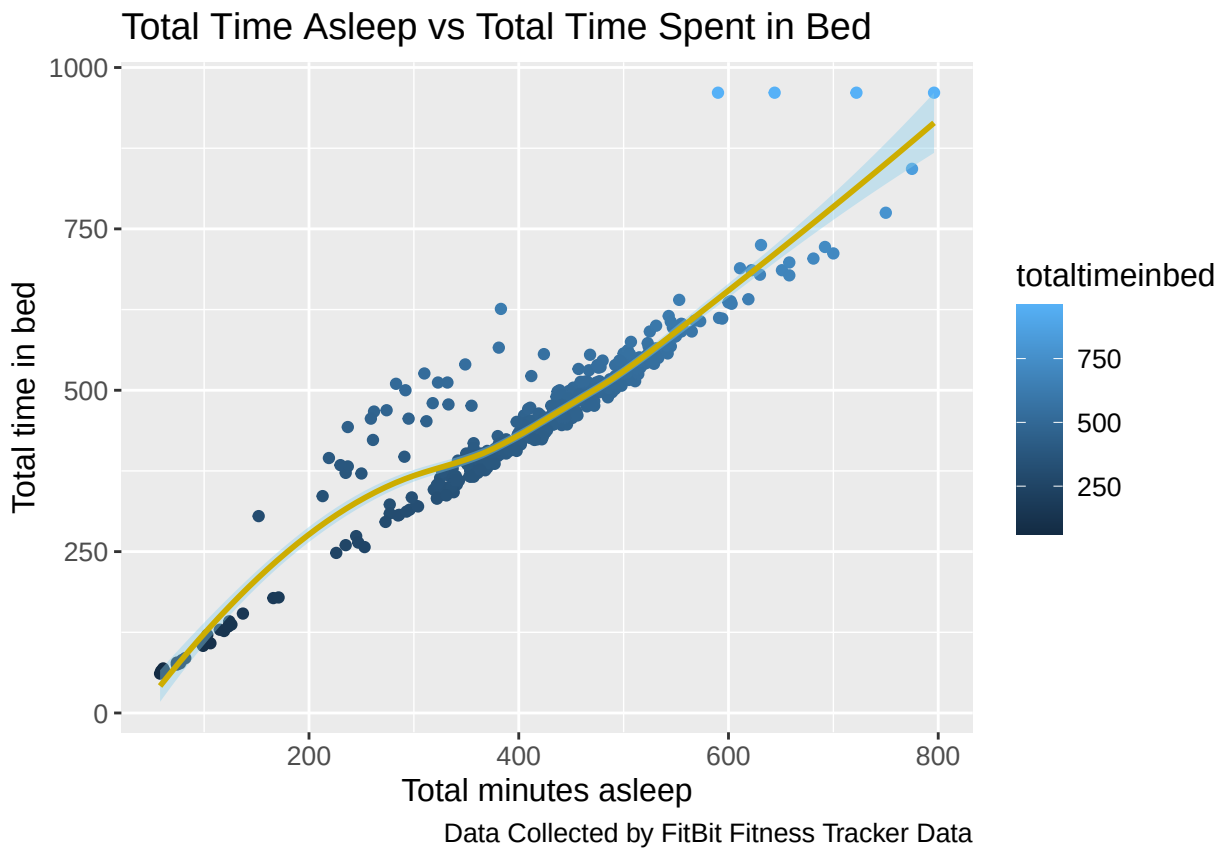


The plot displays a positive trend, indicating that a person tends to burn more calories when they move longer.

Next, we will look into the sleep dataset to understand the relation between the time asleep and total time spent in bed.

```
sleep %>%
  ggplot(aes(x= totalminutesasleep, y= totaltimeinbed))+
  geom_point(aes(colour = totaltimeinbed))+
  geom_smooth(color ='gold3',fill="skyblue")+
  labs(x= 'Total minutes asleep', y= 'Total time in bed', title = 'Total Time Asleep vs Total Time Spen
  theme(text = element_text(size = 12), legend.position = 'right')
```

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



The relationship between sleep time and total time in bed looks linear. So we can improve a feature to notify the users during their sleep time.

I intend to slice and dice the data set in order to get a deeper look into the average amount of calories, steps, distance and hours asleep recorded for each individual per weekday. This can uncover some interesting insights. First, I'll create a new data frame for this;

```
daily_activity_sleep_summary <- daily_activity_sleep %>%
  select(id, date, totalsteps, totaldistance,calories, totalminutesasleep)

head(daily_activity_sleep_summary)
```

```
##           id       date totalsteps totaldistance calories totalminutesasleep
## 1 1503960366 2016-04-12      13162          8.50     1985                327
## 2 1503960366 2016-04-13      10735          6.97     1797                384
## 3 1503960366 2016-04-15       9762          6.28     1745                412
## 4 1503960366 2016-04-16      12669          8.16     1863                340
## 5 1503960366 2016-04-17       9705          6.48     1728                700
## 6 1503960366 2016-04-19      15506          9.88     2035                304
```

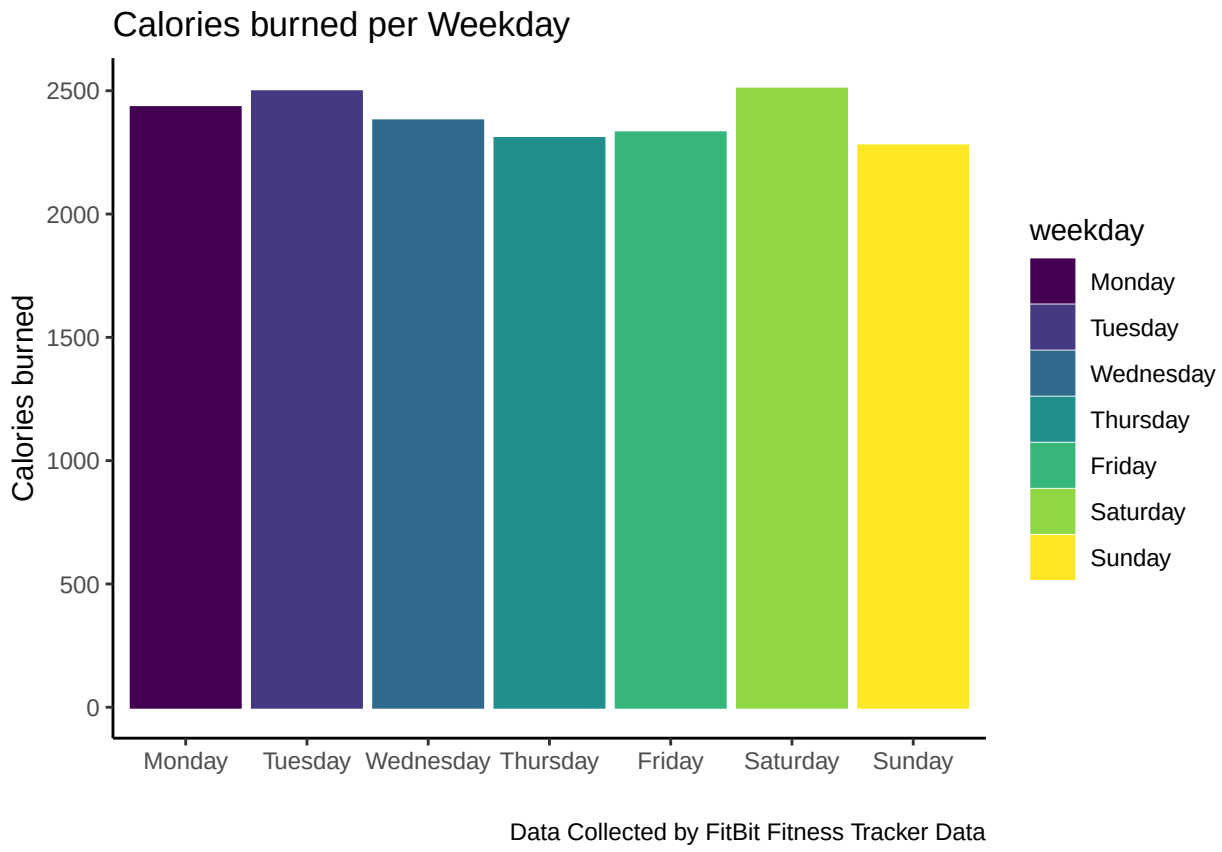
Weekday Summary:

```
weekday_summary <- daily_activity_sleep_summary %>%
  mutate(weekday= weekdays(date), sleep = totalminutesasleep/60)

weekday_summary$weekday <- ordered(weekday_summary$weekday, levels= c("Monday", "Tuesday", "Wednesday",
```

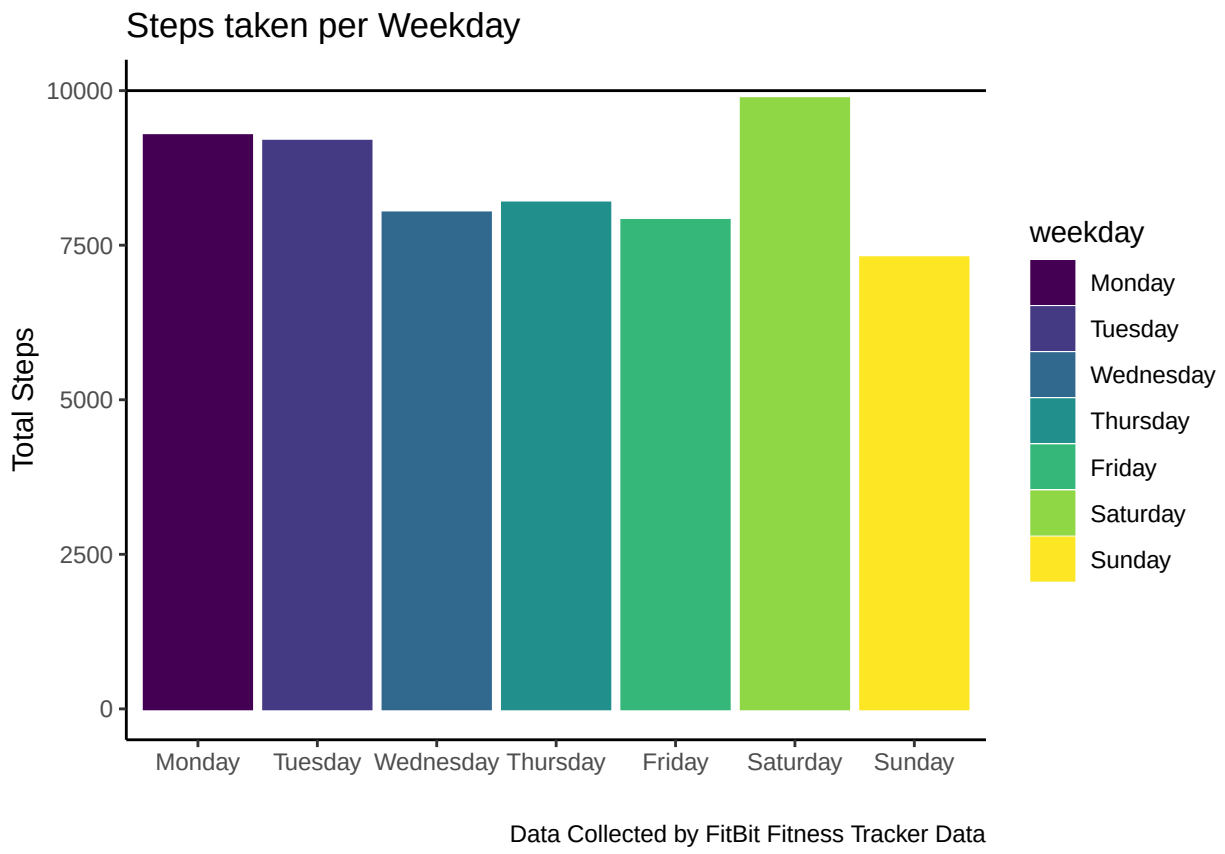
```
weekday_summary <- weekday_summary%>%
  group_by(weekday) %>%
  summarise( steps= mean(totalsteps), distance= mean(totaldistance), sleep= mean(sleep), calories= mean
```

Lets visualize the amount of calories burnt and steps taken per day

```
# Calories burned per weekday
ggplot(weekday_summary) +
  aes(x= weekday, y= calories)+
  geom_col(aes(color= weekday, fill= weekday))+
  labs(x= '', y= 'Calories burned', title = 'Calories burned per Weekday', caption = "Data Collected by
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+ theme_classic()
```

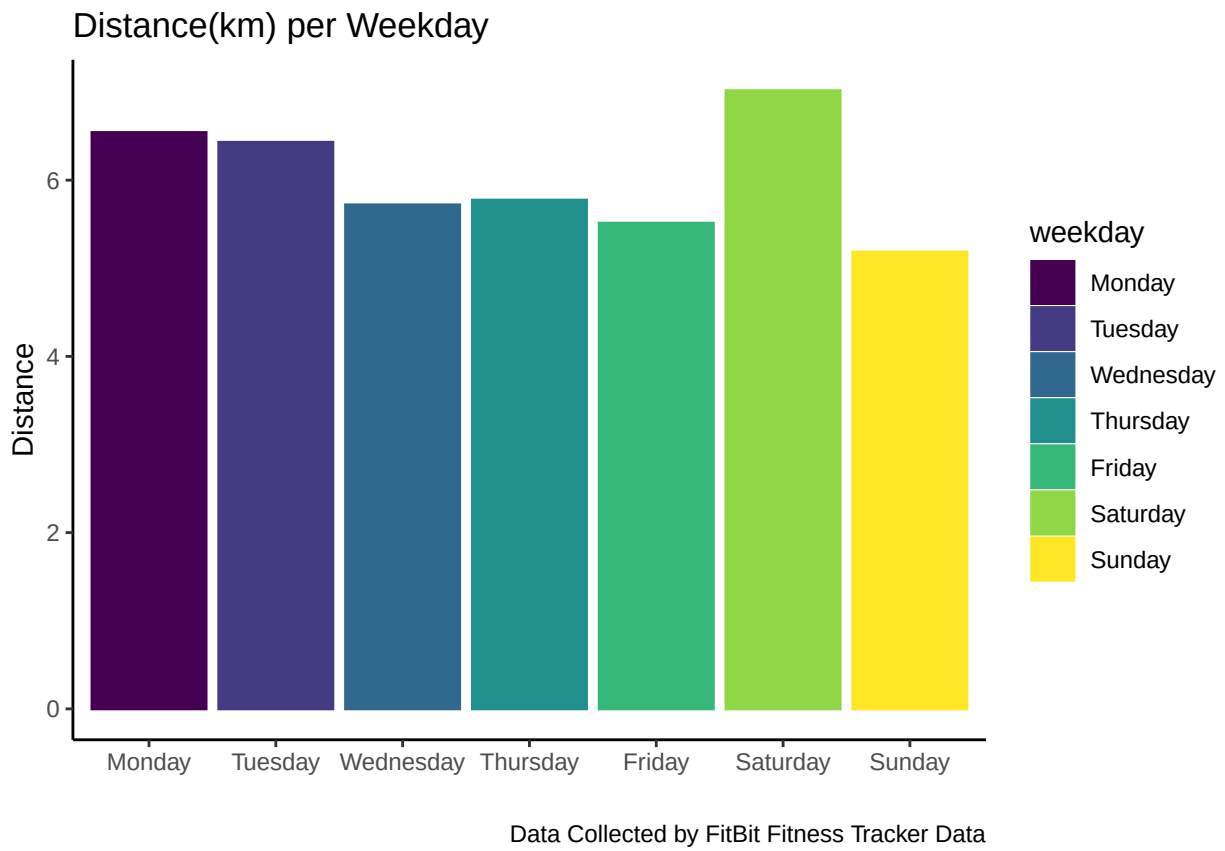



```
# Steps taken per weekday
ggplot(weekday_summary) +
  aes(x= weekday, y= steps)+
  geom_col(aes(color=weekday, fill=weekday))+
  labs(x= '', y= 'Total Steps',title = 'Steps taken per Weekday', caption = "Data Collected by FitBit Fi
theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  geom_hline(yintercept = 10000)+ theme_classic()
```

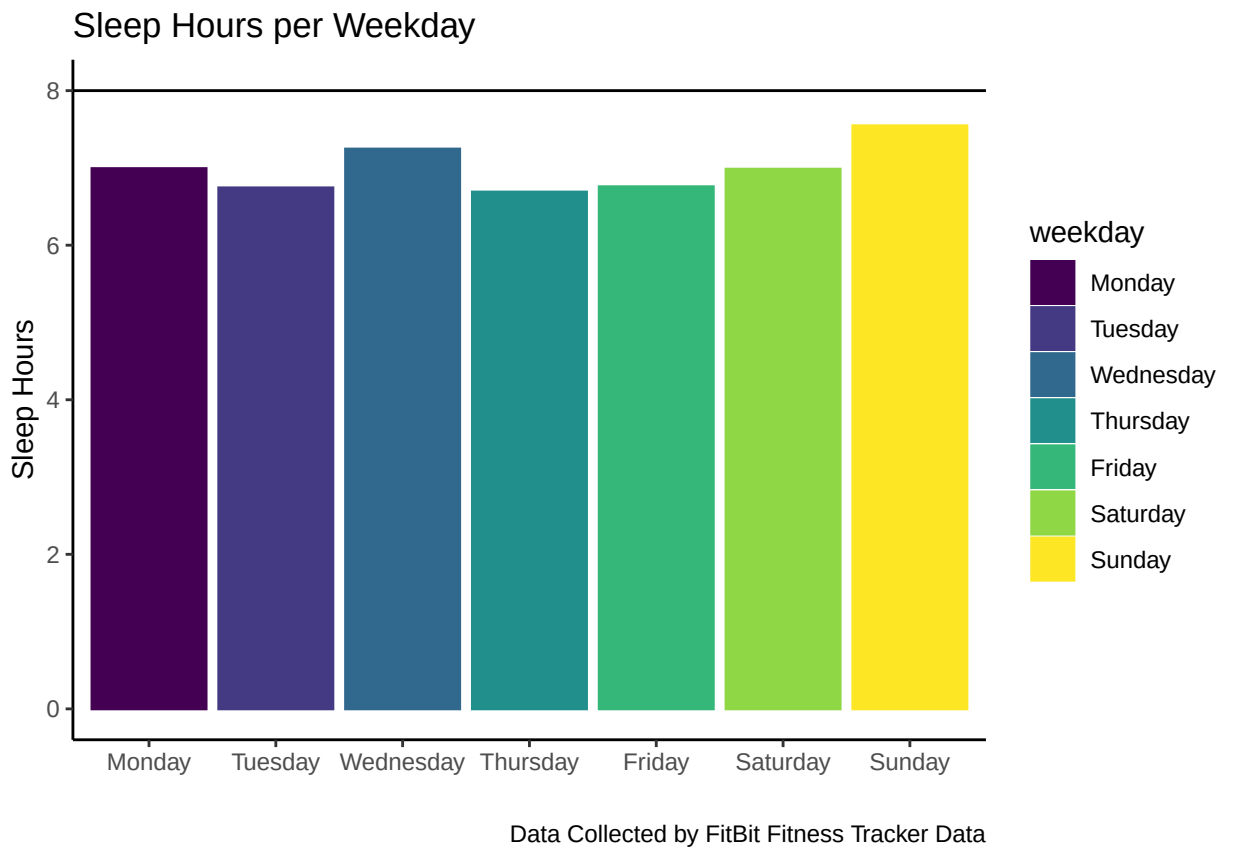


Now lets visualize the distance covered and sleep hours per weekday

```
# Distance per weekday
ggplot(weekday_summary)+
  aes(x=weekday, y= distance)+
  geom_col(aes(fill = weekday, color= weekday))+
  labs(x= "", y="Distance", title = "Distance(km) per Weekday", caption = "Data Collected by FitBit Fitn
  theme(axis.text.x = element_text(angle = 45, hjus= 1))+ theme_classic()
```




```
# Sleep hours per weekday
ggplot(weekday_summary)+
  aes(x= weekday, y=sleep)+
  geom_col(aes(colour = weekday, fill = weekday))+
  labs(x= "", y="Sleep Hours", title = "Sleep Hours per Weekday", caption = "Data Collected by FitBit Fi
  theme(axis.text.x = element_text(angle = 45, hjus= 1))+
  geom_hline(yintercept = 8)+ theme_classic()
```



From the above summary graph we can say the following:

- Users burn more calories on Monday, Wednesday and Sunday
- Users takes more steps on Monday, Tuesday and Wednesday still which is less than the count recommended by CDC
- Distance covered correlates with calories burned and steps taken
- User are getting less than 8 hours of recommended sleep

Some interesting inights can be drawn from how often users use smart devices. To this end, we will first group users according to the number of days they used smart devices.

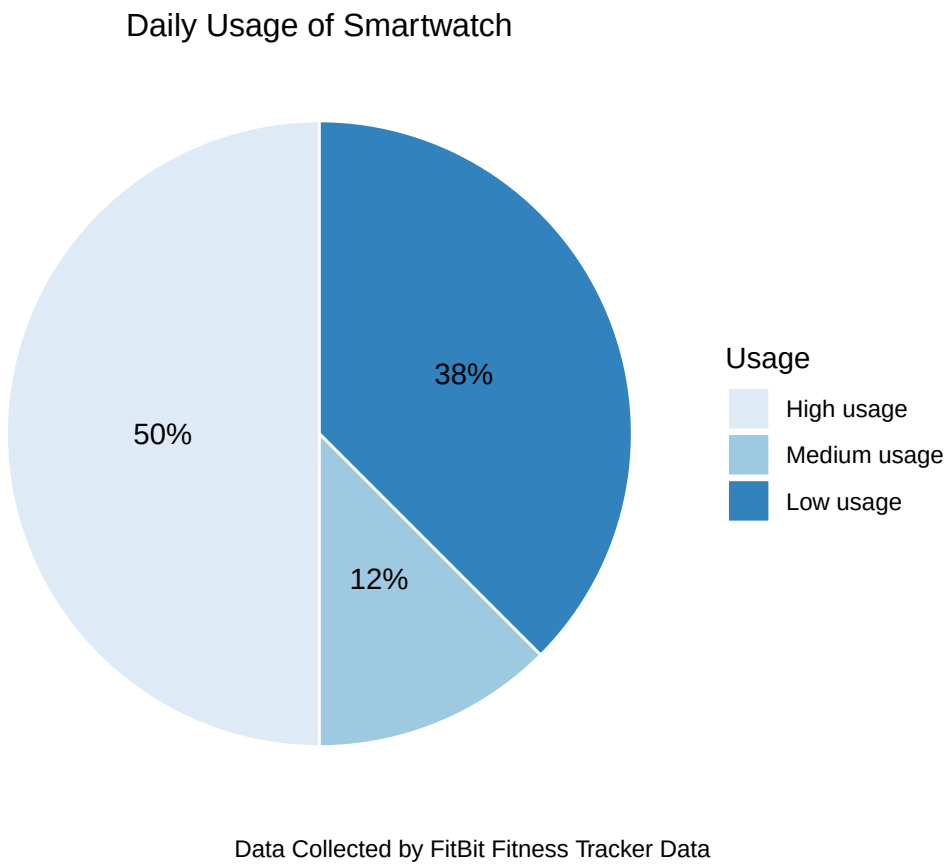
```
# Calculating daily usage level
daily_use <- daily_activity_sleep %>%
  drop_na() %>%
  group_by(id) %>%
  summarise(day_used= n()) %>%
  mutate(usage_level = case_when(
    day_used >=1 & day_used < 10 ~ "Low usage",
    day_used > 10 & day_used <= 20 ~ "Medium usage",
    day_used >20 & day_used <=31 ~"High usage"))

# Calculating daily use percentage
daily_use_perc <- daily_use %>%
  group_by(usage_level) %>%
  summarize(level_totals = n()) %>%
  mutate(total = sum(level_totals)) %>%
```

```
  mutate(perc= level_totals/total) %>%
  mutate(labels = scales::percent(perc))

daily_use_perc$usage_level<- ordered(daily_use_perc$usage_level, levels= c("High usage", "Medium usage"

daily_use_perc
```

```
## # A tibble: 3 x 5
##   usage_level  level_totals total  perc labels
##   <ord>               <int> <int> <dbl> <chr>
## 1 High usage             12    24 0.5   50%
## 2 Low usage               9    24 0.375 38%
## 3 Medium usage            3    24 0.125 12%
```

Plotting the data

```
daily_use_perc %>% ggplot(aes(x="", y= level_totals,fill = usage_level))+
  geom_bar(stat = 'identity', color="white")+
  theme_void()+
  coord_polar("y", start = 0)+
  labs(x="",y="", title = "Daily Usage of Smartwatch",caption = "Data Collected by FitBit Fitness Tracke
  theme(plot.title = element_text(hjust = 0.5, size = 12),
        axis.line = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank())+
        scale_fill_brewer(palette = 1, name = "Usage")+
  geom_text(aes(label = labels), position = position_stack(vjust = 0.5))
```



From the above visual;

- 50% of the users use their device quite frequently - between 21 to 31 days.
- 12% use their device - 11 to 20 days.
- 38% rarely use their device - 1 to 10 days

Step 6: Act

After analyzing the Fitbit fitness tracker data, I came up with some recommendations that will help **Bellabeat to improve its marketing strategy.**

Ideas for Bellabeat app:

1. Average total steps per day are 7638 which a little bit less for having health benefits for according to the CDC research. They found that taking 8,000 steps per day was associated with a 51% lower risk for all-cause mortality (or death from all causes). Taking 12,000 steps per day was associated with a 65% lower risk compared with taking 4,000 steps. Bellabeat should encourage people to take at 10000 per day, through a reminder of remaining step count left to achieve the target.

2. In order to reduce weight and maintain a healthy life, one should reduce calorie intake. Bellabeat can introduce a low calorie diet plan for users and provide ideas for low calorie recipes.

3. Introducing programs and challenges for fitness and health incentives. The reward will come in the form of points, which can then be utilized to get deals on Bellabeat's goods and promotions.

4. Since users are getting an average sleep time less than 8 hours a day, i recommend a notification feature to remind users to go to bed and wake up.

5. Since users have more calories burned, steps taken and distance covered on Monday, Wednesday and Sunday. Bellabeat could use this knowledge to remind users to go for a walk or run on these days. Remaining days users are less active, this can be used to notify users to continue exercise on other days as well.

6. Since 50% of the users use the device frequently. To encourage more usage days, the Bellabeat product can be made to appear more fashionable and elegant to go with a variety of attires.