Bellabeat Case Study

Introduction

Bellabeat is a high-tech company that manufactures health-focused smart products. Its products are intended for women around the world to track their health through the usage of smart devices. As a junior data analyst for the marketing analyst team at Bellabeat, our goal is to give recommendations for their products based on data driven decision making. Urška Sršen and Sando Mur wants us to select one of their four devices and apply these insights into our presentation.

Task

In attempt to boost sales and product effectiveness for Bellabeat, I will be analyzing Fitbit data and identifying trends in their users' data. The goal is to make predictions and suggestions based off of our findings to help Bellabeat better sell their product.

Questions considered

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?

Preparation

To analyze fitness tracker data, we will be using the publicly available FitBit Fitness Tracker Data uploaded on Kaggle. This source contains 18 csv files and its data is from March 2016 to May 2016. However, we will only be using five of these csv files as we will not be needing hourly and minutely data. About 33 FitBit users were consented to the submission of personal tracker data which includes physical activity, heart rate, sleep, daily activity, and steps.

Credibility

Some limitations to this dataset is that the data is from 2016. Since then, technology and users' daily health habits have changed. Another limitation is that this sample size of 33 users is not representative of the worldwide population. However, for the purpose of this case study, we will assume the data is reliable, original, comprehensive, current, and cited.

Processing data and R packages

```r
## Loading libraries
library(readr)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'hms'

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.4     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
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, 6): Library not loaded: /opt/X
##   Referenced from: /Library/Frameworks/R.framework/Versions/4.1/Resources/modules/R_X11.so
##   Reason: image not found

## Could not load tcltk.  Will use slower R code instead.

## Loading required package: RSQLite
```

```
library(lubridate)
```

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

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

```
library(ggplot2)
library(ggpubr)
```

```
## Load datasets
activity = read.csv("dailyActivity_merged.csv")
calories = read.csv("hourlyCalories_merged.csv")
intensity = read.csv("dailyIntensities_merged.csv")
weight = read.csv("weightLogInfo_merged.csv")
sleep = read.csv("sleepDay_merged.csv")
```

Viewing data and distinct values

```
## Activity Data
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```
str(activity)
```

```
## ’data.frame’:    940 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
# Removing unnecessary columns in activity data set
activity = subset(activity, select = -c(LoggedActivitiesDistance,SedentaryActiveDistance))
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   VeryActiveDistance ModeratelyActiveDistance LightActiveDistance
## 1               1.88                     0.55                6.06
## 2               1.57                     0.69                4.71
## 3               2.44                     0.40                3.91
## 4               2.14                     1.26                2.83
## 5               2.71                     0.41                5.04
## 6               3.19                     0.78                2.51
##   VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
## 1                25                  13                  328              728
## 2                21                  19                  217              776
## 3                30                  11                  181             1218
## 4                29                  34                  209              726
## 5                36                  10                  221              773
## 6                38                  20                  164              539
##   Calories
## 1     1985
## 2     1797
## 3     1776
## 4     1745
## 5     1863
## 6     1728
```

```
## Calories Data
head(calories)
```

```
##           Id          ActivityHour Calories
## 1 1503960366 4/12/2016 12:00:00 AM       81
## 2 1503960366  4/12/2016 1:00:00 AM       61
## 3 1503960366  4/12/2016 2:00:00 AM       59
## 4 1503960366  4/12/2016 3:00:00 AM       47
## 5 1503960366  4/12/2016 4:00:00 AM       48
## 6 1503960366  4/12/2016 5:00:00 AM       48
```

```
str(calories)
```

```
## ’data.frame’:    22099 obs. of  3 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr  "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/20
##  $ Calories    : int  81 61 59 47 48 48 48 47 68 141 ...
```

```
## Intensities Data
head(intensity)
```

```
##          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
```

```
str(intensity)
```

```
## ’data.frame’:    940 obs. of  10 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
## Weight Data
head(weight)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

```
str(weight)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/20
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

```
## Sleep Data
head(sleep)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```
str(sleep)
```

```
## 'data.frame':    413 obs. of  5 variables:
```

```
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
## Checking the amount of unique users
length(unique(activity$Id))
```

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

```
length(unique(calories$Id))
```

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

```
length(unique(intensity$Id))
```

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

```
length(unique(weight$Id))
```

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

```
length(unique(sleep$Id))
```

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

This verifies that the number of users in the first three data sets have 33 users. Only 8 of the users have data for weight and 24 have data for their sleep.

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

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

```
sum(duplicated(calories))
```

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

```
sum(duplicated(intensity))
```

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

```
sum(duplicated(weight))
```

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

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

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

The only dataset that contains duplicates is sleep. We can easily fix this with keeping unique rows of the dataset

```
sleep = distinct(sleep)
```

Furthermore, it is important to look for outliers in our data. We can see that there are some 0 values in the column of total steps. These are outliers as the total step count for the rest of the data is all significantly higher than 0.

```
## Outliers
no_steps = activity %>%
  filter(TotalSteps ==0)
length(no_steps$TotalSteps)
```

```
## [1] 77
```

Of the 940 total steps observations in the data set, there 77 values of 0 total steps. This indicates that our fitbit users did not wear the tracker on those days. Considering this would affect the outcome of our data analyses, we will remove these rows of data.

```
## Removing rows of data where Fitbit was not worn
activity[activity$TotalSteps == '0',] = NA
activity = activity[complete.cases(activity),]
str(activity)
```

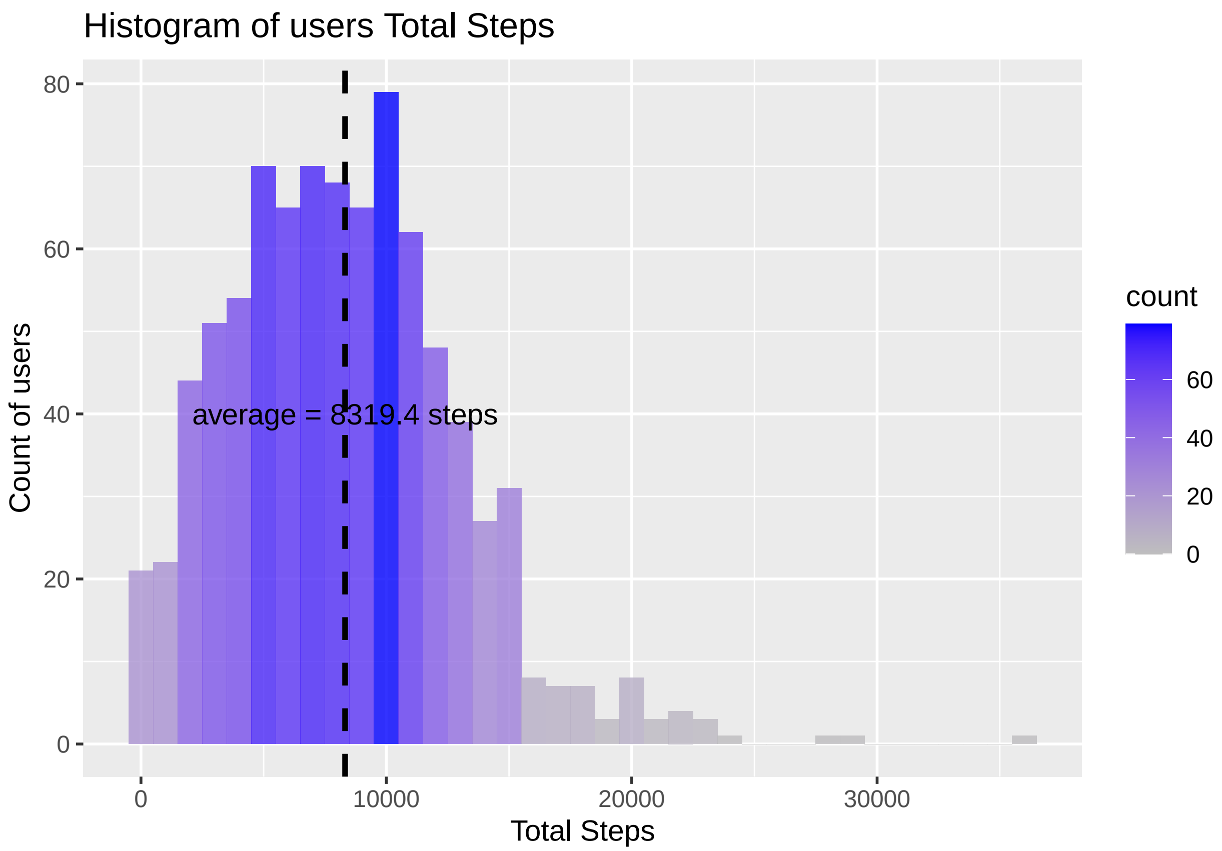
```
## 'data.frame':    863 obs. of  13 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

Analyzing and Visualizing

```
## Comparing average steps and calories of fitbit users to the general population
mean(activity$TotalSteps)
```

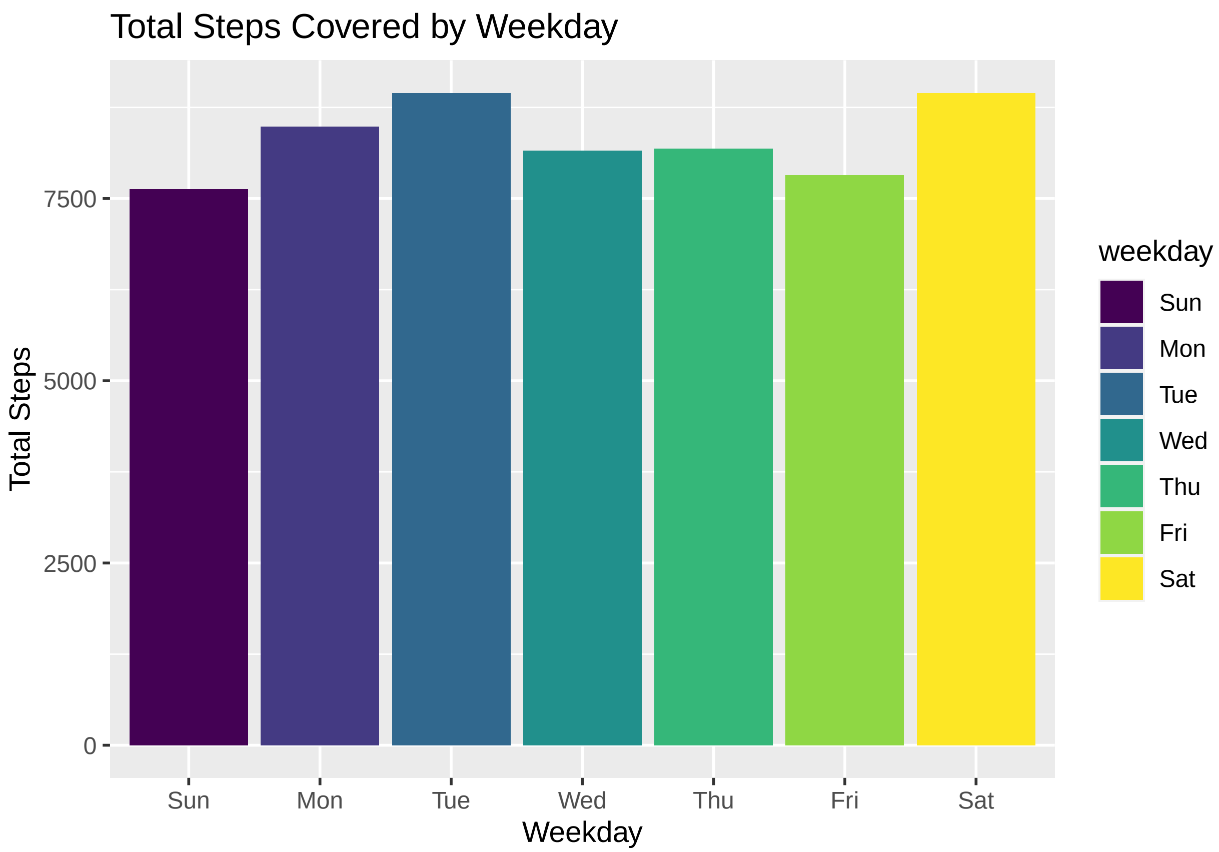
```
## [1] 8319.393
```

```
ggplot(activity, aes(x = TotalSteps, fill = ..count..)) + geom_histogram(alpha = .8, binwidth = 1000) +
```



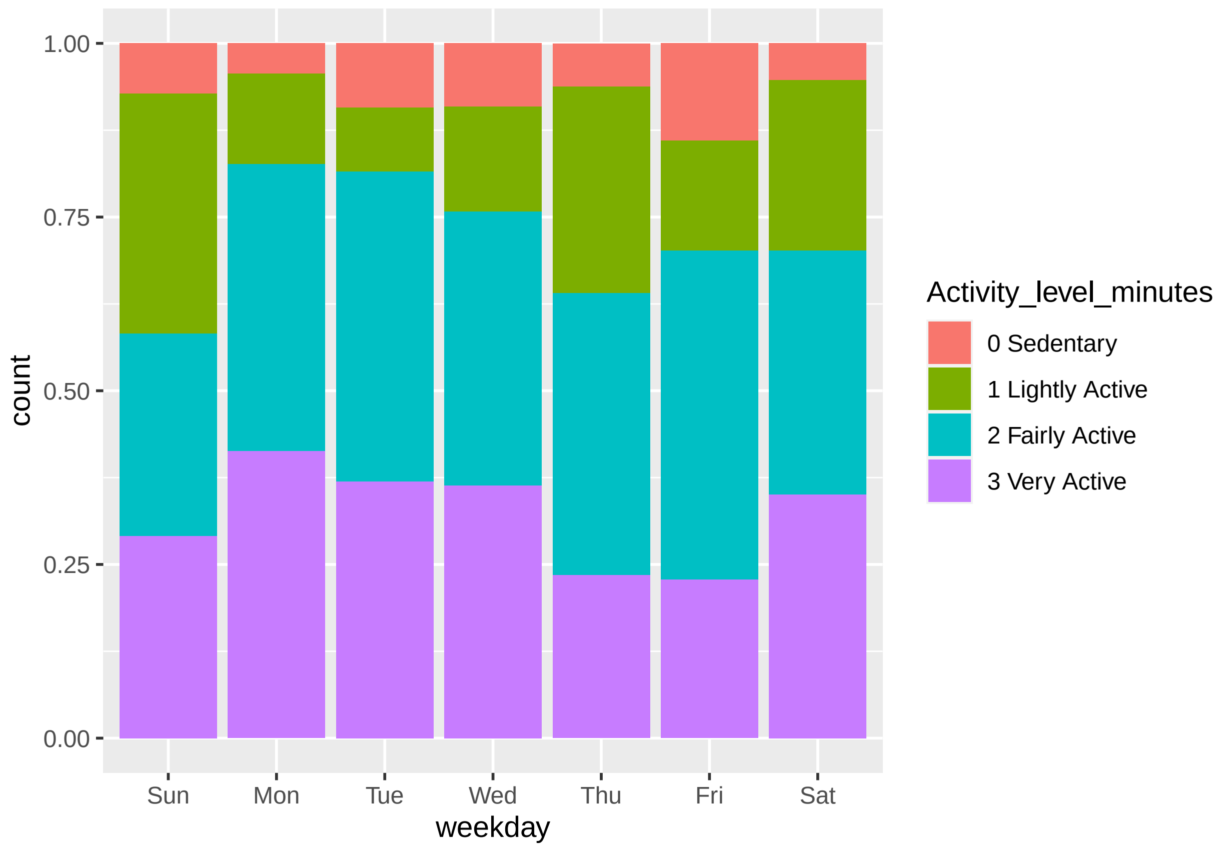
The average step count of the fitbit users in this dataset is 8319. This is much higher than the amount that an average American walks, which is 3,000 to 4,000 steps a day, according to Mayo Clinic.

```
## Comparing activities during the week and the weekend
activity = activity %>%
  distinct() %>%
  mutate(ActivityDate = as.Date(ActivityDate, format = "%m/%d/%Y"))

sleep = sleep %>%
  distinct() %>%
  mutate(SleepDay = as.Date(SleepDay, format = "%m/%d/%Y"))
sleep = rename(sleep, ActivityDate = SleepDay)
activity$weekday = wday(activity$ActivityDate, label = TRUE)
sleep$sleep_week_day = wday(sleep$ActivityDate, label = TRUE)
viz = activity %>%
  group_by(weekday) %>%
  summarize(TotalSteps = mean(TotalSteps)) %>%
  ggplot(aes(x = weekday, y = TotalSteps)) + geom_col(mapping=aes(fill=weekday)) + labs(title = "Total
viz
```



As we can see, our total step count is highest on Tuesdays and it gradually gets lower until Saturday, where it increases and then goes back down on Sunday. We can further break this down by looking at the activity levels performed throughout the week. This will require a new column that indicates how active the user was on that day. To do this, we will be merging the sleep and activity calories as well.

```
## Minute values to limit and categorize sedentarity

minutes1 = 360 #lower limit of waking sedentarity in minutes pere day of what qualifies a sedentary per
minutes2 = 25 #lower limit of moderate / intense physical activity in minutes / day of what qualifies a
steps2 = 5000 #lower limit in steps / day of what qualifies a sedentary person according to different s

## Categorizing users data into levels of activity
merged_activity_sleep = merge(sleep,activity, by=c("Id","ActivityDate")) %>%
  mutate(Activity_level_minutes = case_when ((SedentaryMinutes-TotalMinutesAsleep >= minutes1)&(VeryActi
                                          (SedentaryMinutes-TotalMinutesAsleep >= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep >= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep >= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep <= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep <= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep <= minutes1)&(VeryActive
                                          (SedentaryMinutes-TotalMinutesAsleep <= minutes1)&(VeryActive
## Making columns as a factor
merged_activity_sleep$Activity_level_minutes = as.factor(merged_activity_sleep$Activity_level_minutes)
merged_activity_sleep$weekday = as.factor(merged_activity_sleep$weekday)

## Visualizing the user activity level by weekday
ggplot(data = merged_activity_sleep, aes(x = weekday, fill=Activity_level_minutes)) + geom_bar(stat = "
```



This chart shows us that the Mondays and Tuesdays have the highest levels of being very active and active. As the days go by, this number slowly drops and finally picks up on Friday where it then drops on Sunday. A finding we can make from these two graphs is that these fitbit users tend to be most active during the beginning of the week, then it slowly drops until the weekend where they get more active again. After the Saturday, people tend to be less active on Sundays. Lets also take a look at the distribution of the amount of users whose total steps indicate their activity level. This will tell us the amount of users that are classified into each category for our activity levels in the chart above.

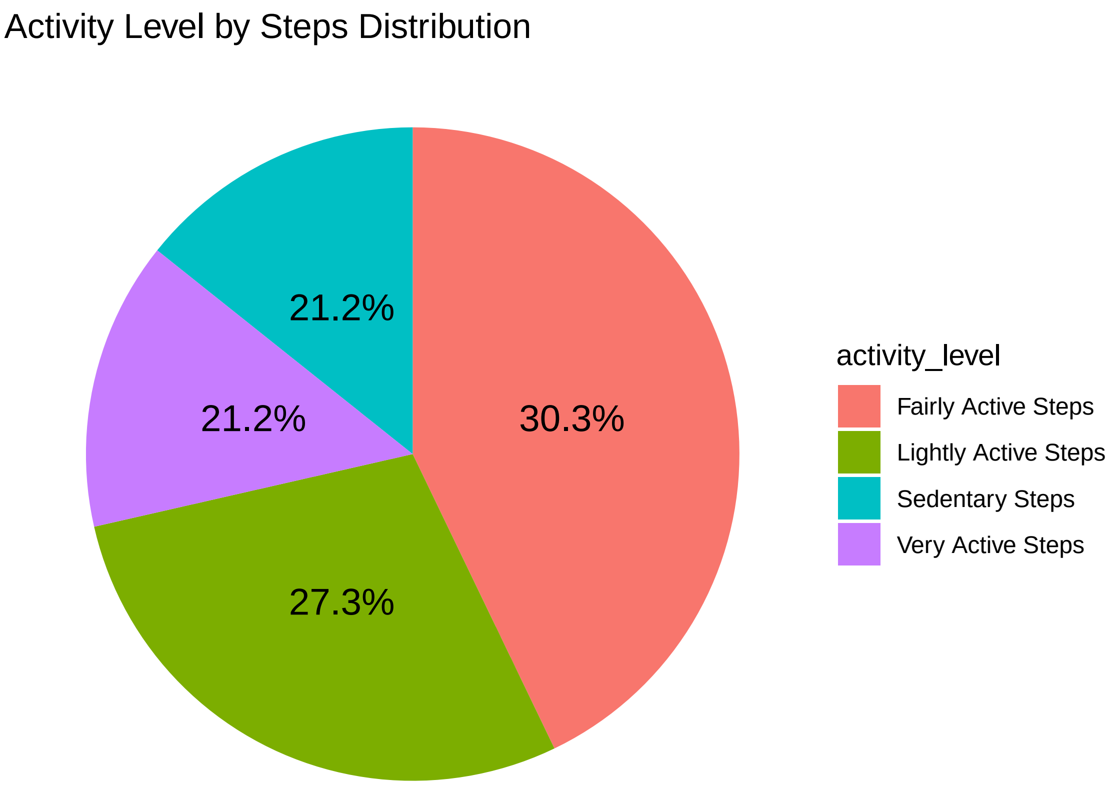
```
## Classifying users' total steps into levels

low = 5000
fair = 7500
high = 10000

activity_level_steps = activity %>%
  group_by(Id)%>%
  summarize(avg_daily_steps = mean(TotalSteps))%>%
  mutate(activity_level = case_when(
    avg_daily_steps <= low ~ "Sedentary Steps",
    avg_daily_steps >= low & avg_daily_steps <= fair ~"Lightly Active Steps",
    avg_daily_steps >= fair & avg_daily_steps <= high ~ "Fairly Active Steps",
    avg_daily_steps > high ~ "Very Active Steps"
  ))
activity_level_steps
```

```
## # A tibble: 33 x 3
##            Id avg_daily_steps activity_level
##         <dbl>           <dbl> <chr>
##  1 1503960366          12521. Very Active Steps
##  2 1624580081           5744. Lightly Active Steps
##  3 1644430081           7283. Lightly Active Steps
##  4 1844505072           3809. Sedentary Steps
##  5 1927972279           1671. Sedentary Steps
##  6 2022484408          11371. Very Active Steps
##  7 2026352035           5567. Lightly Active Steps
##  8 2320127002           4717. Sedentary Steps
##  9 2347167796           9520. Fairly Active Steps
## 10 2873212765           7556. Fairly Active Steps
## # ... with 23 more rows
```

```
## Getting percents for our activity level based on steps
activity_level_steps_percents = activity_level_steps %>%
  group_by(activity_level)%>%
  summarise(total = n()) %>%
  mutate(percent = scales::percent(total/sum(total)))

activity_level_steps_percents
```

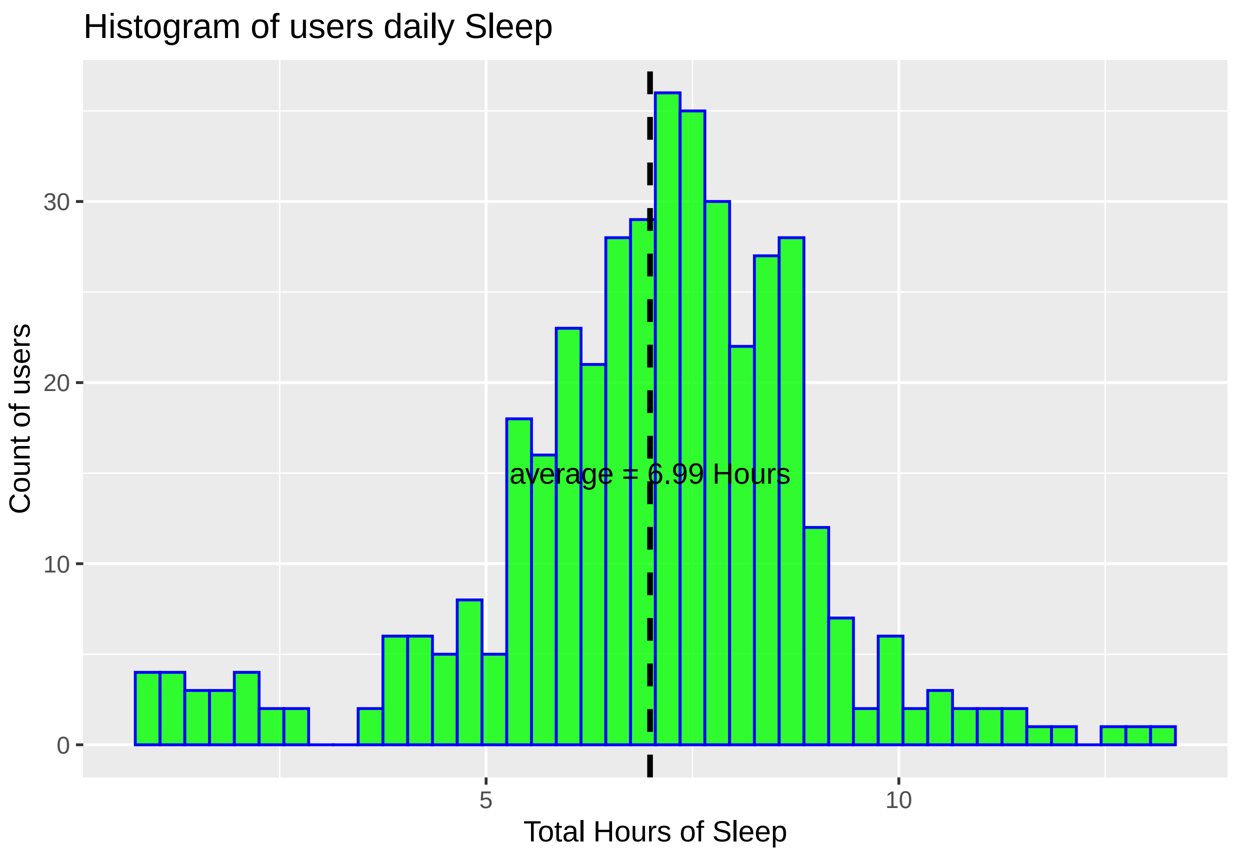
```
## # A tibble: 4 x 3
##   activity_level       total percent
##   <chr>                <int> <chr>
## 1 Fairly Active Steps     10 30.3%
## 2 Lightly Active Steps     9 27.3%
## 3 Sedentary Steps          7 21.2%
## 4 Very Active Steps        7 21.2%
```

```
## Plotting our pie chart
ggplot(activity_level_steps_percents, aes(x="", y = percent, fill = activity_level)) + geom_bar(stat="i
```



From our findings, we can see that the amount of users' activity levels based on their mean total steps is fairly distributed across the board. We have 9 users that were classified in lightly and fair active steps, while there were 8 users in sedentary and 7 users in very active steps. This distribution is fairly even and further justifies our chart above proving its aggregation technique.

```
## Visualizing the amount of hours slept daily
ggplot(sleep, aes(x = TotalMinutesAsleep/60)) + geom_histogram(alpha = .8, binwidth = .3, color = "blue
```

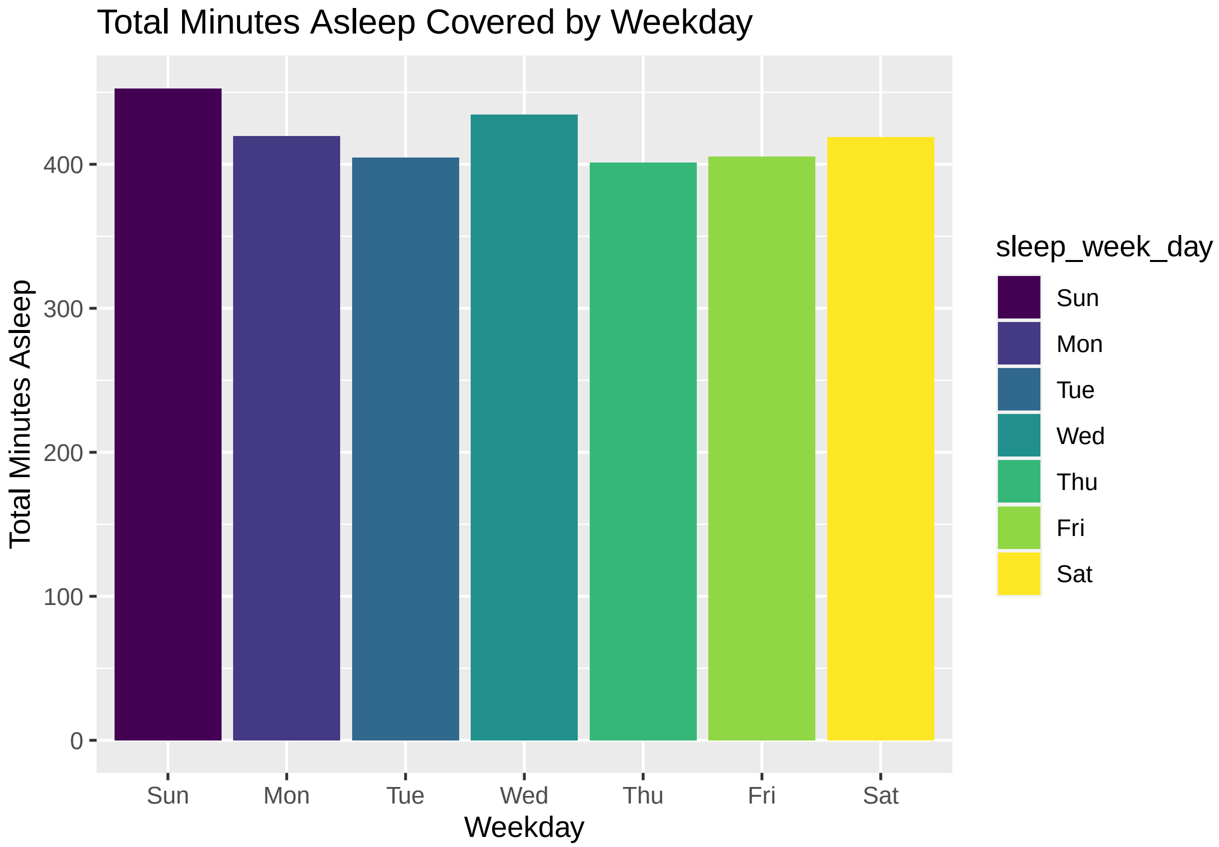


```
summary(sleep$TotalMinutesAsleep/60)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9667  6.0167  7.2083  6.9862  8.1667 13.2667
```

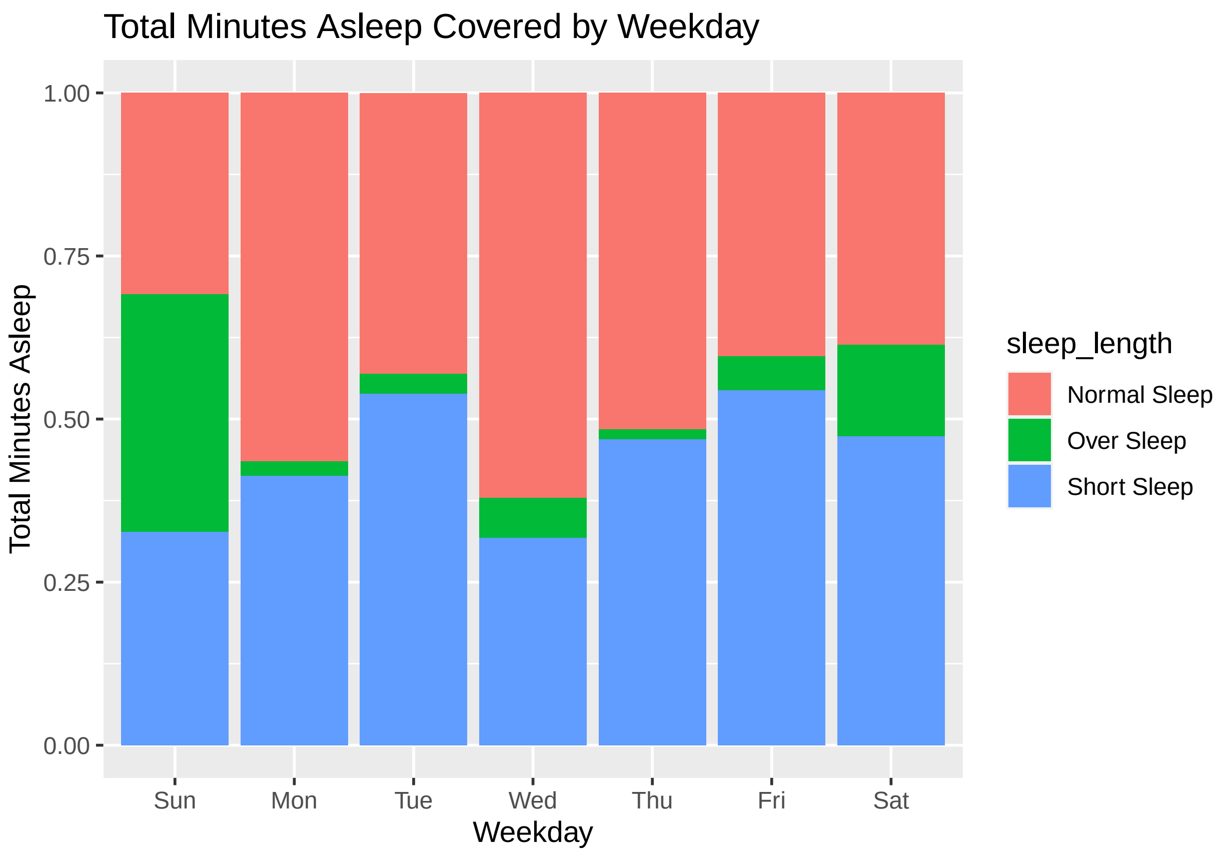
The data distribution just about follows a bell curve with outliers on both sides. Most users slept more than 7 hours. Lets see how our users' sleep habits change throughout the weekdays.

```
## Visualizing the amount of sleep covered by the day of the week

viz_2 = sleep %>%
  group_by(sleep_week_day) %>%
  summarize(TotalMinutesAsleep = mean(TotalMinutesAsleep)) %>%
  ggplot(aes(x = sleep_week_day, y = TotalMinutesAsleep)) + geom_col(mapping = aes(fill=sleep_week_day)
viz_2
```



Looking at our chart, we see that the amount of sleep is highest on Sunday and decreases until Wednesday, where it slowly increases again until Saturday. Lets look into this further by seeing which days people tend to oversleep.

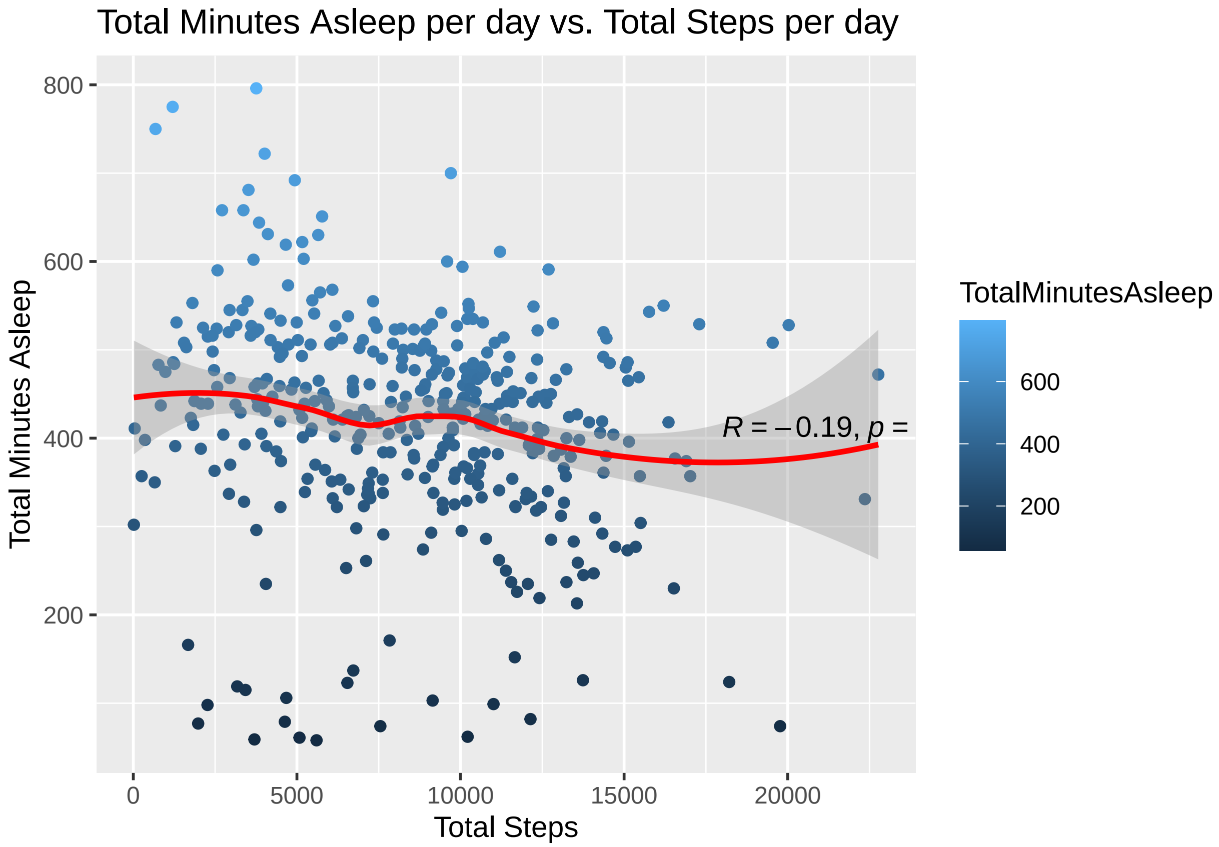
```
## Categorizing sleep durations
sleep1 = 420 ##420 minutes is equivalent to 7 hours. Anything sleep less than 7 hours will be classifie
sleep2 = 540 ##540 minutes is equivalent to 9 hours. Sleep duration that is between 7 and 9 hours will 

sleep_with_length = sleep %>%
  mutate(sleep_length = case_when(
    TotalMinutesAsleep < sleep1 ~ "Short Sleep",
    TotalMinutesAsleep >= sleep1 & TotalMinutesAsleep <= sleep2 ~ "Normal Sleep",
    TotalMinutesAsleep > sleep2 ~ "Over Sleep"
  ))

ggplot(sleep_with_length, aes(x = sleep_week_day, fill = sleep_length)) + geom_bar(stat="count",positio
```



As we can see, short sleeping is variant throughout the days. However, we can see that on Sundays, many of the users over slept. We can use this information to give accommodated sleep reminders from our devices. Next, I will see how sleep can be affected by the amount of total steps.

```
## Comparing sleep and total steps
merged_activity_sleep %>%
  ggplot(aes(x = TotalSteps, y = TotalMinutesAsleep, color = TotalMinutesAsleep)) + geom_point() + geom_
```

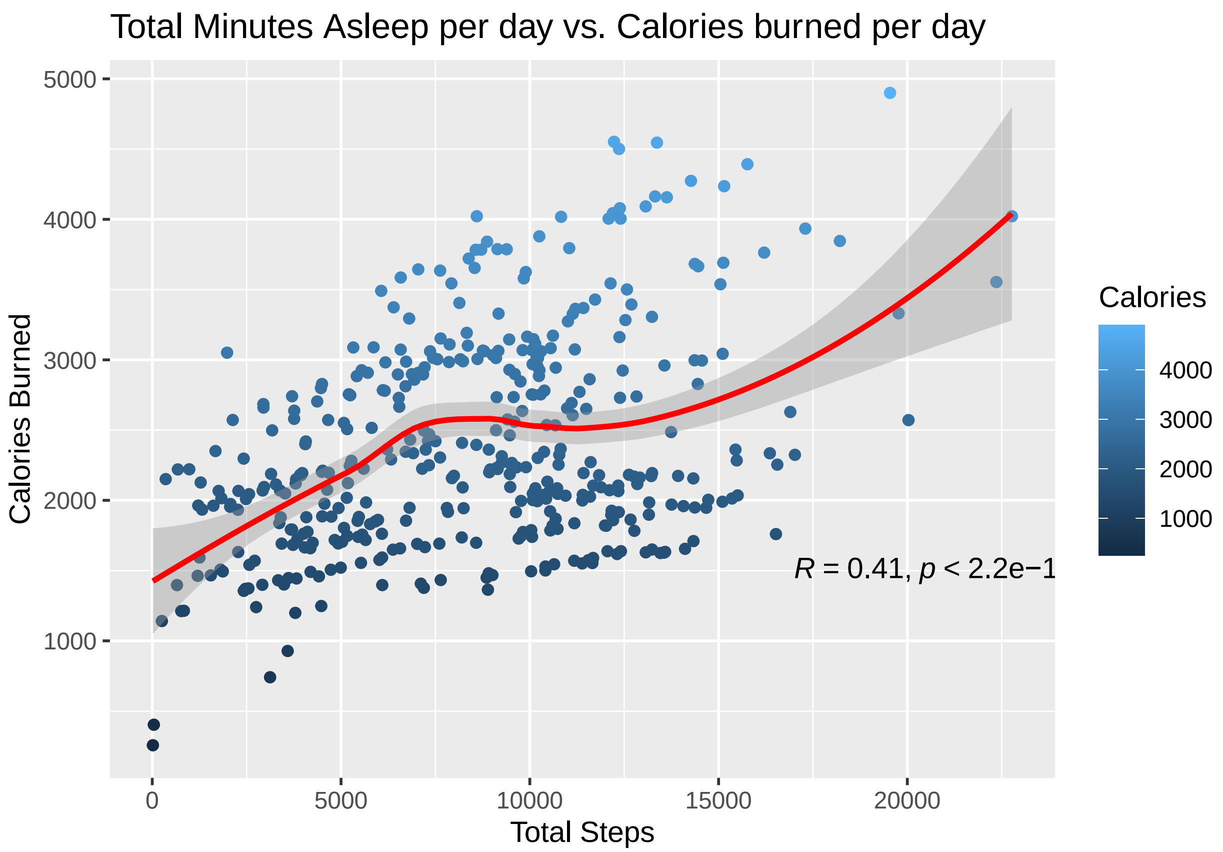
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



As we can see, the data is very spread out here and we can't come to a conclusion with their relationship. Looking at the correlation coefficient, we have a value of -0.19 which indicates a very weak correlation between our two variables. We can't really conclude with anything here, but we can continue to compare total steps and calories burned.

```
## Comparing total steps and calories burned
merged_activity_sleep %>%
  ggplot(aes(x = TotalSteps, y = Calories, color = Calories)) + geom_point() + geom_smooth(color = "red
```

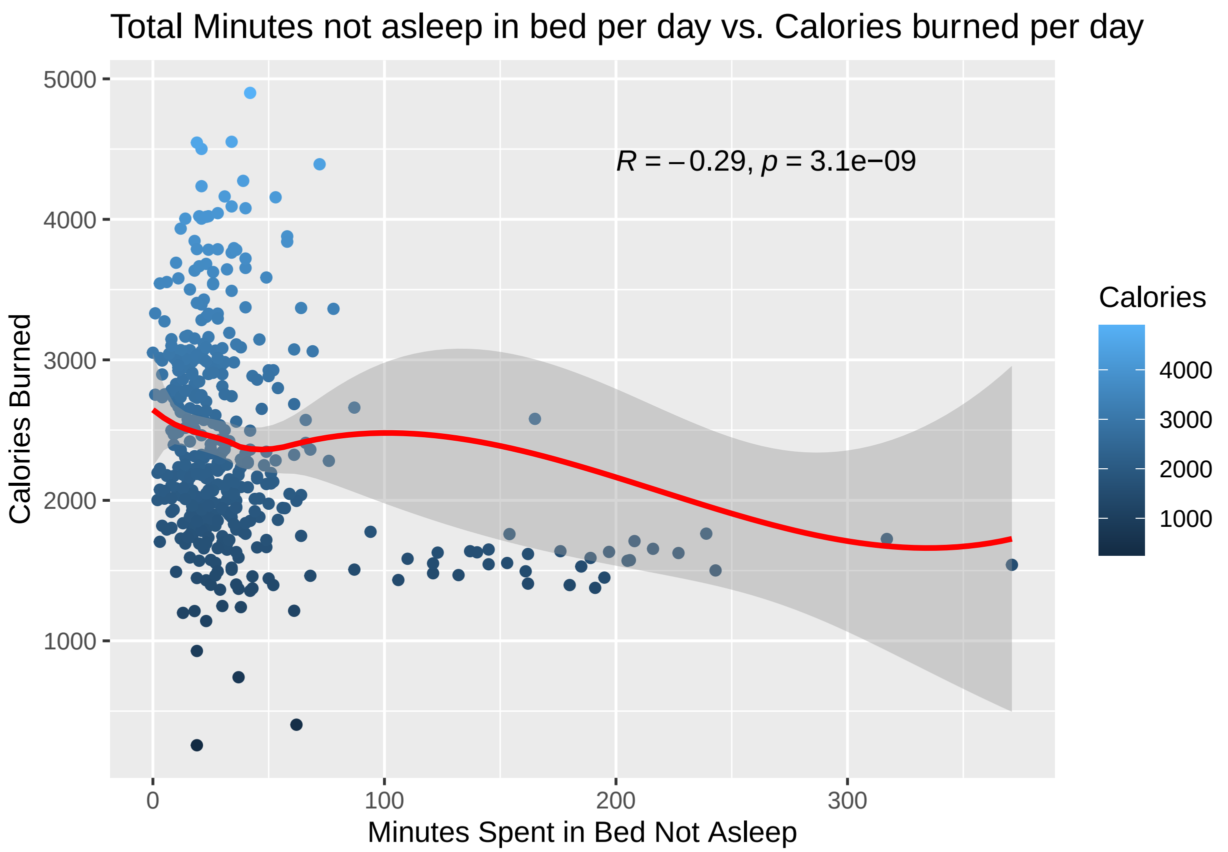
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



As we can see, the data follows a positive trend and our correlation coefficient is a value of 0.41. This indicates that there is a moderate association with total amount of steps and calories burned. We can conclude that the more steps we take, the higher the amount of calories burned. It will also be beneficial to see how the time spent in bed but not asleep is correlated with calories burned per day.

```
# Comparing time spent in bed not sleeping and calories burned per day
## Creating new column calculating time spent in bed not sleeping
merged_activity_sleep %>%
  mutate(Total_minutes_in_bed_not_asleep = TotalTimeInBed - TotalMinutesAsleep) %>%
  ggplot(aes(x = Total_minutes_in_bed_not_asleep, y = Calories, color = Calories)) + geom_point() + geor
```

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



Looking at our graph, we can see that there is a negative correlation with calories burned and minutes spent in bed not asleep. As the amount of time spent in bed no asleep increases, the overall trend of calories burned tends to decrease. The R squared value is -0.29, which indicates a negative weak correlation.

Sharing Conclusions

1. Fitbit users have an average of 8319 total steps per day. This is almost double the amount that the average American walks, which is 3,000 to 4,000 steps a day, according to Mayo Clinic. Bellabeat can use this information to simply inform potential and existing customers to wear their device, as it will increase their daily step count.
2. The average total step count of fitbit users is highest on Tuesdays and Saturdays. Throughout other days of the week, they tend to vary. This information can be used as a Bellabeat reminder to influence users to increase their step counts on certain days.
3. Monday and Tuesday are days where users had the highest level of being fairly and very active. Throughout other days of the week, they tend to vary as well. This information can be used for Bellabeat reminders of when to influence users to be more active.
4. The average total minutes of sleep per night is highest on Sundays and Wednesdays. Bellabeat can use this information and give out users reminders on low days like Tuesday and Thursday to sleep early.
5. It was found that sleeping short of the recommended sleeping hours was highest on Tuesday and Friday. Bellabeat could use this information to remind users the day of to possibly avoid short sleeping. It was also found that Sundays and Saturdays are the most common days of over sleeping. Bellabeat could use this information to remind users before the weekend starts to avoid over sleeping.

6. The higher the amount of total steps, the higher the amount of calories burned. Bellabeat can use this information to set out motivational reminders that the more steps the more calories.
7. The longer the time spent in bed not sleeping, the less amount of calories burned throughout the day. Bellabeat can use this information to remind users to spend less time in bed when not sleeping, as this will affect their calories burned throughout the day.

Act/Recommendations

Bellabeat's existing and potential customers are users who like to wear a device that promotes overall health and wellness. Their smart devices are relied on to give users accurate and helpful data regarding their daily steps, quality of activities, heart rate, sleep, and much more. In the purpose of this case study, we have revealed some helpful tactics that Bellabeat can use to gain effectiveness in their devices. From the data that we have collected, it would be helpful to focus our findings on the Bellabeat Leaf, a tracker that can be worn as a bracelet, necklace, or clip. This device, compared to the Bellbeat Time watch, ensures that users can wear the device comfortably throughout the day and night. The following recommendations will focus on the marketing campaign of the Bellabeat Leaf.

1. Informing existing and potential Bellabeat customers that people who wear smart trackers are healthier and tend to take more steps in a day. The average user in our study was found to have almost double the amount of an American's average total steps per day. Trackers are effective in systematically increasing overall activity.
2. Encourage users to stay active throughout the week. Daily steps tend to often vary, consider setting weekly goals for our users and possibly further advocating for them on days where they are likely to drop. We can also apply this technique to the type of activities performed. It would be beneficial to consider setting motivational reminders to perform the same amount or possibly even more of certain types of intensity compared to previous weeks or days.
3. Inspire users to have healthy sleeping habits throughout the week. Durations of sleep tend to vary throughout the week, consider setting weekday and weekend goals for our users. In addition, Bellabeat could also show how being in bed while not sleeping could negatively affect other health aspects such as calories burned.
4. Promote the Bellabeat Leaf as a versatile, stylish device that can be worn throughout the day. Considering the Bellabeat leaf can be worn on various parts of the body, make this a special feature in the marketing. Its ability to comfortably track during our sleep is very powerful. Sleeping habits are often overlooked in smart tracking devices and the Leaf has potential to fill in the gaps where competitors will lack.
5. Consider a type of tracking for stress and menstrual cycles. From our data, it is clear that activities and sleep tend to vary. For Bellabeat, it would be highly beneficial to consider tracking stress and menstrual cycles as part of the factors that play into the variance. This could really make Bellabeat a leader in that area for women.

Next Steps

Our case study was designed to identify trends and make suggestions for Bellabeat customers. Although we analyzed Fitbit users in our data set, Fitbit and Bellabeat are both similar companies striving to provide the best health and wellness features in their smart device trackers. Like many, our case study was not perfect. The data we obtained was not the most credible and reliable. Some limitations to this case study are as follows.

1. Our dataset only had 33 users. 33 users is not a very good representation of the general population. We would've liked to see thousands and possibly millions of different users to better serve the general population. In addition, the demographics of our users were unknown. Considering Bellabeat's consumer base is primarily focused on women, we would want to consider a larger dataset with a focus on female users for our next project.

2. Fitbit users were not consistent in logging their activities. We do not know if our users wore their device for the entire day or if they took them off for a certain period. This is very much displayed in our sleeping data and does compromise the completeness of our observations.

3. Our dataset is from 2016 and only spans about two months. Since 2016, technology along with health interventions have changed. It would be helpful to have a data set that is within the past couple years for this purpose.