Bellabeat Fitness Tracker Case Study

Chris Fair

2022-03-01

Introduction and background

Bellabeat is a high-tech manufacturer of health-focused devices and technology for women. As a company they are interested in investigating how their target-consumers use currently use other smart-devices. This case study is meant to fulfill that need by looking at current smart device usage data to gain insights that could be useful for Bellabeat's product development and marketing.

For this analysis, FitBit Fitness Tracker Data was utilized. The data contains the personal tracker data of 30 FitBit users and includes daily activity (calories, intensities, and steps), heart rate, calories burned, sleep, and weight. The data comes from a publicly licensed dataset available via Kaggle and a detailed description of the dataset, as well as the dataset itself, can be found here: https://www.kaggle.com/arashnic/fitbit.

Loading necessary packages and libraries

```
library(tidyverse)  #helps wrangle data
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```
library(lubridate)  #helps wrangle date attributes
```

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

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

```
library(ggplot2)  #helps visualize data
library(readxl) #importing excel files
library(tinytex) #knitting to PDF
#library(reshape2)
#library(janitor) #helps with cleaning data
```

Importing CSV files

Note: The data contained in the daily_calories, daily_intensities, and daily_steps files are included in the daily_activity file, so those files will not be imported or analyzed.

Data cleaning

Exploring the columns in each dataframe.

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

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

```
## [1] "Id"               "ActivityHour"     "TotalIntensity"   "AverageIntensity"
```

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

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

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

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

Note that all of the datasets have 'Id' as a column, so this will be used when merging them.

Taking a look at the daily_activity data and formatting the date.

```
daily_activity$date <- as.Date(daily_activity$ActivityDate, "%m/%d/%Y")
daily_activity$weekDay <- weekdays(daily_activity$date)
head(daily_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       date
## 1                  13                  328              728     1985 2016-04-12
## 2                  19                  217              776     1797 2016-04-13
## 3                  11                  181             1218     1776 2016-04-14
## 4                  34                  209              726     1745 2016-04-15
## 5                  10                  221              773     1863 2016-04-16
## 6                  20                  164              539     1728 2016-04-17
##     weekDay
## 1   Tuesday
## 2 Wednesday
## 3  Thursday
## 4    Friday
## 5  Saturday
## 6    Sunday
```

Taking a look at the hourly data.

```
head(hourly_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
```

```
head(hourly_intensities)
```

```
##           Id          ActivityHour TotalIntensity AverageIntensity
## 1 1503960366 4/12/2016 12:00:00 AM             20         0.333333
## 2 1503960366  4/12/2016 1:00:00 AM              8         0.133333
## 3 1503960366  4/12/2016 2:00:00 AM              7         0.116667
## 4 1503960366  4/12/2016 3:00:00 AM              0         0.000000
## 5 1503960366  4/12/2016 4:00:00 AM              0         0.000000
## 6 1503960366  4/12/2016 5:00:00 AM              0         0.000000
```

```r
head(hourly_steps)
```

```
##           Id          ActivityHour StepTotal
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366  4/12/2016 1:00:00 AM       160
## 3 1503960366  4/12/2016 2:00:00 AM       151
## 4 1503960366  4/12/2016 3:00:00 AM         0
## 5 1503960366  4/12/2016 4:00:00 AM         0
## 6 1503960366  4/12/2016 5:00:00 AM         0
```

Taking a look at the sleep data and formatting the time, as well as adding columns for the hour and weekday for each activity. This will be useful for trending later.

```r
minute_sleep$hour <- format(as.POSIXct(minute_sleep$date,format = "%m/%d/%Y %I:%M:%S %p"), "%H")
sleep_day$date <- as.Date(sleep_day$SleepDay, "%m/%d/%Y %I:%M:%S %p")
sleep_day$weekDay <- weekdays(sleep_day$date)
head(sleep_day)
```

```
##           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       date   weekDay
## 1            346 2016-04-12   Tuesday
## 2            407 2016-04-13 Wednesday
## 3            442 2016-04-15    Friday
## 4            367 2016-04-16  Saturday
## 5            712 2016-04-17    Sunday
## 6            320 2016-04-19   Tuesday
```

```r
head(minute_sleep)
```

```
##           Id                 date value       logId hour
## 1 1503960366 4/12/2016 2:47:30 AM     3 11380564589   02
## 2 1503960366 4/12/2016 2:48:30 AM     2 11380564589   02
## 3 1503960366 4/12/2016 2:49:30 AM     1 11380564589   02
## 4 1503960366 4/12/2016 2:50:30 AM     1 11380564589   02
## 5 1503960366 4/12/2016 2:51:30 AM     1 11380564589   02
## 6 1503960366 4/12/2016 2:52:30 AM     1 11380564589   02
```

Data cleaning

There seem to be differing numbers of participants in the data. It looks like not all of the participants provided sleep or weight data, so this needs to be kept in mind when merging the data. Additionally, this takes the already small sample size and makes it even smaller, so any insights gained from either the sleep or weight analysis will required further investigation to confirm.

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

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

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

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

```r
#n_distinct(daily_calories$Id)
#n_distinct(daily_intensities$Id)
#n_distinct(daily_steps$Id)
n_distinct(hourly_steps$Id)
```

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

```r
n_distinct(hourly_intensities$Id)
```

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

```r
n_distinct(hourly_calories$Id)
```

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

```r
n_distinct(minute_sleep$Id)
```

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

```r
n_distinct(weight_log$Id)
```

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

Checking to see how many observations are in each dataframe.

```r
nrow(daily_activity)
```

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

```r
nrow(sleep_day)
```

```
## [1] 413
```

```r
#nrow(daily_calories)
#nrow(daily_intensities)
#nrow(daily_steps)
nrow(hourly_steps)
```

```
## [1] 22099
```

```
nrow(hourly_intensities)
```

```
## [1] 22099
```

```
nrow(hourly_calories)
```

```
## [1] 22099
```

```
nrow(minute_sleep)
```

```
## [1] 188521
```

```
nrow(weight_log)
```

```
## [1] 67
```

Grouping the hourly data together and formatting the date-time to be used later.

```
merge1 <- merge(hourly_steps, hourly_calories, all = TRUE)
hourly_data <- merge(hourly_intensities, merge1, all = TRUE)
hourly_data$actHour <- as.POSIXct(hourly_data$ActivityHour,format = "%m/%d/%Y %I:%M:%S %p")
hourly_data$date <- as.Date(hourly_data$ActivityHour, "%m/%d/%Y")
hourly_data$hour <- format(hourly_data$actHour, "%H:%M")
n_distinct(hourly_data$Id)
```

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

```
sum(is.na(hourly_data))     #checking for NA values
```

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

Before proceeding further, all days where the step count is equal to zero will be removed from the daily_activity dataframe as this is indicative of the user not wearing their device or an issue with the sensors in the device, either of which would throw off the data. Additionally, those dates will be removed from the hourly dataframes for those userIDs.

```
nrow(daily_activity)
```

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

```
nrow(hourly_data)
```

```
## [1] 22099
```

```
empty_days <- daily_activity %>% filter(TotalSteps==0)
empty_days <- select(empty_days, Id, date)
hourly_data <- hourly_data %>% filter(Id != empty_days$Id & date != empty_days$date)
daily_activity <- daily_activity %>% filter(TotalSteps > 0)
nrow(daily_activity)
```

```
## [1] 863
```

```r
nrow(hourly_data)
```

```
## [1] 20733
```

```r
nrow(empty_days)
```

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

Summary statistics for each dataframe

From the daily activity dataframe, the daily summary statistics for total steps, total distance moved, sedentary minutes, and calories burned can be seen below.

```r
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         Calories) %>%
  summary()
```

```
##    TotalSteps    TotalDistance   SedentaryMinutes    Calories
##  Min.   :    4   Min.   : 0.00   Min.   :   0.0   Min.   :  52
##  1st Qu.: 4923   1st Qu.: 3.37   1st Qu.: 721.5   1st Qu.:1856
##  Median : 8053   Median : 5.59   Median :1021.0   Median :2220
##  Mean   : 8319   Mean   : 5.98   Mean   : 955.8   Mean   :2361
##  3rd Qu.:11092   3rd Qu.: 7.90   3rd Qu.:1189.0   3rd Qu.:2832
##  Max.   :36019   Max.   :28.03   Max.   :1440.0   Max.   :4900
```

From the hourly dataframe, the hourly summary statistics for average intensity, total steps, and calories burned can be seen below.

```r
hourly_data %>%
  select(TotalIntensity,
         AverageIntensity,
         StepTotal,
         Calories) %>%
  summary()
```

```
##  TotalIntensity   AverageIntensity   StepTotal          Calories
##  Min.   :  0.00   Min.   :0.0000   Min.   :    0.0   Min.   : 42.0
##  1st Qu.:  0.00   1st Qu.:0.0000   1st Qu.:    0.0   1st Qu.: 63.0
##  Median :  3.00   Median :0.0500   Median :   45.0   Median : 83.0
##  Mean   : 12.19   Mean   :0.2032   Mean   :  324.5   Mean   : 97.5
##  3rd Qu.: 17.00   3rd Qu.:0.2833   3rd Qu.:  364.0   3rd Qu.:109.0
##  Max.   :180.00   Max.   :3.0000   Max.   :10554.0   Max.   :948.0
```

From the sleep dataframe, the summary statistics for daily total sleep records, daily total time asleep, and daily total time in bed can be seen below.

```
sleep_day %>%
  select(TotalSleepRecords,
         TotalMinutesAsleep,
         TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

From the weight dataframe, the summary statistics for weight (in pounds) and BMI can be seen below.

```
weight_log %>%
  select(WeightPounds,
         BMI) %>%
  summary()
```

```
##   WeightPounds        BMI
##  Min.   :116.0   Min.   :21.45
##  1st Qu.:135.4   1st Qu.:23.96
##  Median :137.8   Median :24.39
##  Mean   :158.8   Mean   :25.19
##  3rd Qu.:187.5   3rd Qu.:25.56
##  Max.   :294.3   Max.   :47.54
```

Checking to see if there were any significant weight changes throughout the dataset.

```
weight_log%>%
  group_by(Id)%>%
  summarise(min(WeightPounds),max(WeightPounds),max(WeightPounds)-min(WeightPounds))
```

```
## # A tibble: 8 x 4
##           Id `min(WeightPounds)` `max(WeightPounds)` `max(WeightPounds) - min(~`
##        <dbl>               <dbl>               <dbl>                       <dbl>
## 1 1503960366                116.                116.                       0
## 2 1927972279                294.                294.                       0
## 3 2873212765                125.                126.                       1.32
## 4 4319703577                159.                160.                       0.220
## 5 4558609924                152.                155.                       2.65
## 6 5577150313                200.                200.                       0
## 7 6962181067                134.                138.                       3.31
## 8 8877689391                185.                189.                       3.97
```

Looking at a few correlations

From a quick check of the correlations below we can see a strong correlation between calories burned, step total, and total intensity in the hourly dataframe. There is a modest correlation between very active minutes and total distance moved from the daily activity dataframe and a weak correlation between sedentary minutes and total daily steps.

Additionally, there is a very strong correlation between total steps and total distance moved, which is expected given most users daily activity usually comes from walking and or jogging, instead of cycling.

```
cor(daily_activity$TotalSteps, daily_activity$SedentaryMinutes)
```

```
## [1] -0.1890291
```

```
cor(daily_activity$VeryActiveMinutes, daily_activity$TotalDistance)
```

```
## [1] 0.6756469
```

```
cor(daily_activity$LightlyActiveMinutes, daily_activity$TotalDistance)
```

```
## [1] 0.3831733
```

```
cor(daily_activity$Calories, daily_activity$SedentaryMinutes)
```

```
## [1] -0.03106289
```

```
cor(hourly_data$Calories, hourly_data$TotalIntensity)
```

```
## [1] 0.8969005
```

```
cor(hourly_data$TotalIntensity, hourly_data$StepTotal)
```

```
## [1] 0.8960473
```

```
cor(daily_activity$TotalDistance,daily_activity$TotalSteps)
```

```
## [1] 0.9826911
```

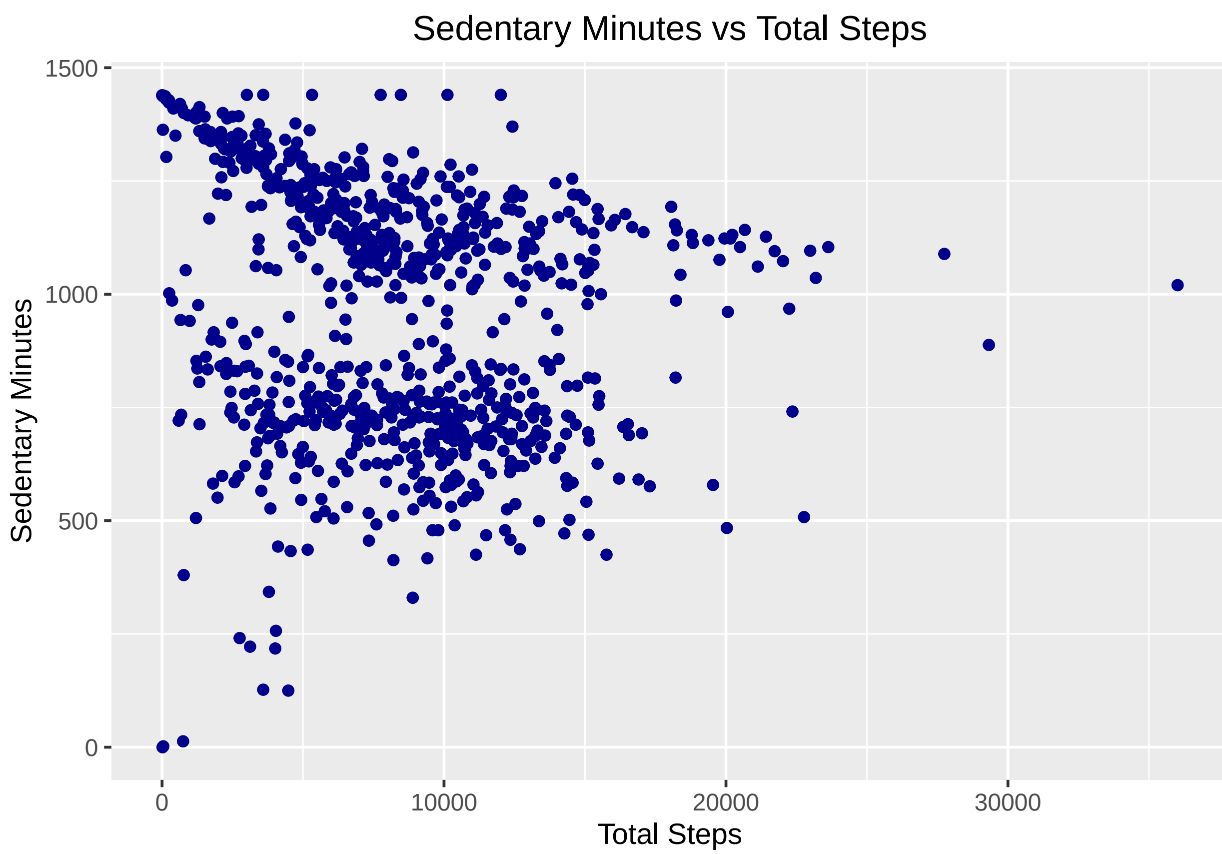
Plotting the data for visual explorations

Daily Activity Trends

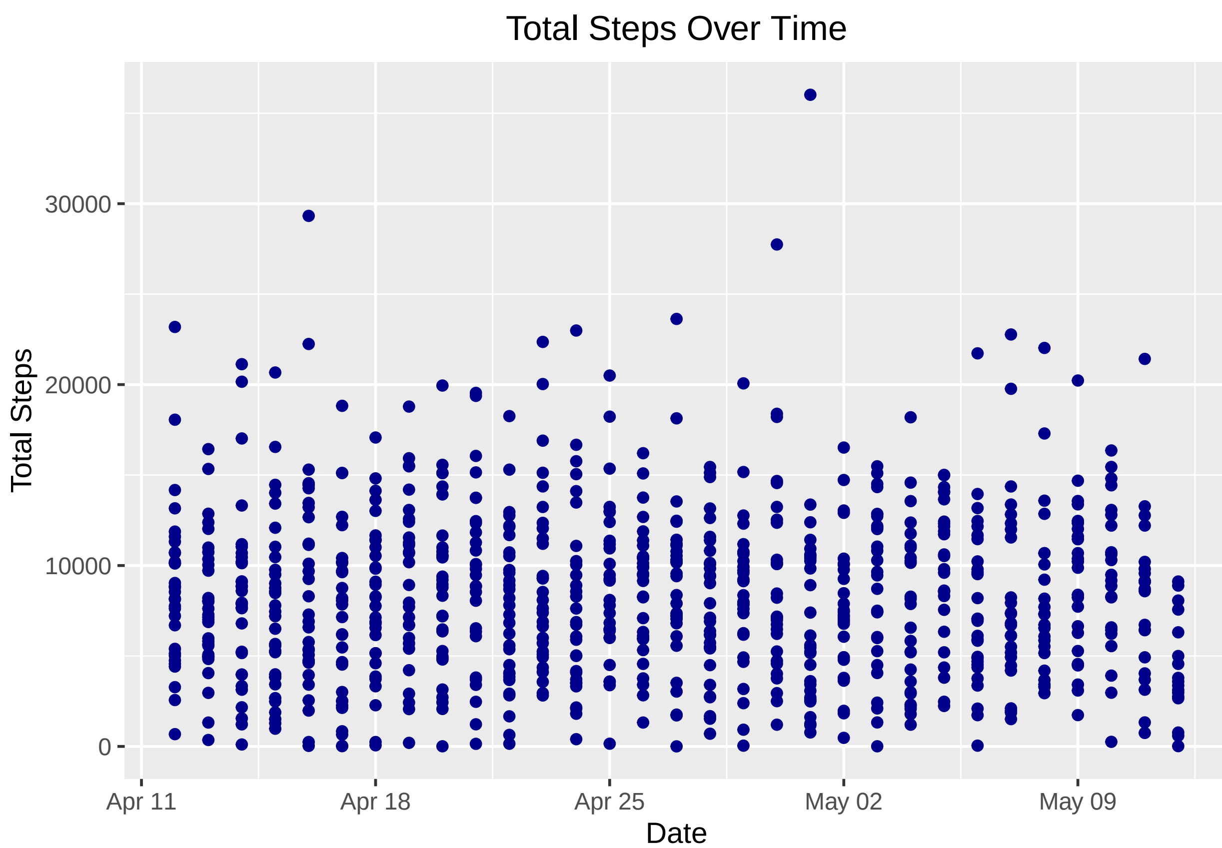
As can be seen below, there is a weak negative correlation between daily steps and sedentary minutes. This could potentially be useful in promoting a healthy lifestyle by encouraging users to increase their daily step count.

There did not appear to be any change in user physical activity as measured by step count over the course of this data set.

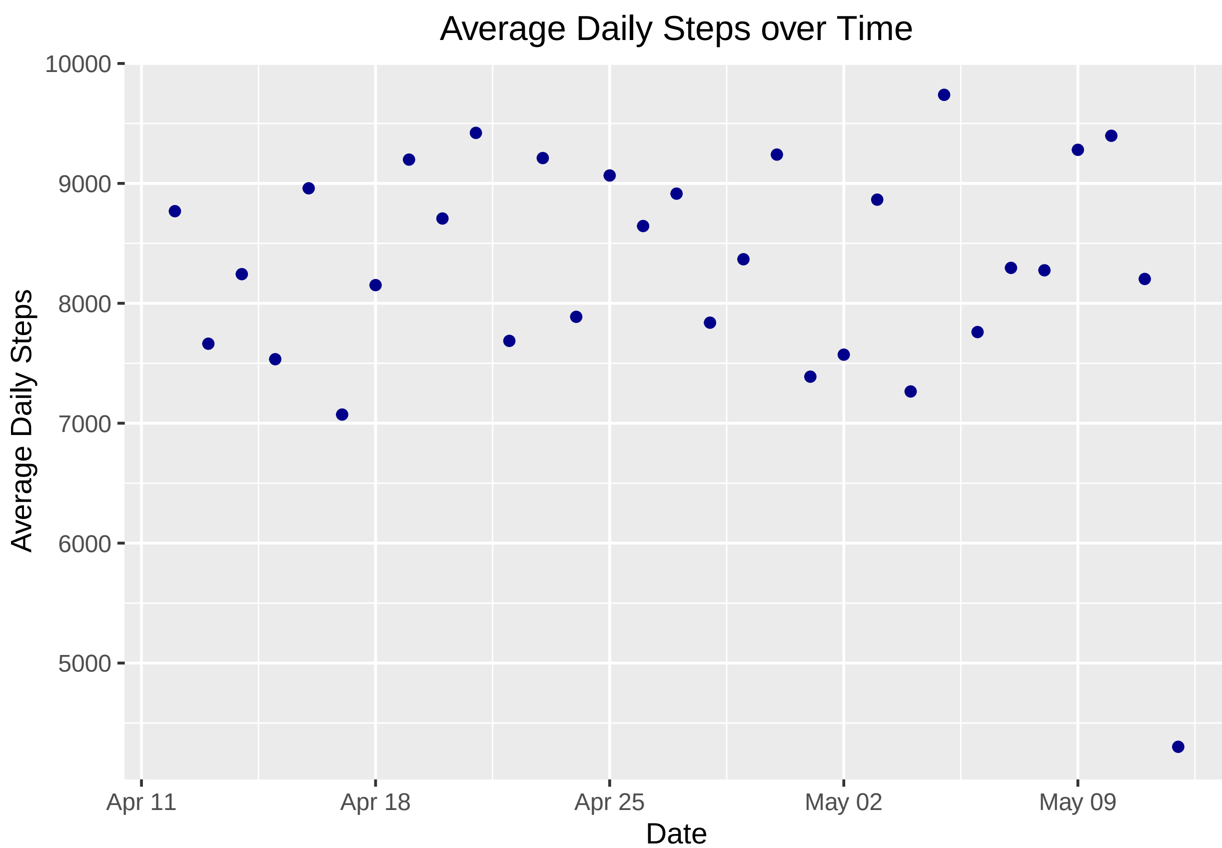
```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes)) +
  geom_point(color = 'blue4') +
  labs(x = "Total Steps", y = "Sedentary Minutes", title = "Sedentary Minutes vs Total Steps") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=daily_activity, aes(x=date, y=TotalSteps)) +
  geom_point(color = 'blue4') +
  labs(x = "Date", y = "Total Steps", title = "Total Steps Over Time") +
  theme(plot.title = element_text(hjust = 0.5))
```

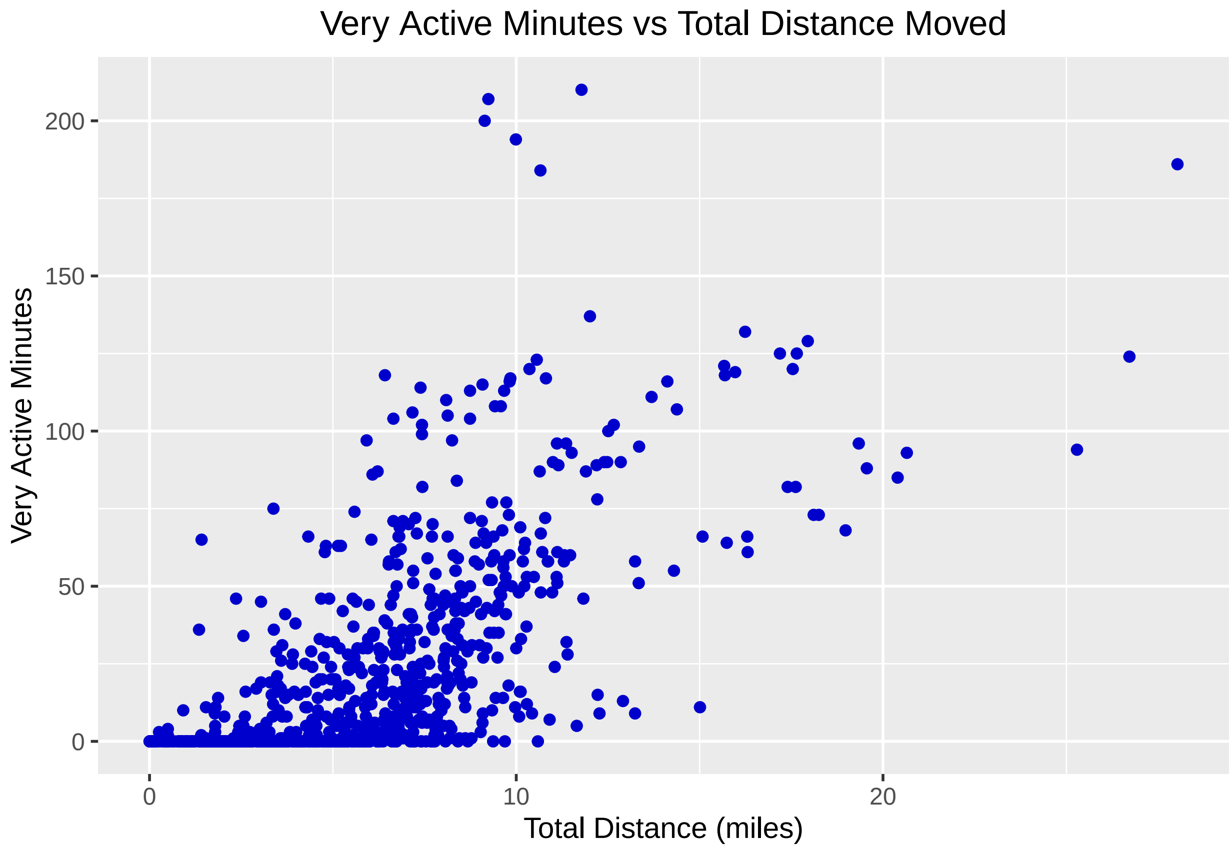



```
daily_activity %>%
  group_by(date) %>%
  #filter(TotalSteps > 0) %>%
  summarise(avgDailySteps = mean(TotalSteps)) %>%
  ggplot(aes(x=date, y=avgDailySteps)) +
  geom_point(color = 'blue4') +
  labs(x = "Date", y = "Average Daily Steps", title = "Average Daily Steps over Time") +
  theme(plot.title = element_text(hjust = 0.5))
```

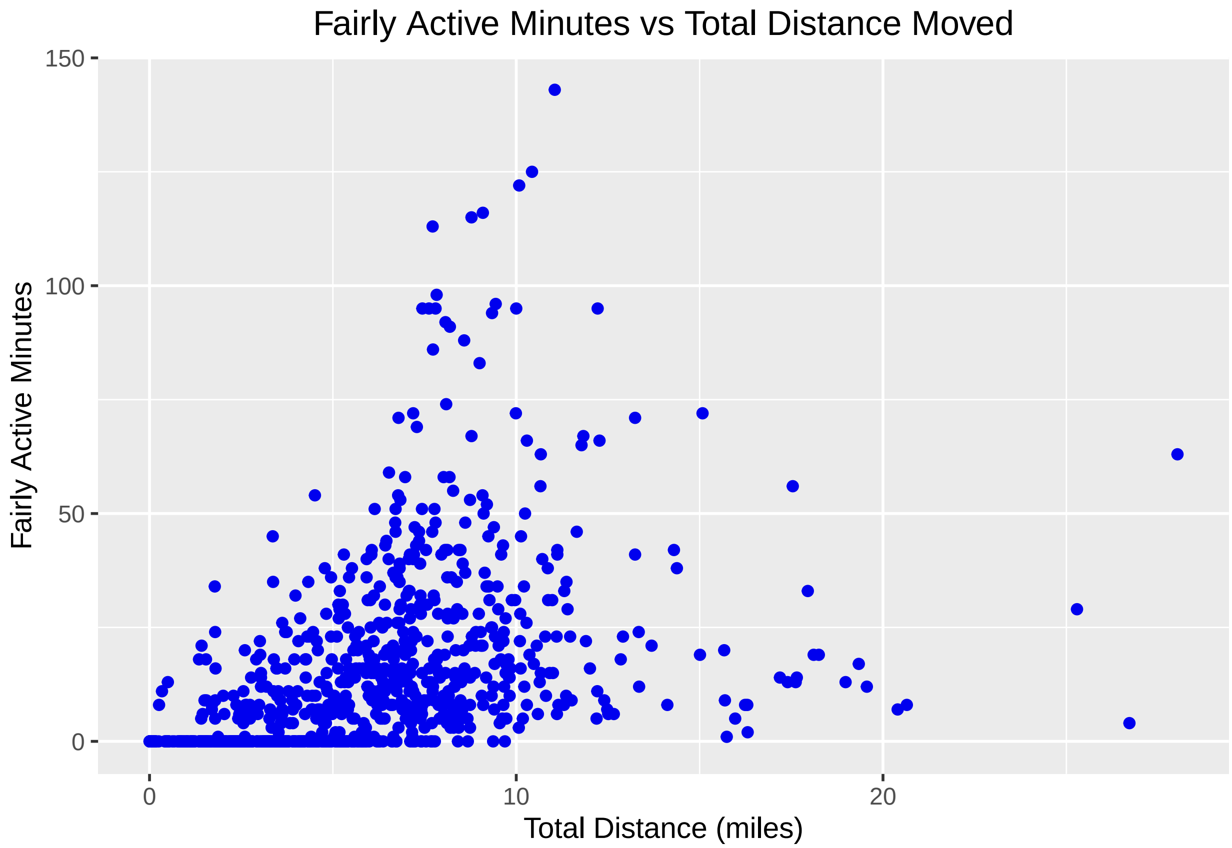


Below can be seen the relationships between the various activity levels (Very Active, Fairly Active, and Lightly Active) and the total distance moved that day. All three show a positive correlation.

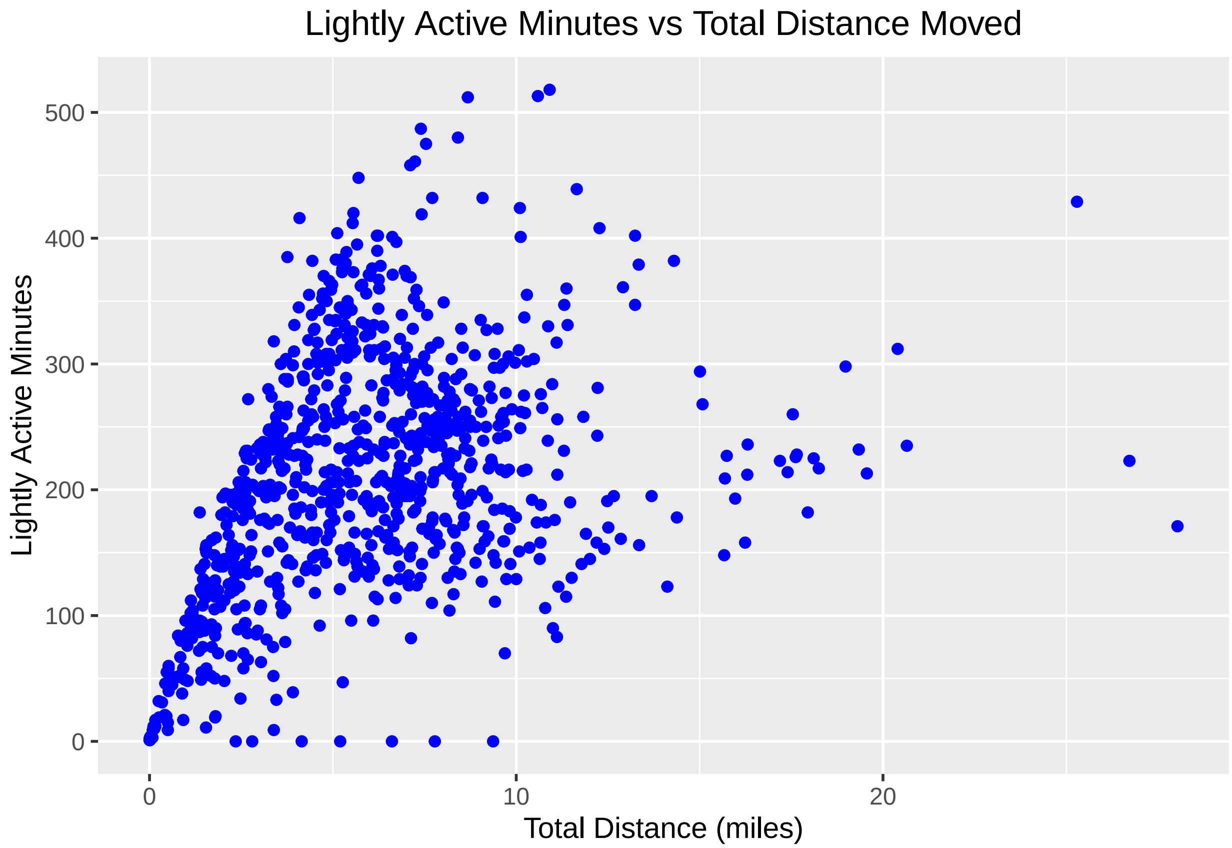
```
ggplot(data=daily_activity, aes(x=TotalDistance, y=VeryActiveMinutes)) +
  geom_point(color = 'blue3') +
  labs(x = "Total Distance (miles)", y = "Very Active Minutes", title = "Very Active Minutes vs Total Di
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=daily_activity, aes(x=TotalDistance, y=FairlyActiveMinutes)) +
  geom_point(color = 'blue2') +
  labs(x = "Total Distance (miles)", y = "Fairly Active Minutes", title = "Fairly Active Minutes vs Tota
  theme(plot.title = element_text(hjust = 0.5))
```

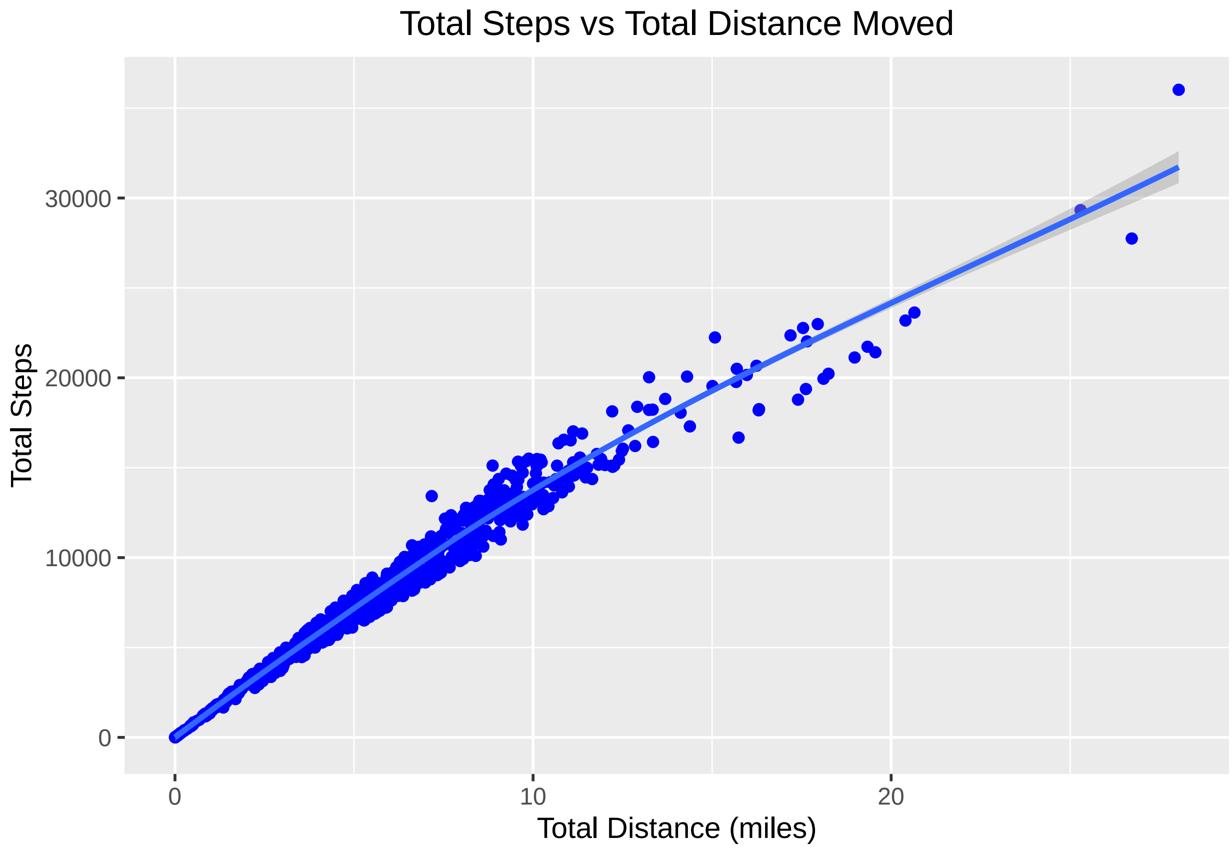



```
ggplot(data=daily_activity, aes(x=TotalDistance, y=LightlyActiveMinutes)) +
  geom_point(color = 'blue1') +
  labs(x = "Total Distance (miles)", y = "Lightly Active Minutes", title = "Lightly Active Minutes vs To
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=daily_activity, aes(x=TotalDistance, y=TotalSteps)) +
  geom_point(color = 'blue') +
  labs(x = "Total Distance (miles)", y = "Total Steps", title = "Total Steps vs Total Distance Moved") +
  geom_smooth() +
  theme(plot.title = element_text(hjust = 0.5))
```

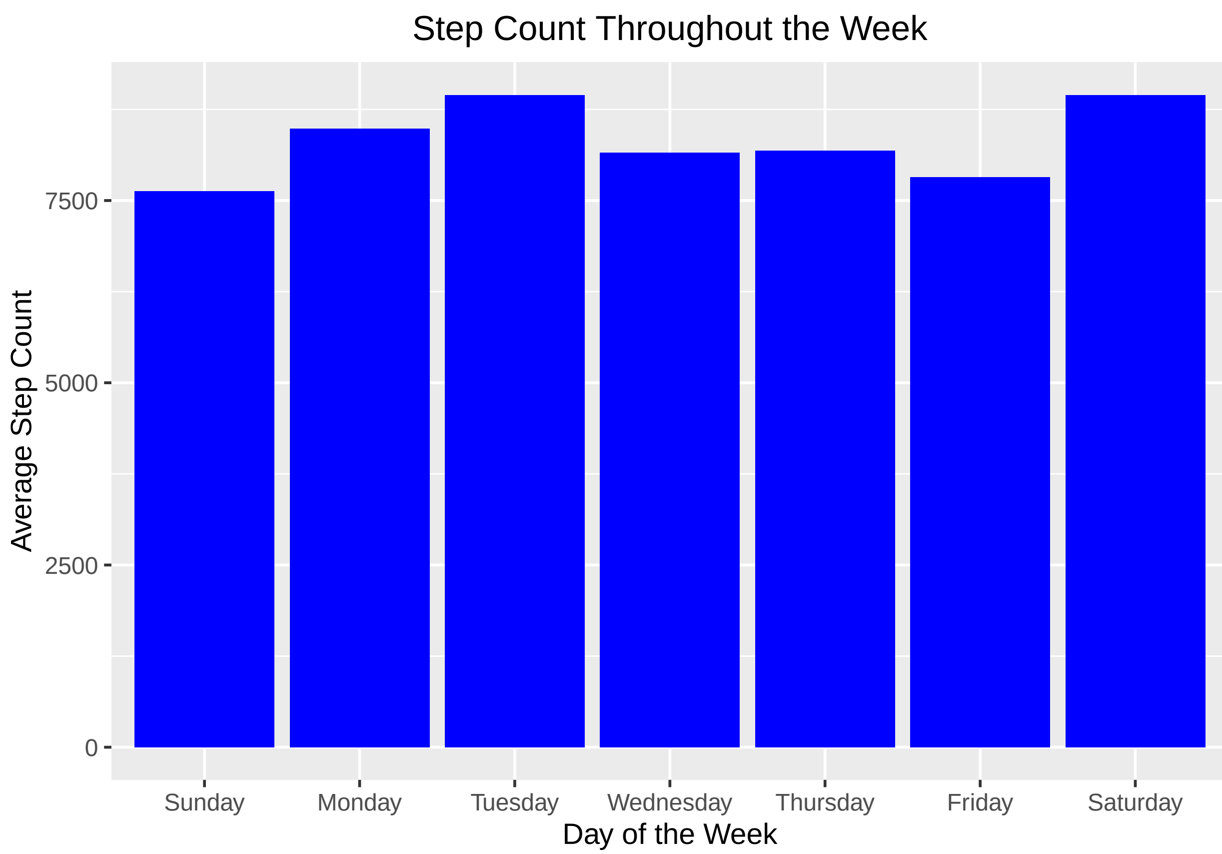
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Looking at the step count throughout the week, Tuesday and Saturday have the highest daily averages, while Sunday has the lowest.

```
#setting order for days of the week when graphing
daily_activity$weekDay <- ordered(daily_activity$weekDay, levels=c("Sunday", "Monday", "Tuesday", "Wedne

daily_activity %>%
  group_by(weekDay) %>%
  summarise (avg = mean(TotalSteps)) %>%
  ggplot(aes(x=weekDay, y=avg)) +
  geom_col(fill = 'blue1') +
  labs(x = "Day of the Week", y = "Average Step Count", title = "Step Count Throughout the Week") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
daily_activity %>%
  group_by(weekDay) %>%
  summarise (avg = mean(TotalSteps), min = min(TotalSteps), max=max(TotalSteps))
```

```
## # A tibble: 7 x 4
##   weekDay      avg   min   max
##   <ord>      <dbl> <int> <int>
## 1 Sunday     7627.    16 36019
## 2 Monday     8488.    62 20500
## 3 Tuesday    8949.     9 23186
## 4 Wednesday 8158.      4 23629
## 5 Thursday   8185.    17 21129
## 6 Friday     7821.    42 21727
## 7 Saturday   8947.    31 29326
```

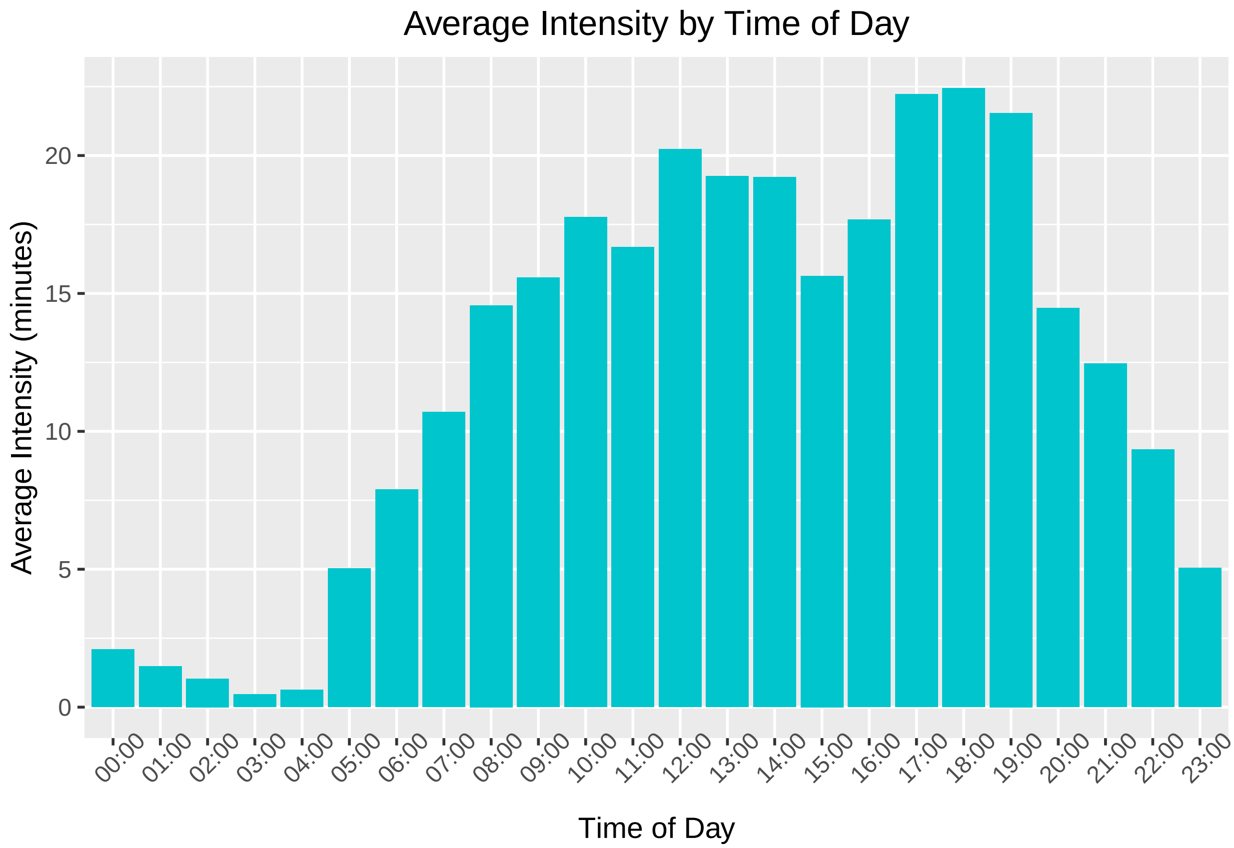
Hourly Trends

By plotting the hourly_data it can be seen that there is a positive relationship between the total steps a user takes and their total intensity minutes. Additionally, the users' hourly data shows that they were most active around lunch time (12-2pm) and immediately after work (5-7pm).

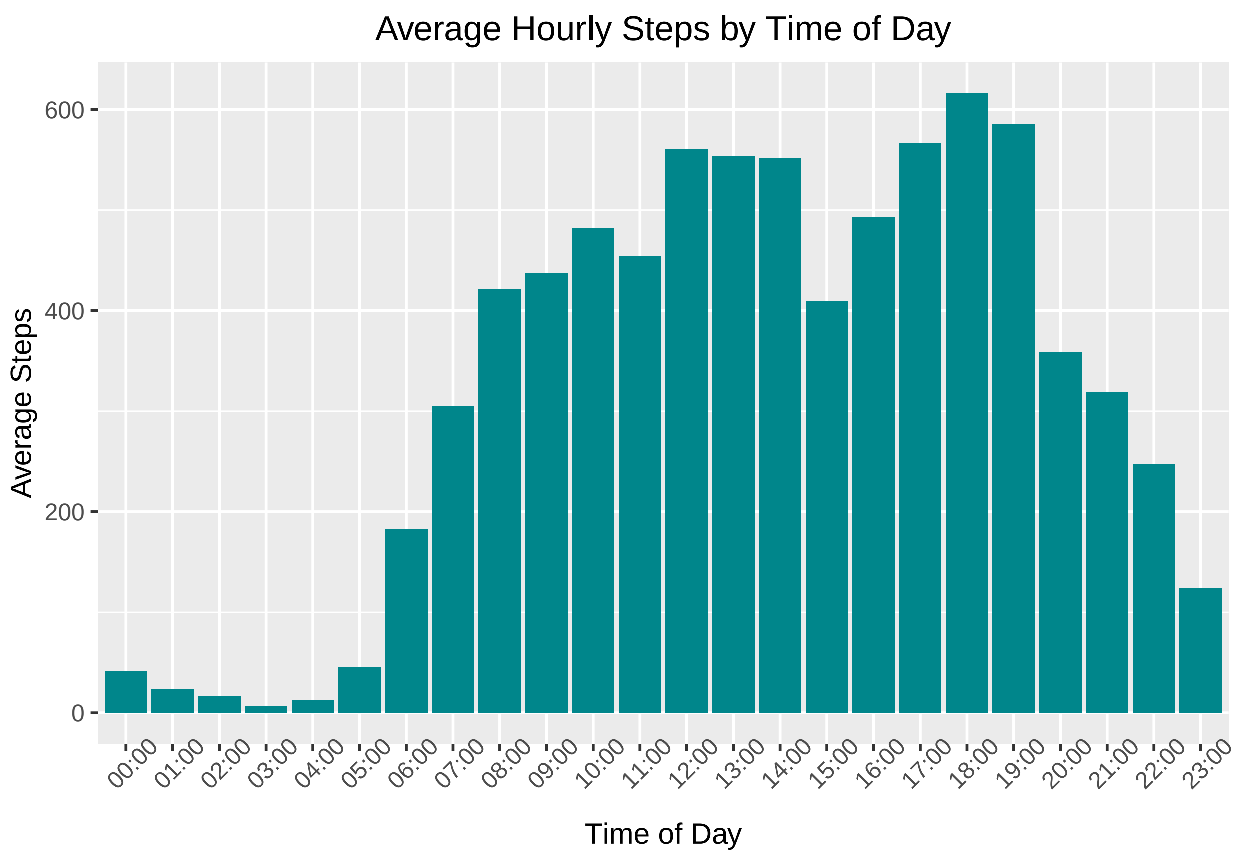
```r
ggplot(data=hourly_data, aes(x=StepTotal, y=TotalIntensity)) +
  geom_point(color = 'blue') +
  labs(x = "Total Steps", y = "Total Intensity"
```

```
       , title = "Total Intensity Relationship to Total Steps") +
  theme(plot.title = element_text(hjust = 0.5))
```

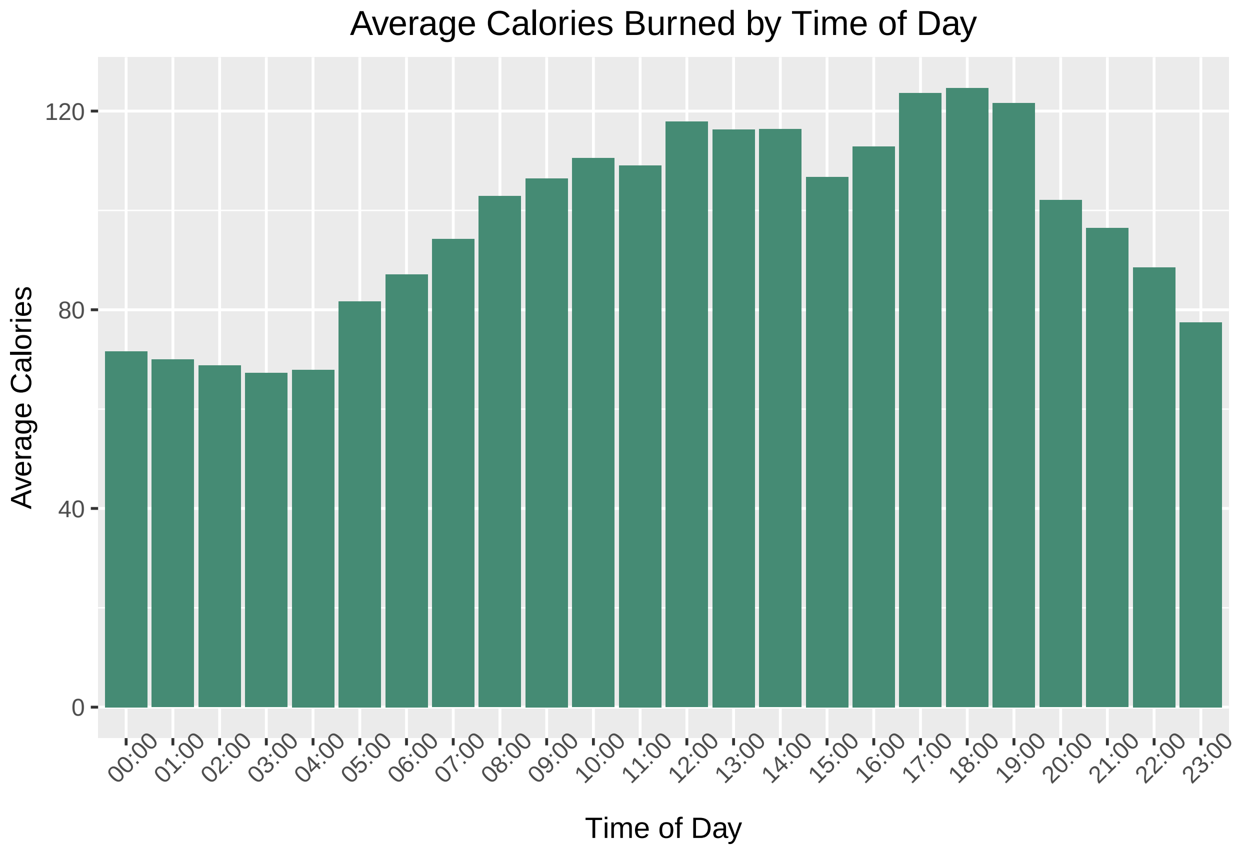

```
hourly_data %>%
  group_by(hour) %>%
  summarise(averageIntensity = mean(TotalIntensity)) %>%
  ggplot(aes(x=hour,y=averageIntensity))+geom_col(fill = 'turquoise3') +
  labs(x = 'Time of Day', y = 'Average Intensity (minutes)',
       title = 'Average Intensity by Time of Day') +
  theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45))
```




```
hourly_data %>%
  group_by(hour) %>%
  summarise(avgSteps = mean(StepTotal)) %>%
  ggplot(aes(x=hour,y=avgSteps))+geom_col(fill = 'turquoise4') +
  labs(x = 'Time of Day', y = 'Average Steps', title = 'Average Hourly Steps by Time of Day') +
  theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45))
```



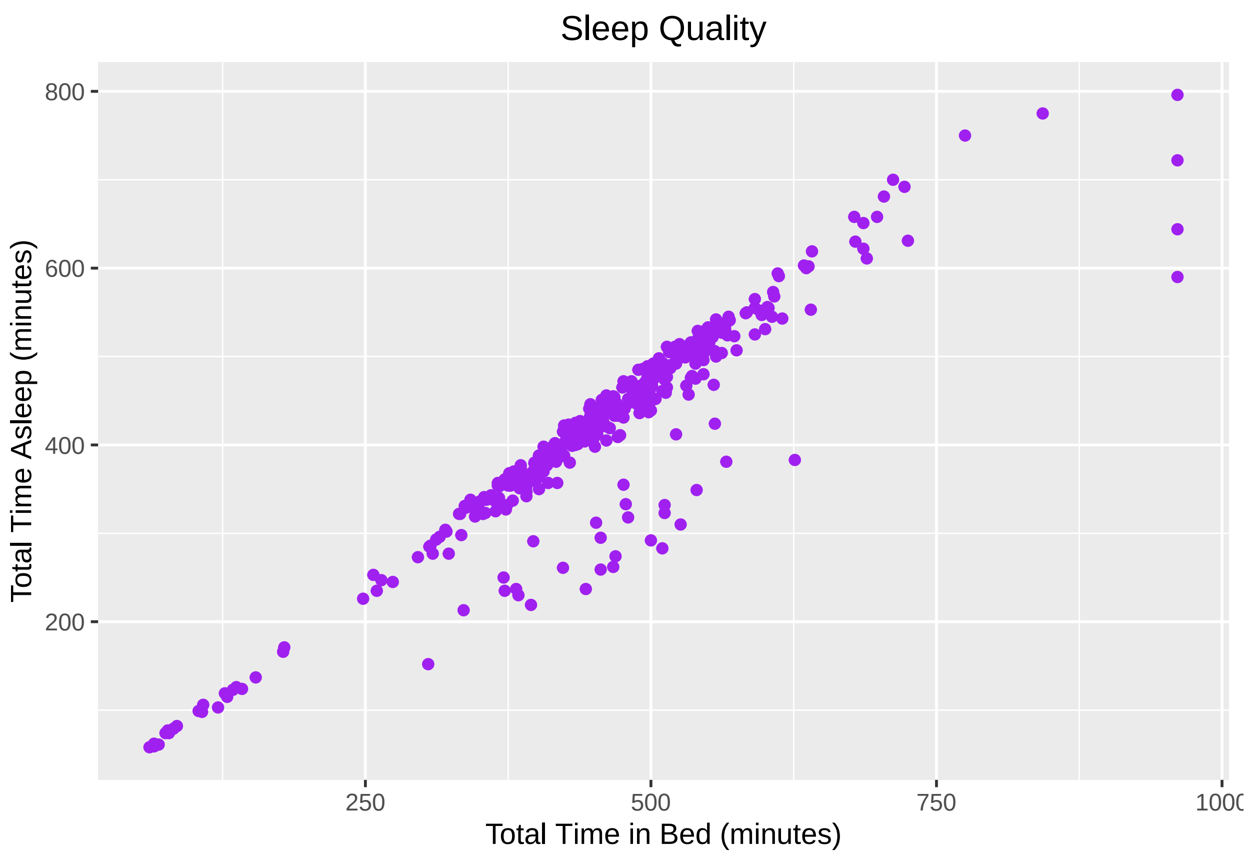

```
hourly_data %>%
  group_by(hour) %>%
  summarise(avgCalories = mean(Calories)) %>%
  ggplot(aes(x=hour,y=avgCalories))+geom_col(fill = 'aquamarine4') +
  labs(x = 'Time of Day', y = 'Average Calories'
       , title = 'Average Calories Burned by Time of Day') +
  theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45))
```



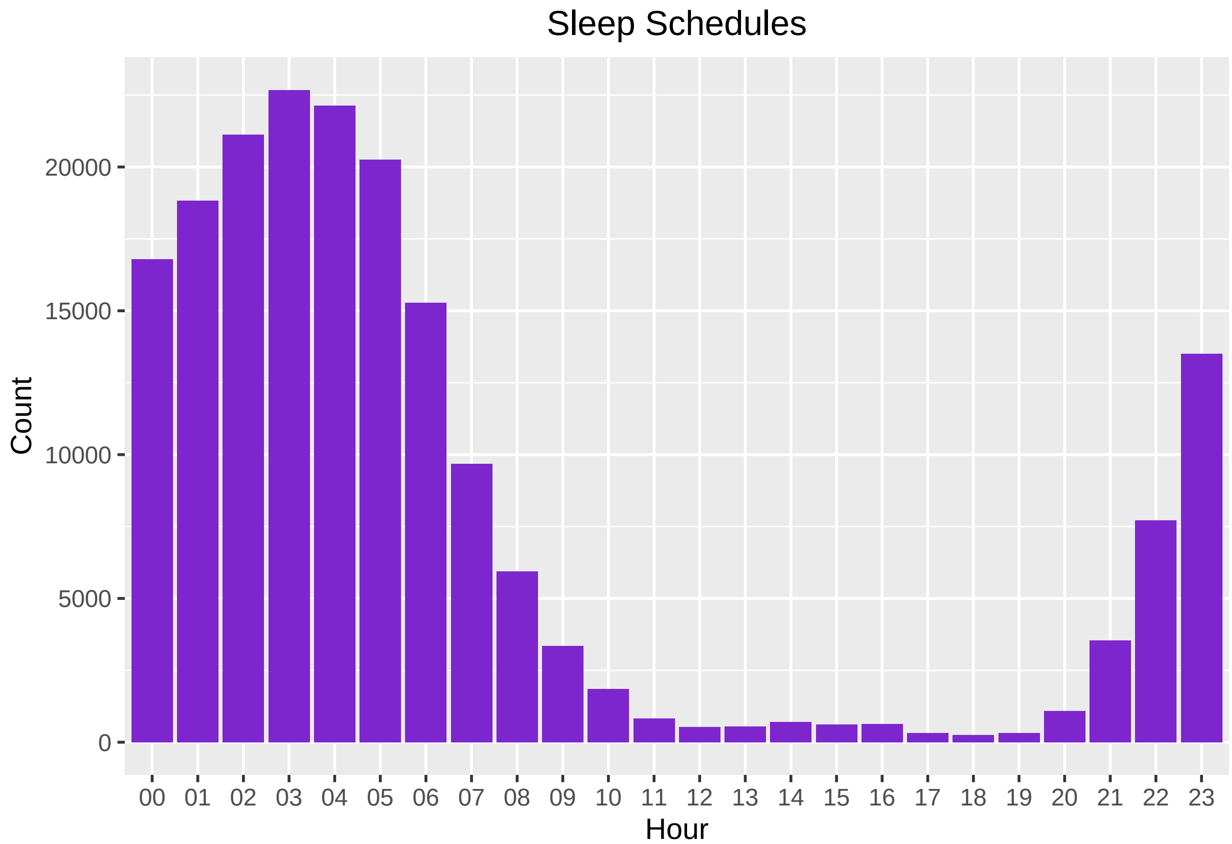
Sleep Trends

Looking at some plots to understand participants sleeping patterns. As is expected, there is a linear correlation between time spent in bed and time asleep. There is some deviation from this trend mainly between 370 and 570 minutes in bed. This could be the indicative of people having trouble falling or staying asleep.

```
ggplot(data=sleep_day, aes(x=TotalTimeInBed, y=TotalMinutesAsleep)) + geom_point(color = "purple") +
  labs(x = "Total Time in Bed (minutes)", y = "Total Time Asleep (minutes)", title = "Sleep Quality") +
  theme(plot.title = element_text(hjust = 0.5))
```

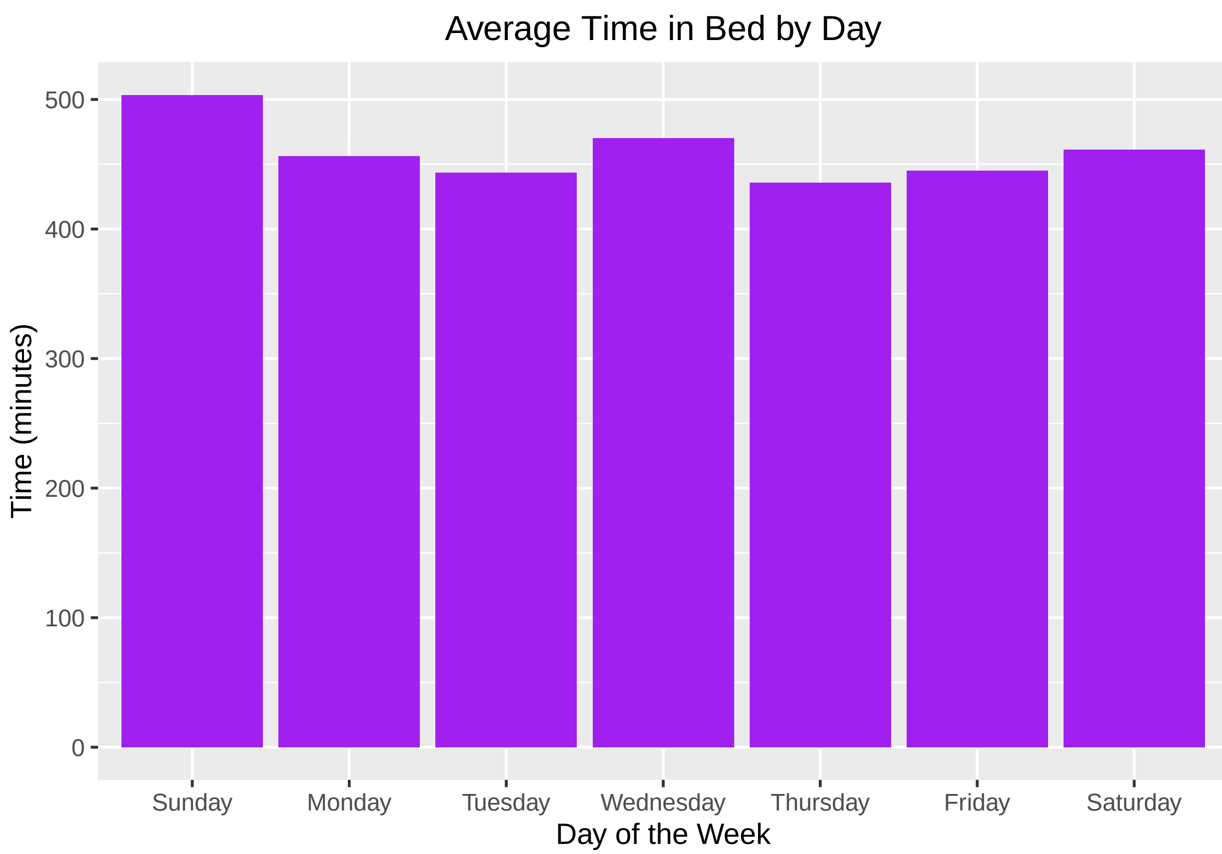



```
ggplot(data=minute_sleep, aes(x=hour)) + geom_bar(fill = "purple3") +
  labs(x = "Hour", y = "Count", title = "Sleep Schedules") +
  theme(plot.title = element_text(hjust = 0.5))
```

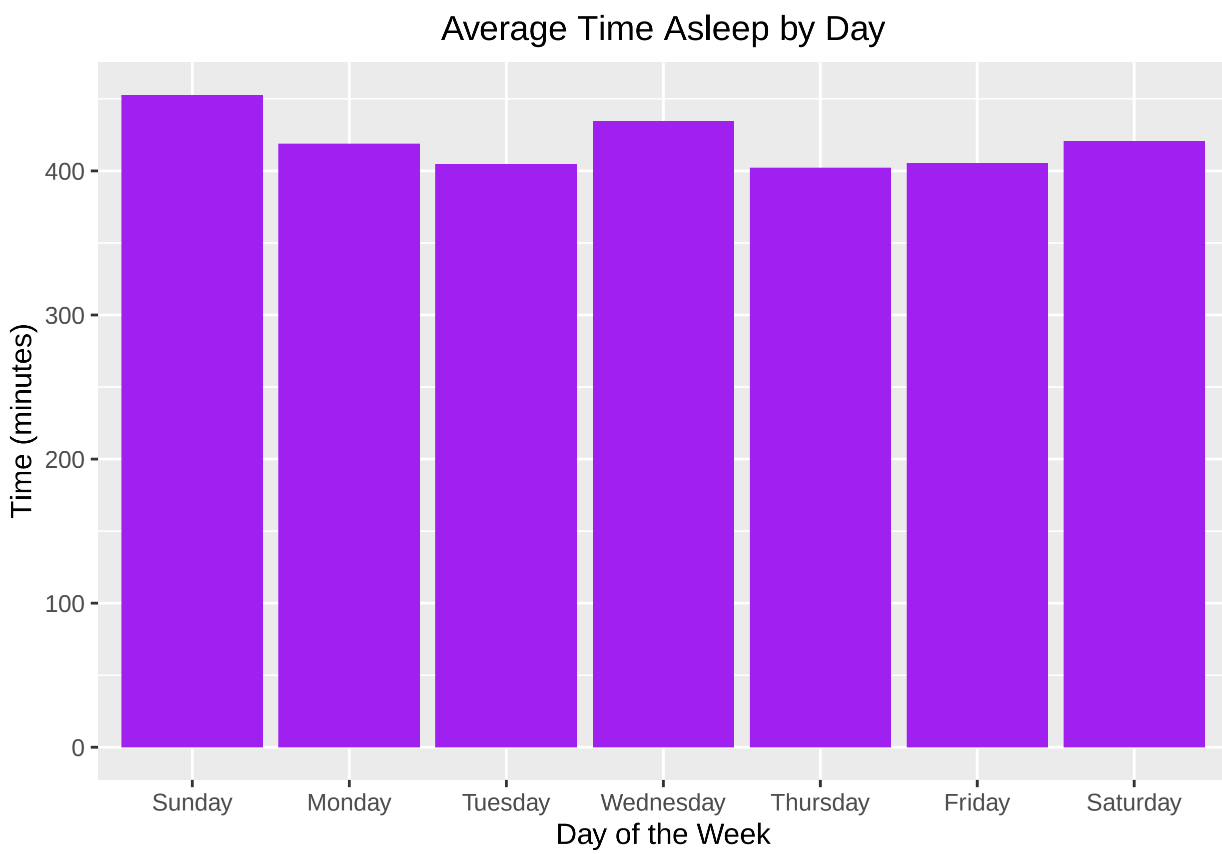


The day users slept the most was Sunday, while the day of the week they had the least amount of sleep was Thursday. Monday through Thursday, participants spent the least amount of time awake in bed, while on the weekends they spent more time awake and in bed, with the maximum occurring on Sunday. This is likely due to having more free time on the weekends to relax in bed.

```
#setting order for days of the week when graphing
sleep_day$weekDay <- ordered(sleep_day$weekDay, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Th
sleep_day %>%
  group_by(weekDay) %>%
  summarise(average_bed = mean(TotalTimeInBed)) %>%
  ggplot(aes(x=weekDay, y=average_bed)) + geom_col(fill = 'purple') +
  labs(x = "Day of the Week", y = "Time (minutes)", title = "Average Time in Bed by Day") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
sleep_day %>%
  group_by(weekDay) %>%
  summarise(average_sleep = mean(TotalMinutesAsleep)) %>%
  ggplot(aes(x=weekDay, y=average_sleep)) + geom_col(fill = 'purple') +
  labs(x = "Day of the Week", y = "Time (minutes)", title = "Average Time Asleep by Day") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
sleep_day %>%
  group_by(weekDay) %>%
  summarise(Time_Bed = mean(TotalTimeInBed), Time_Asleep = mean(TotalMinutesAsleep)
            , Awake_Bed = Time_Bed - Time_Asleep)
```

```
## # A tibble: 7 x 4
##   weekDay   Time_Bed Time_Asleep Awake_Bed
##   <ord>        <dbl>       <dbl>     <dbl>
## 1 Sunday        504.        453.      50.8
## 2 Monday        456.        419.      37.3
## 3 Tuesday       443.        405.      38.8
## 4 Wednesday     470.        435.      35.3
## 5 Thursday      436.        402.      33.4
## 6 Friday        445.        405.      39.6
## 7 Saturday      461.        421.      40.5
```

Merging sleep and daily activity

Combining these two to investigate any potential relationships. Note that some data will be lost using this merge since sleep_day has fewer participants/unique Ids than combined_data.

```
combined_data <- merge(sleep_day, daily_activity, by = c("Id","date"), all=FALSE)
n_distinct(combined_data$Id) #number of users dropped from 33 to 24
```

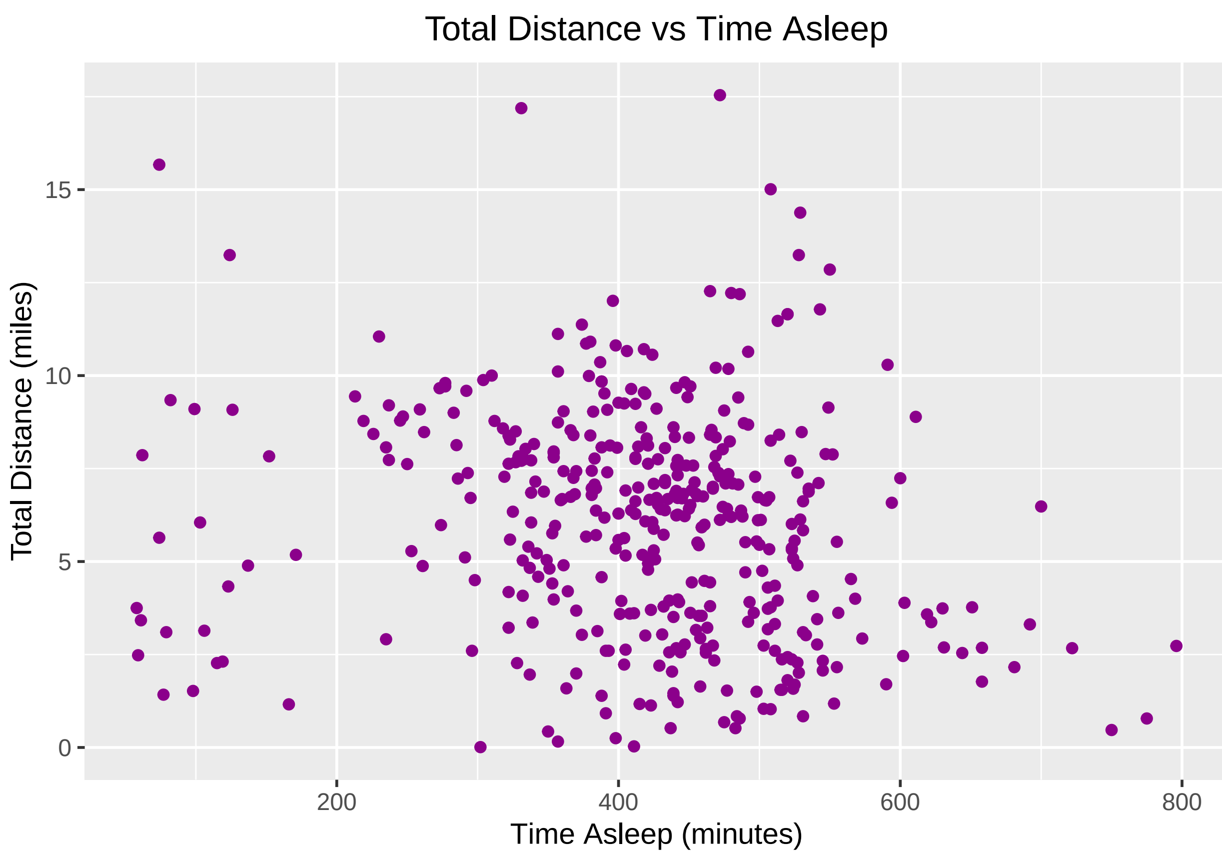
```
## [1] 24
```

```
sum(is.na(combined_data)) #checking for NA values
```

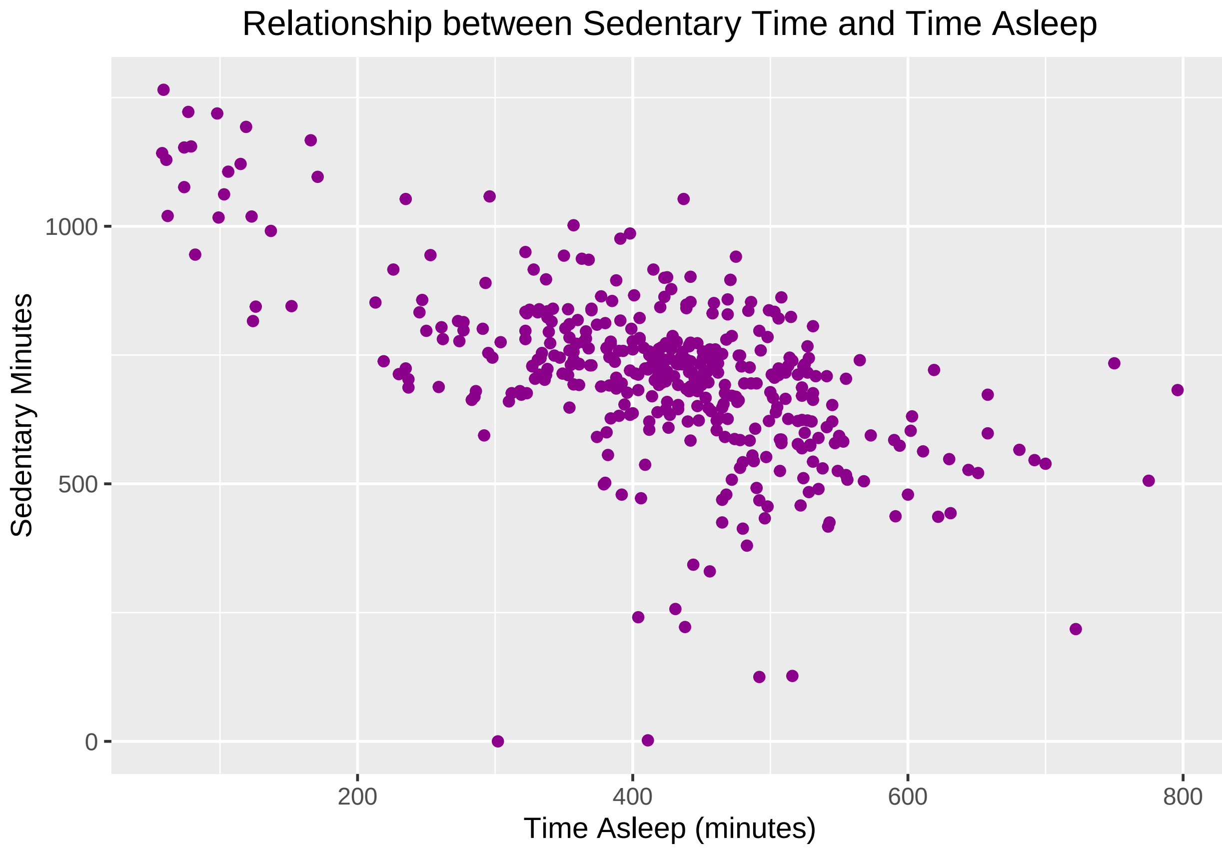
```
## [1] 0
```

There is a moderate negative correlation between sedentary minutes and time spent asleep, however given the limited amount of data points, this would require further investigation to know what's driving this. Outside of that, there doesn't appear to be a notable relationship between time asleep or quality of sleep and any of the variables explored in this data set.

```
ggplot(data=combined_data, aes(x=TotalMinutesAsleep, y=TotalDistance)) +
  geom_point(color = "magenta4") +
  labs(x="Time Asleep (minutes)", y="Total Distance (miles)", title="Total Distance vs Time Asleep") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=combined_data, aes(x=TotalMinutesAsleep, y=SedentaryMinutes)) +
  geom_point(color = "magenta4") +
  labs(x="Time Asleep (minutes)", y="Sedentary Minutes", title="Relationship between Sedentary Time and
  theme(plot.title = element_text(hjust = 0.5))
```




```
cor(combined_data$TotalSteps,combined_data$TotalMinutesAsleep)
```

```
## [1] -0.1868665
```

```
cor(combined_data$SedentaryMinutes,combined_data$TotalMinutesAsleep)
```

```
## [1] -0.599394
```

```
cor(combined_data$SedentaryMinutes,combined_data$TotalMinutesAsleep/combined_data$TotalTimeInBed)
```

```
## [1] 0.01967645
```

```
cor(combined_data$LightlyActiveMinutes,combined_data$TotalMinutesAsleep/combined_data$TotalTimeInBed)
```

```
## [1] 0.1323084
```

Weight Activity Trends

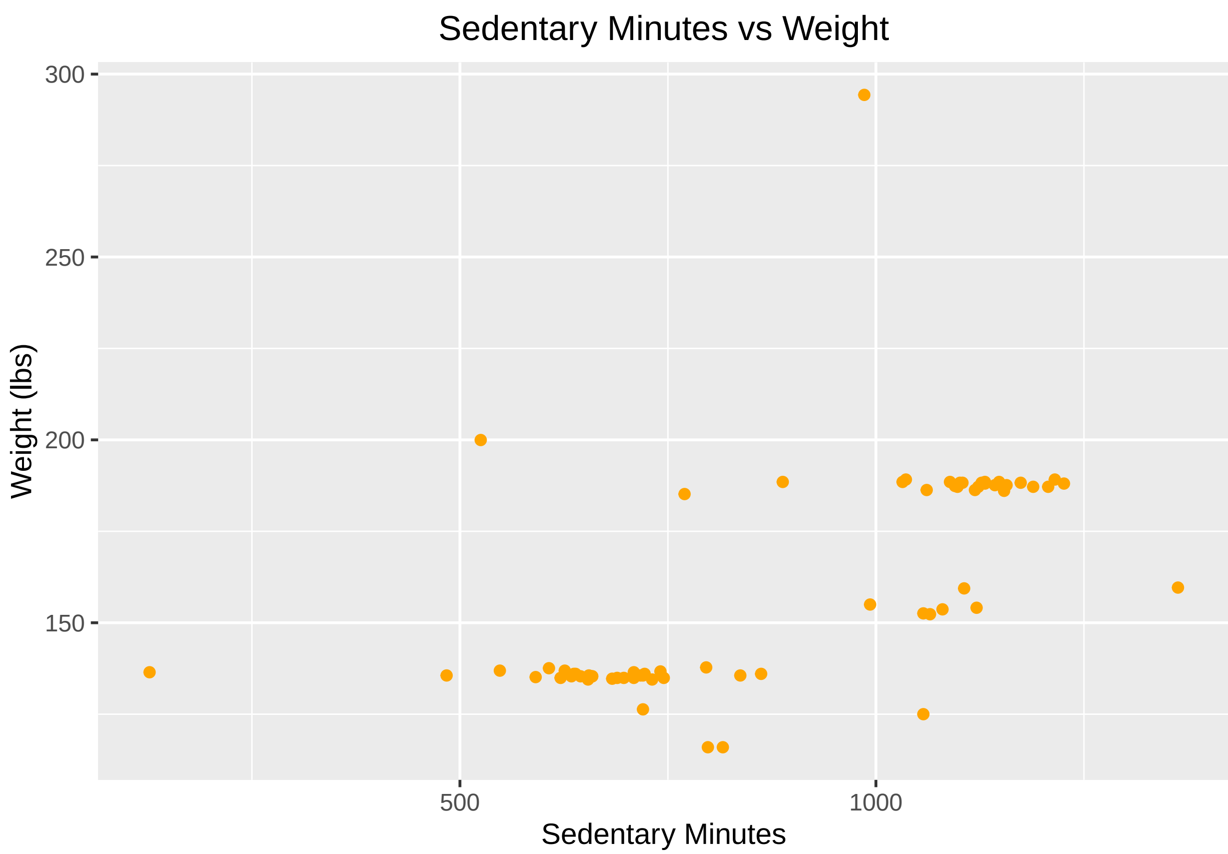
The weight_log and daily_activity dataframes were merged in order to investigate any potential relationship between the two. Due to the severely limited sample size of the weight log data set (only 8 participants), this exercise is just being done out of curiosity and no results will be able to be considered statistically valid.

The plots initially seem to indicate that there are two segments of the population (likely one for men and one for women) based on weight as the weight data seem to cluster around two weights. Upon closer inspection the two clusters are actually caused by two participants supplying significantly more data points than the rest and skewing the data.

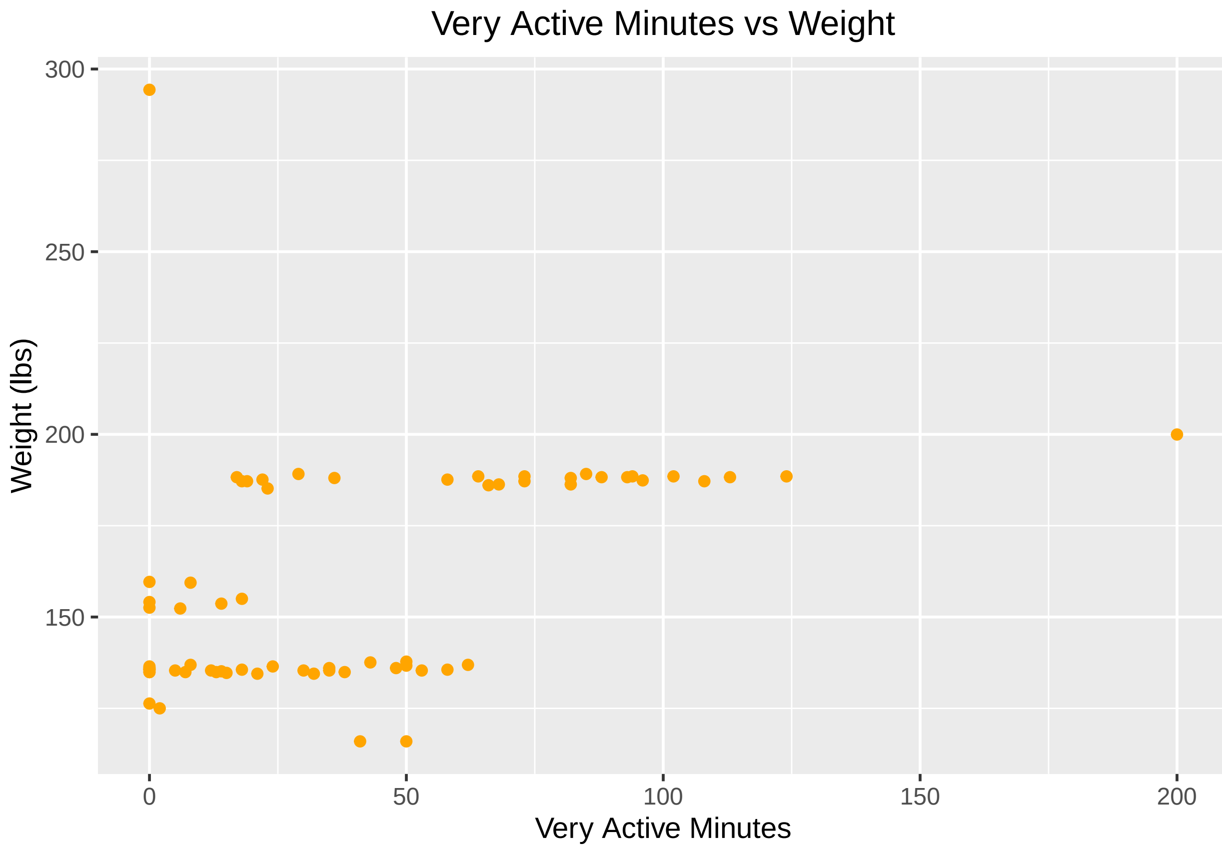
```
weight_log$date <- as.Date(weight_log$Date, "%m/%d/%Y")
weight_activity <- merge(daily_activity, weight_log, by = c("Id","date"), all = FALSE)

ggplot(data=weight_activity, aes(x=Calories, y=WeightPounds)) +
  geom_point(color = 'orange1') +
  labs(x="Calories",y="Weight (lbs)", title="Relationship between Calories Burned and Weight") +
  theme(plot.title = element_text(hjust = 0.5))
```

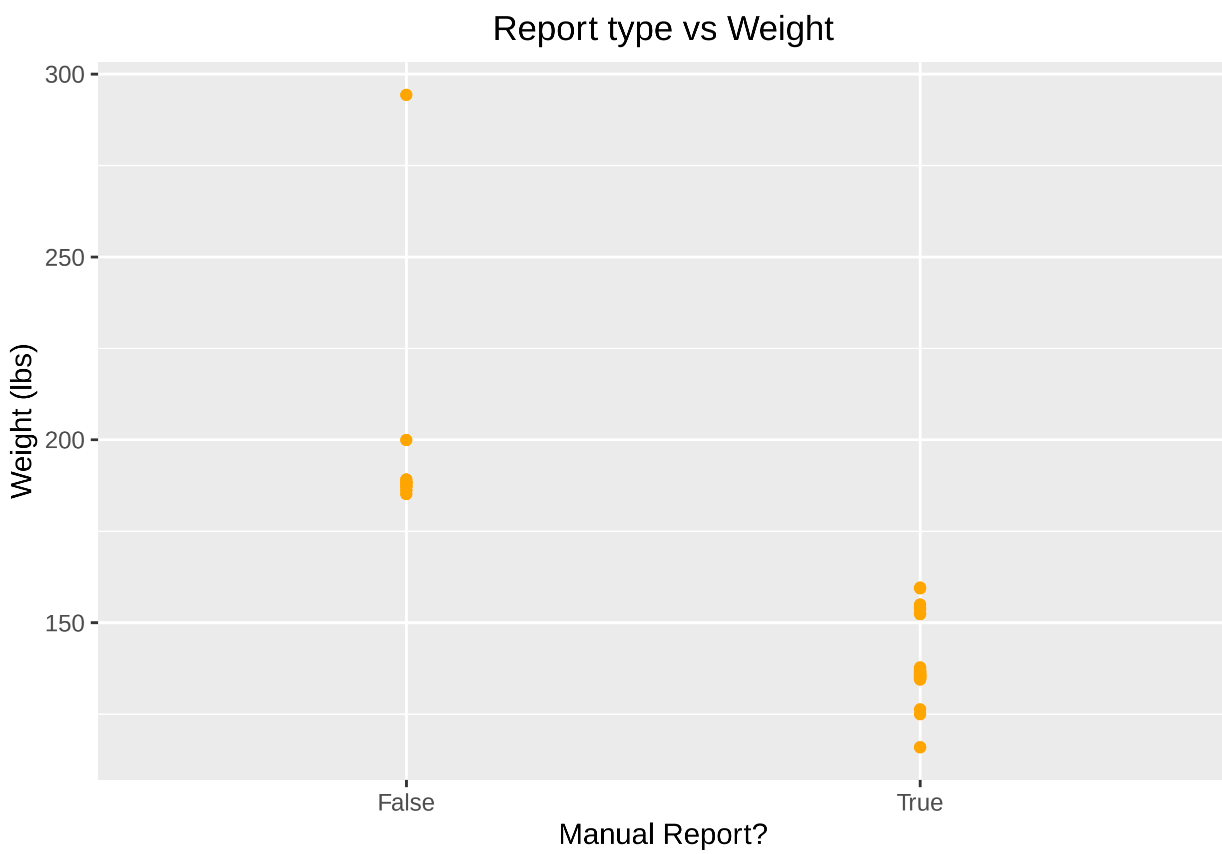

```
ggplot(data=weight_activity, aes(x=SedentaryMinutes, y=WeightPounds)) +
  geom_point(color = 'orange1') +
  labs(x="Sedentary Minutes", y="Weight (lbs)",title="Sedentary Minutes vs Weight") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=weight_activity, aes(x=VeryActiveMinutes, y=WeightPounds)) +
  geom_point(color = 'orange1') +
  labs(x="Very Active Minutes", y="Weight (lbs)",title="Very Active Minutes vs Weight") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=weight_activity, aes(x=IsManualReport, y=WeightPounds)) +
  geom_point(color = 'orange1') +
  labs(x="Manual Report?", y="Weight (lbs)",title="Report type vs Weight") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(data=weight_activity, aes(x=Calories, y=WeightPounds)) +
  geom_point(color = 'orange1') + facet_wrap(~Id) +
  labs(x="Calories", y="Weight (lbs)", title="Weight vs Calories by User") +
  theme(plot.title = element_text(hjust = 0.5))
```

Logged Activity Trends

Users had the ability to log physical exercise/activity if they desired. Unfortunately, not all participants did, so this sample size is significantly smaller with only 2 participants providing a total of 20 data points for the sleep-containing group and 4 participants providing 32 data points if sleep data is excluded. This means the data can not provide any statistically useful information. However, it is interesting to investigate logged activity distance to see if there are any relationships with intentionally logged activities compared to just using passive step count.

Participants who logged activities averaged 12,042 steps, while those that did not averaged 8,176. Additionally, users who logged an activity, on average, had a higher sleep ratio (total minutes asleep)/(total minutes in bed) than participants who did not log an activity.

```
filtered <- combined_data %>% filter(LoggedActivitiesDistance > 0)
nrow(filtered)
```

```
## [1] 20
```

```
n_distinct(filtered$Id)
```

```
## [1] 2
```

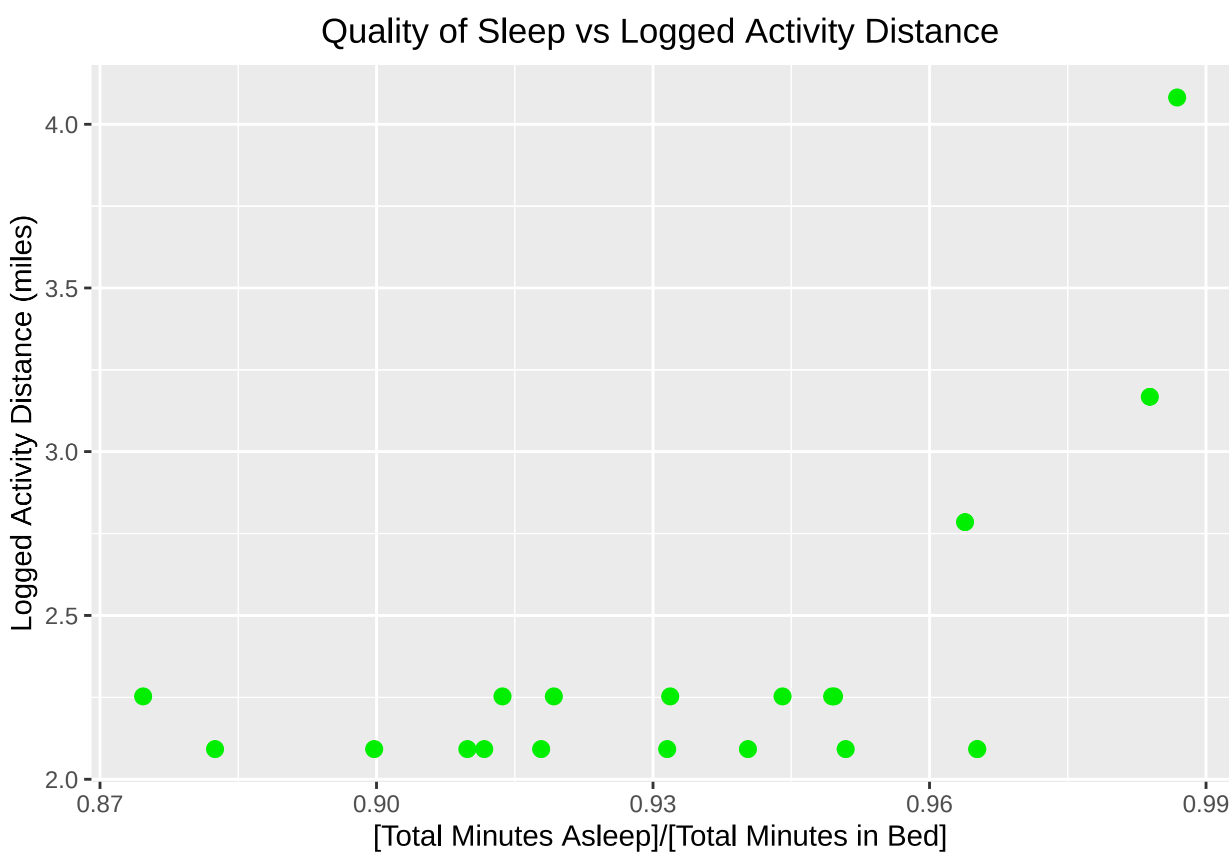
```
filtered2 <- daily_activity %>% filter(LoggedActivitiesDistance > 0)
nrow(filtered2)
```

```
## [1] 32
```

```r
n_distinct(filtered2$Id)
```

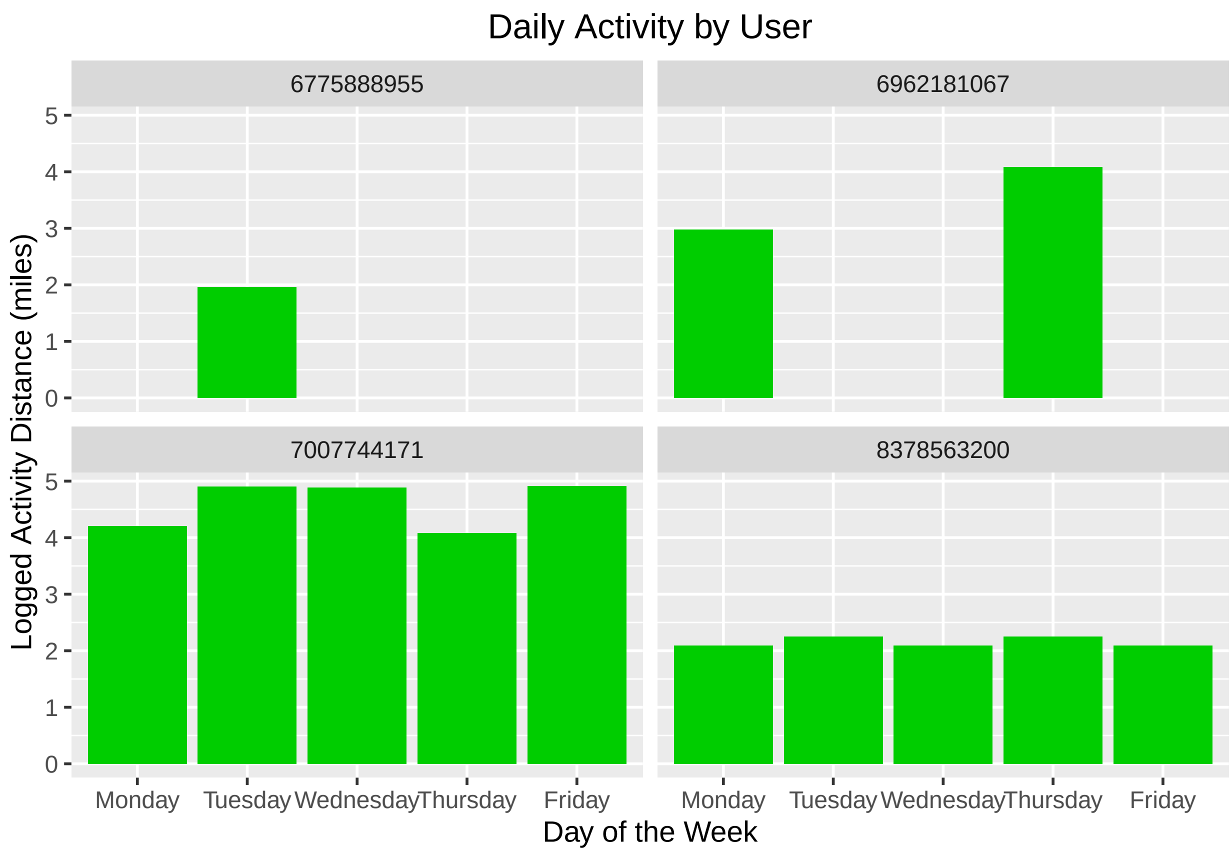
```
## [1] 4
```

```r
combined_data %>%
  filter(LoggedActivitiesDistance > 0) %>%
  ggplot(aes(x=(TotalMinutesAsleep/TotalTimeInBed), y=LoggedActivitiesDistance)) +
  geom_point(color = 'green2', size=2.5) +
  labs(y="Logged Activity Distance (miles)",x="[Total Minutes Asleep]/[Total Minutes in Bed]"
       ,title = "Quality of Sleep vs Logged Activity Distance") +
  theme(plot.title = element_text(hjust = 0.5))
```




```r
daily_activity %>%
  filter(LoggedActivitiesDistance > 0) %>%
  group_by(weekDay, Id) %>%
  summarise(avgLoggedActivity = mean(LoggedActivitiesDistance)) %>%
  ggplot(aes(x=weekDay, y=avgLoggedActivity)) +
  geom_col(fill = 'green3') +
  labs(y="Logged Activity Distance (miles)",x="Day of the Week"
       ,title = "Daily Activity by User") +
  theme(plot.title = element_text(hjust = 0.5)) +
   facet_wrap(~Id)
```

```
## `summarise()` has grouped output by 'weekDay'. You can override using the
## `.groups` argument.
```




```
combined_data %>%
  filter(LoggedActivitiesDistance > 0) %>%
  summarise(avgSleepQuality = mean(TotalMinutesAsleep/TotalTimeInBed))
```

```
##   avgSleepQuality
## 1       0.9345979
```

```
combined_data %>%
  filter(LoggedActivitiesDistance <= 0) %>%
  summarise(avgSleepQuality = mean(TotalMinutesAsleep/TotalTimeInBed))
```

```
##   avgSleepQuality
## 1       0.9158619
```

```
daily_activity %>%
  filter(LoggedActivitiesDistance > 0) %>%
  summarise(avgSteps = mean(TotalSteps))
```

```
##   avgSteps
## 1  12042.5
```

```
daily_activity %>%
  filter(LoggedActivitiesDistance <= 0) %>%
  summarise(avgSteps = mean(TotalSteps))
```

```
##   avgSteps
## 1 8176.024
```

Export data for investigations in Tableau

Below was used to export summary files for further analysis in Tableau. The tableau viz can be found here

```
write.csv(daily_activity, file = '~/Documents/DAILY_ACTIVITY.csv')
write.csv(hourly_data, file = '~/Documents/HOURLY_DATA.csv')
write.csv(combined_data, file = '~/Documents/COMBINED_DATA.csv')
write.csv(sleep_day, file = '~/Documents/SLEEP_DAY.csv')
```

Conclusions

Overall, device usage seemed to be focused on tracking daily activity, with all 33 participants collecting data on their daily step activity for at least some portion of the study. The next most popular area to track was sleep with 24 of the 33 participants tracking their sleep, and finally the least important data point seemed to be weight. This could indicate the relative importance of these metrics to customers or could be the result of use preference for the devices (e.g. not wanting to wear a device while sleeping) or ease of use (e.g. not wanting to manual enter weight each day).

Regarding daily activity trends, activity levels did not very significantly from day to day for the average participant. Throughout the day activity levels seemed to increase with a relative maximum occurring around lunch time (12-2pm) and an absolute maximum occurring at the end of the typical work day (5-7pm). There also seemed to be a dip in activity post-lunch (around 3pm).

Furthermore, participants who logged activities averaged 12,042 steps, while those that did not averaged 8,176. If that pattern holds true for larger sample sizes, it could prove useful for the marketing team as a way to encourage smart device users to increase their steps. Additionally, since participants who logged an activity, on average, had a higher sleep ratio, this could indicate that participants who logged an activity had an easier time falling asleep and staying asleep compared to those who did not and be used as a way to entice users to exercise more or help if they are having trouble sleeping.

On average, participants got 6 hrs 59.5 minutes of sleep each night, with users getting the most sleep on Wednesdays and Sundays. There was also a slightly negative correlation between sedentary minutes and time spent asleep, so the less sedentary a user was, the more sleep they tended to get. As mentioned previously, further analysis needs to be done with more participants for both the sleep and the weight data sets to be able to gain more accurate and useful insights.

Recommendations

- If we notice there is a significant amount of time where the user is in bed, but not asleep, send notification directing them to the Bellabeat website, where there are posts with tips to help fell asleep faster and get better quality sleep.
- Incorporate daily workout reminders/recommendations to help users increase their total daily activity.
  - Time these reminders to be around 5:00 pm or slightly before, as this is when users seem to be the most naturally inclined to be active.

- If users have been sedentary for an extended period of time (>1 hr), send a notification encouraging them to get up and walk around as decreases in sedentary time have been associated with increased daily step count, as well as more sleep.
  - Similarly, send reminders in the afternoon (around 3 pm) to remind users to be active as this is when their tends to be a dip in activity.
- As this is a fitness tracker specifically targeted towards the female market, add an optional log to input cycle information, which could be used in providing recommendations for everything from sleep, to hydration, exercise, etc.
- No strong relationships with sleep data and overall activity, however, recommend exploring larger sample size as it is expected there would be some sort of relationship.
- Incorporate weight measurements automatically via a Bellabeat-branded scale that could be associated with a user or via bluetooth connection to other available scales to enable easier tracker of weight.