Capstone Case Study: Bellabeat

Andrea Hayling

2022-05-16

INTRO

How Can a Wellness Technology Company Play It Smart?

Bellabeat data analysis case study, following the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

SCENARIO

You are a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide marketing strategy for the company. You will present your analysis to the Bellabeat executive team along with your high-level recommendations for Bellabeat's marketing strategy.

ASK

Analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices.

Questions to help guide analysis:

1.What are some trends in smart device usage?

2.How could these trends apply to Bellabeat customers?

3.How could these trends help influence Bellabeat marketing strategy?

PREPARE

Use public data that explores smart device users daily habits. Specific data set from Kaggle Fitbit Fitness Tracker Data. This data set consists of about thirty Fitbit users personal tracker data. It includes information about daily activity, steps, sleep monitoring, and heart rate. Source: Kaggle Fitbit Data Set

I installed the packages and the files I will be working with.

I will use the following packages to work on the data.

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

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

```
install.packages("here")
```

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

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

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

```r
install.packages("janitor")
```

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

```r
library(tidyverse)
```

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

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## 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()
```

```r
library(here)
```

```
## here() starts at /cloud/project
```

```r
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(lubridate)
```

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

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

I will be using the following files to analyze the information and find any insights or trends in the data.

```r
library(readr)
DailyActivity <- read_csv("dailyActivity_merged.csv")
```

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

```r
library(readr)
SleepDay <- read_csv("sleepDay_merged.csv")
```

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

```r
library(readr)
WeightLog <- read_csv("weightLogInfo_merged.csv")
```

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

PROCESS

I will use R programming language as my tool of choice to work on this data set. I will check the files selected for errors or inconsistencies. I will use glimpse and head to get a brief view of the data columns in the files I will also use n_distinct to see how many uniquic participants Ids are in each file.

```r
glimpse(DailyActivity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 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        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```r
n_distinct(DailyActivity$Id)
```

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

```r
head(SleepDay)
```

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

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

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

```r
head(WeightLog)
```

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

```r
n_distinct(WeightLog$Id)
```

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

After glancing at the data in all three files I can see that there are actually 33 participants instead of 30 in the main file dailyActivity. A less participants logged their sleep and even less logged their weight information. Right away you can see there are a lot of limitations for this data set. The main problem for this data set is the time frame is very limited to only about a month worth of data. This is a very small group of participants to use for analysis. The data is not that current, it's from 6yrs ago. Not all of the files have the same number of participants, so there are lots of missing information.

There are some data type issues that need to be address so we can work with this data. I need to update the Date and Id columns in the files to the correct data types. Updating the Date columns to the Date data type and the Id columns to the character data type so they become string values instead. I also updated one column in the sleep file as it is the same name as the file name and it could cause confusion.

```r
DailyActivity$ActivityDate <- as.Date(DailyActivity$ActivityDate, format = "%m/%d/%Y")
DailyActivity$Id <- as.character(DailyActivity$Id)
SleepDay <- rename(SleepDay, SleepDate = SleepDay)
SleepDay$SleepDate <- as.Date(SleepDay$SleepDate, format = "%m/%d/%Y")
SleepDay$Id <- as.character(SleepDay$Id)
WeightLog$Date <- as.Date(WeightLog$Date, format = "%m/%d/%Y")
WeightLog$Id <- as.character(WeightLog$Id)
```

Also need to run checks for duplicates in the files so we can make sure the data is more usable. After running the function only the SleepDay file had duplicates. The file contained three duplicate rows that have been removed.

```r
DailyActivity <- DailyActivity %>% distinct()
SleepDay <- SleepDay %>% distinct()
```

```
WeightLog <- WeightLog %>% distinct()
```

ANALYZE

Looking at the summary of a few key daily health goals for each of the files should give a good idea of how the participants used their smart devices.

The range of daily steps is wide for this data set. The average daily steps is close to the range of 10000 daily steps.

```
DailyActivity %>%
  summarise(min_TotalSteps = min(TotalSteps, na.rm = TRUE),
            mean_TotalSteps = mean(TotalSteps, na.rm = TRUE),
            max_TotalSteps = max(TotalSteps, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##   min_TotalSteps mean_TotalSteps max_TotalSteps
##            <dbl>           <dbl>          <dbl>
## 1              0           7638.          36019
```

To see what the days of the week looks like for steps for this group, we need to add another column for the days of the week.

```
DailyActivity$Day <- wday(DailyActivity$ActivityDate, label = TRUE, abbr = FALSE)
```

Move it over next to the date column for easier viewing as we analyze the data.

```
DailyActivity <- select(DailyActivity,Id,ActivityDate,Day,everything())
```

Now to see what the steps range looks like over the different days of the week. Tuesday and Saturday seem to have the best averages, while Sunday has the lowest average.

```
DailyActivity %>%
  group_by(Day) %>%
  summarise(min_TotalSteps = min(TotalSteps, na.rm = TRUE),
            mean_TotalSteps = mean(TotalSteps, na.rm = TRUE),
            max_TotalSteps = max(TotalSteps, na.rm = TRUE))
```

```
## # A tibble: 7 x 4
##   Day       min_TotalSteps mean_TotalSteps max_TotalSteps
##   <ord>              <dbl>           <dbl>          <dbl>
## 1 Sunday                 0           6933.          36019
## 2 Monday                 0           7781.          20500
## 3 Tuesday                0           8125.          23186
## 4 Wednesday              0           7559.          23629
## 5 Thursday               0           7406.          21129
## 6 Friday                 0           7448.          21727
## 7 Saturday               0           8153.          29326
```

Looking at the groups activity levels to see how they compare.

Starting with the Sedentary minutes you see some high numbers there. Converting the minutes to hours to get a better idea of how much time the group in this data set spends sedentary. These are some high numbers, more than half the participants days are spent sedentary.

```
DailyActivity %>%
  mutate(SedentaryHours = SedentaryMinutes/60) %>%
  summarise(min_SedentaryHours = min(SedentaryHours, na.rm = TRUE),
```

```
            mean_SedentaryHours = mean(SedentaryHours, na.rm = TRUE),
            max_SedentaryHours = max(SedentaryHours, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##   min_SedentaryHours mean_SedentaryHours max_SedentaryHours
##                <dbl>               <dbl>              <dbl>
## 1                  0                16.5                 24
```

Looking at the activity minutes there isn't a column that totals all the active time columns so we need a column that adds the information together.

```
DailyActivity <- DailyActivity %>%
  mutate(TotalActiveMinutes = VeryActiveMinutes + FairlyActiveMinutes + LightlyActiveMinutes)
```

Since we looked at the sedentary time in hours lets see the same total for active minutes too. Not a lot off active hours in the day for this group, especially compared to the hours spent sedentary.

```
DailyActivity %>%
   mutate(ActiveHours = TotalActiveMinutes/60) %>%
   summarise(min_ActiveHours = min(ActiveHours, na.rm = TRUE),
            mean_ActiveHours = mean(ActiveHours, na.rm = TRUE),
            max_ActiveHours = max(ActiveHours, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##   min_ActiveHours mean_ActiveHours max_ActiveHours
##             <dbl>            <dbl>           <dbl>
## 1               0             3.79             9.2
```

Let's also see what we find by grouping the data by the day of the week. Saturday is the most active day of the week but the group looks consistent in getting more than 3 hours of activity daily.

```
DailyActivity %>%
  group_by(Day) %>%
   mutate(ActiveHours = TotalActiveMinutes/60) %>%
   summarise(min_ActiveHours = min(ActiveHours, na.rm = TRUE),
            mean_ActiveHours = mean(ActiveHours, na.rm = TRUE),
            max_ActiveHours = max(ActiveHours, na.rm = TRUE))
```

```
## # A tibble: 7 x 4
##   Day       min_ActiveHours mean_ActiveHours max_ActiveHours
##   <ord>               <dbl>            <dbl>           <dbl>
## 1 Sunday                  0             3.47            7.4
## 2 Monday                  0             3.82            7.57
## 3 Tuesday                 0             3.91            8
## 4 Wednesday               0             3.73            7.63
## 5 Thursday                0             3.61            8.12
## 6 Friday                  0             3.94            8.65
## 7 Saturday                0             4.07            9.2
```

Instead of typing out the hour conversion each time lets add two columns for the sedentary minutes and the total active minutes to make it easier for when graph the data.

```
DailyActivity <- DailyActivity %>%
  mutate(TotalActiveHours = TotalActiveMinutes/60) %>%
  mutate(SedentaryHours = SedentaryMinutes/60)
```

We should be getting at least 30 minutes of exercise daily, moderate activity or more. We can check the very active minutes for the group and see what the range looks like. Looking at the sum of the range of very active minutes, this group does not meet the recommended daily exercise time frame, coming in just below that.

```
DailyActivity %>%
  summarise(min_VeryActiveMinutes = min(VeryActiveMinutes, na.rm = TRUE),
            mean_VeryActiveMinutes = mean(VeryActiveMinutes, na.rm = TRUE),
            max_VeryActiveMinutes = max(VeryActiveMinutes, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##   min_VeryActiveMinutes mean_VeryActiveMinutes max_VeryActiveMinutes
##                   <dbl>                  <dbl>                 <dbl>
## 1                     0                   21.2                   210
```

Let's see how the averages look for the data by the day of the week. Surprised to see that the Monday and Tuesday have the higher average very active minutes than the weekend days. The bulk of active minutes for this group is spent being lightly active. The group averages just over 3 hours of light activity daily.

```
DailyActivity %>%
  group_by(Day) %>%
  summarise(mean_VeryActiveMinutes = mean(VeryActiveMinutes, na.rm = TRUE),
            mean_FairlyActiveMinutes = mean(FairlyActiveMinutes, na.rm = TRUE),
            mean_LighlyActivenutes = mean(LightlyActiveMinutes, na.rm = TRUE))
```

```
## # A tibble: 7 x 4
##   Day       mean_VeryActiveMinutes mean_FairlyActiveMinutes mean_LighlyActivenu~
##   <ord>                      <dbl>                    <dbl>                <dbl>
## 1 Sunday                      20.0                     14.5                 174.
## 2 Monday                      23.1                     14                   192.
## 3 Tuesday                     23.0                     14.3                 197.
## 4 Wednesday                   20.8                     13.1                 190.
## 5 Thursday                    19.4                     12.0                 185.
## 6 Friday                      20.1                     12.1                 204.
## 7 Saturday                    21.9                     15.2                 207.
```

Let's look at the sleep file next and also add a column for the days of the week as well. We can see if sleep differs by day of the week.

```
SleepDay$Day <- wday(SleepDay$SleepDate, label = TRUE, abbr = FALSE)
```

Moving the new column closer to the date column for easier viewing.

```
SleepDay <- select(SleepDay,Id,SleepDate,Day,everything())
```

Lets do some more calculations on the data and add a column for sleep by the hour to see what we find.

```
SleepDay <- SleepDay %>% mutate(TotalHoursAsleep = TotalMinutesAsleep/60)
```

Overall the average sleep for this group just missed the mark for amount of sleep adults need nightly. The range is 7 to 9 hours of sleep and the groups average is slightly below 7 hours of sleep most of the week. Too much sleep and too little sleep is not good but we need to make sure that we get at minimum 7 hours of sleep at night as adults. Source: Sleep Foundation Article

```
SleepDay %>%
  summarise(min_SleepHrs = min(TotalHoursAsleep),
            mean_SleepHrs = mean(TotalHoursAsleep),
            max_SleepHrs = max(TotalHoursAsleep))
```

```
## # A tibble: 1 x 3
##   min_SleepHrs mean_SleepHrs max_SleepHrs
##          <dbl>         <dbl>        <dbl>
```

```
## 1        0.967           6.99          13.3
```

We can take a brief look at what our range looks like by day of the week. Sunday and Wednesday look like the best averaging days for this group to get sleep.

```
SleepDay %>%
  group_by(Day) %>%
  summarise(min_SleepHrs = min(TotalHoursAsleep),
            mean_SleepHrs = mean(TotalHoursAsleep),
            max_SleepHrs = max(TotalHoursAsleep))
```

```
## # A tibble: 7 x 4
##   Day       min_SleepHrs mean_SleepHrs max_SleepHrs
##   <ord>            <dbl>         <dbl>        <dbl>
## 1 Sunday           0.967          7.55        11.7
## 2 Monday           1.03           6.99        13.3
## 3 Tuesday          1.72           6.74        12.5
## 4 Wednesday        2.53           7.24        11.0
## 5 Thursday         0.983          6.69         9.08
## 6 Friday           1.37           6.76        11.0
## 7 Saturday         1.02           6.98        12.9
```

A brief analysis of the weight file we see it's just 8 participants for the period of time in this data set. Since were not provided much details on the participants we can just see what the range in weight of the participants looks like. To do any further analysis we need to get more information on the participants like the age or whether they are male or female. this type of information would help get more meaningful insights.

```
WeightLog %>% summarise(min_Pounds = min(WeightPounds),
                        mean_Pounds = mean(WeightPounds),
                        max_Pounds = max(WeightPounds))
```

```
## # A tibble: 1 x 3
##   min_Pounds mean_Pounds max_Pounds
##        <dbl>       <dbl>      <dbl>
## 1       116.        159.       294.
```

Let's also add the days of the week column to this file to use when visualizing the data.

```
WeightLog$Day <- wday(WeightLog$Date, label = TRUE, abbr = FALSE)
```

A brief summation of my assessment of the data:

-The group averages over 7000 steps daily, not the recommended daily 10000 steps but close. this is more steps than the average person takes of 3000 to 4000 steps daily. So would say the group is not doing too bad with with their daily steps. Source: Healthy Lifestyle Article.

-The group does have an excessive amount of sedentary hours that need to be brought down much lower. It would help this group to do more moderate activities daily to up there very active minutes and get the recommended daily minutes of exercise in. Source: Washington Post Article.

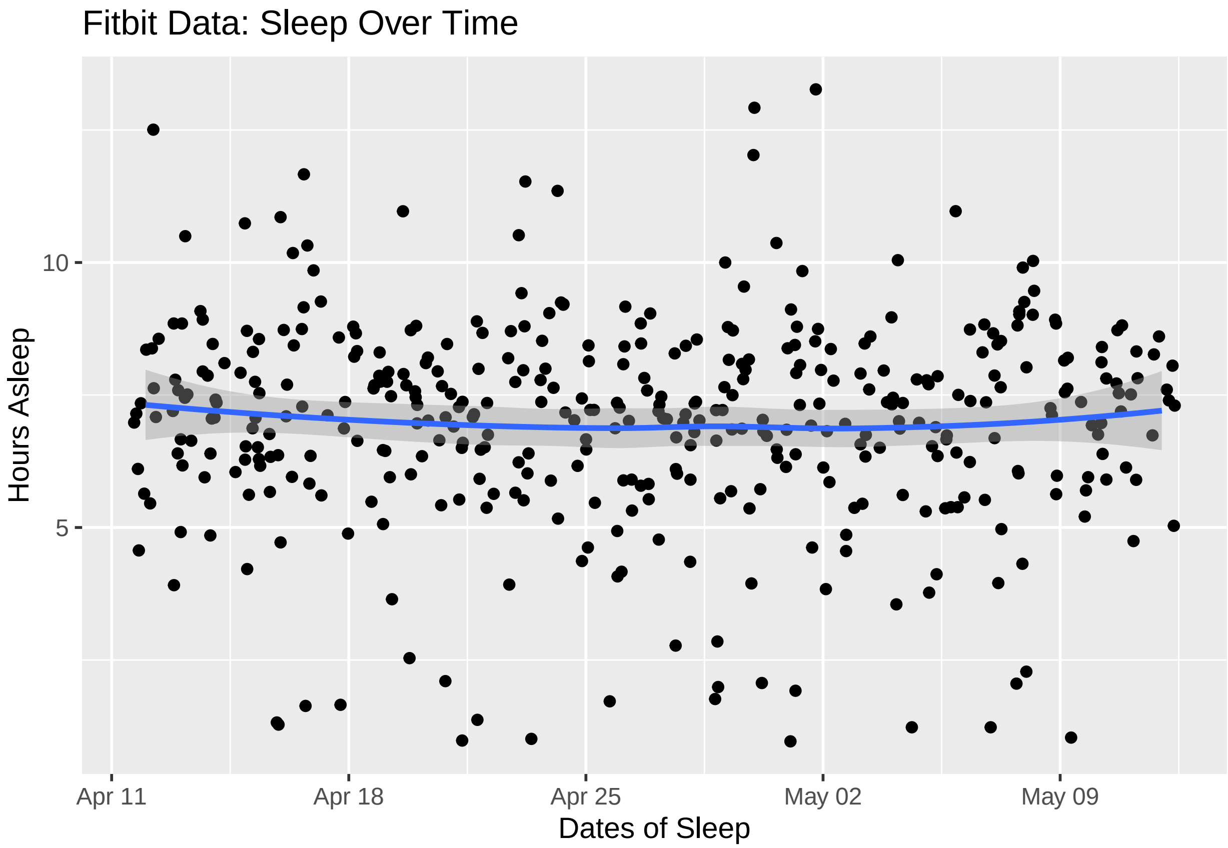
-We also can see from the review of the data that Saturday is the most active day for the group in terms of steps and activity minutes.

SHARE

Using a trend line to see the sleep pattern of the participants during the time frame for this data set, we can see in the graph that the group gets a consistent amount of sleep. They are just not getting the recommended number of hours of sleep regularly.

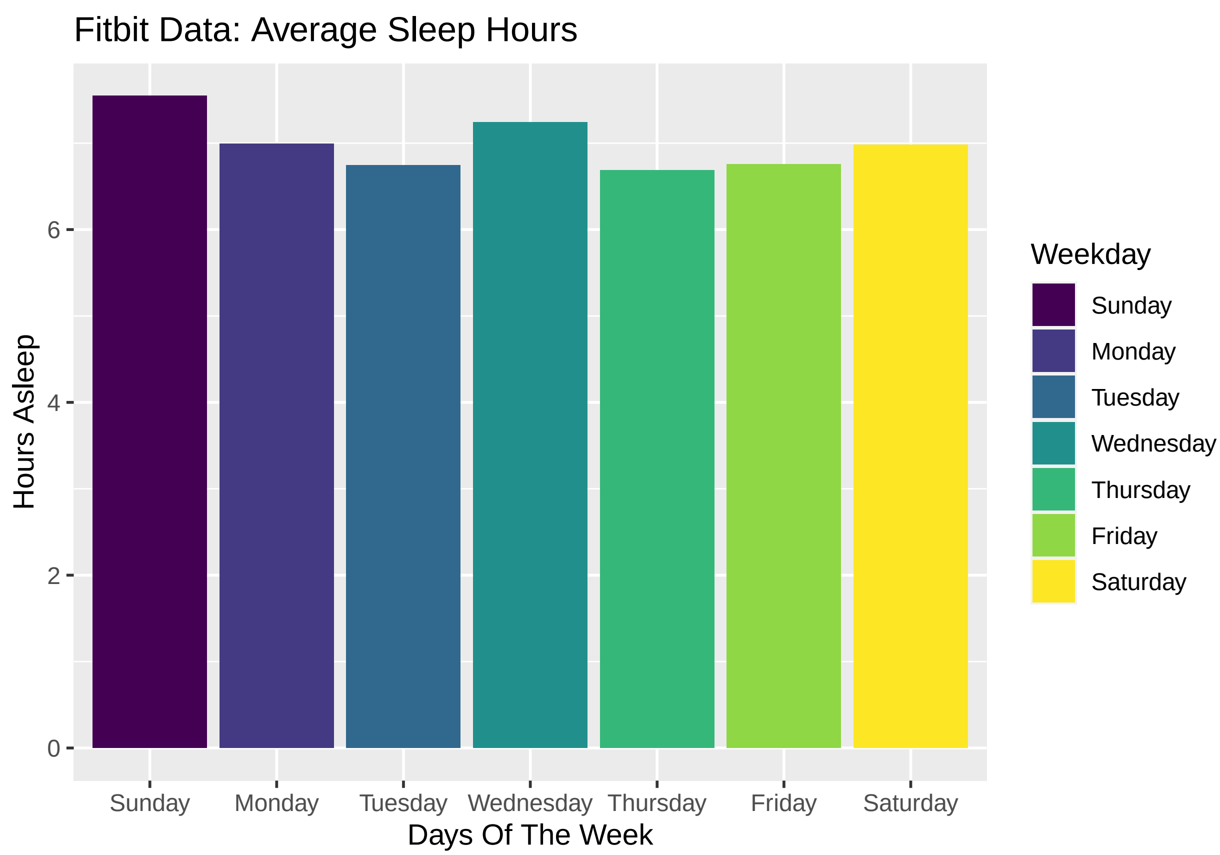
```r
ggplot(data = SleepDay, mapping = aes(x = SleepDate, y = TotalHoursAsleep)) +
  geom_jitter() +
  geom_smooth() +
  labs(title = "Fitbit Data: Sleep Over Time", x = "Dates of Sleep", y = "Hours Asleep")
```

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



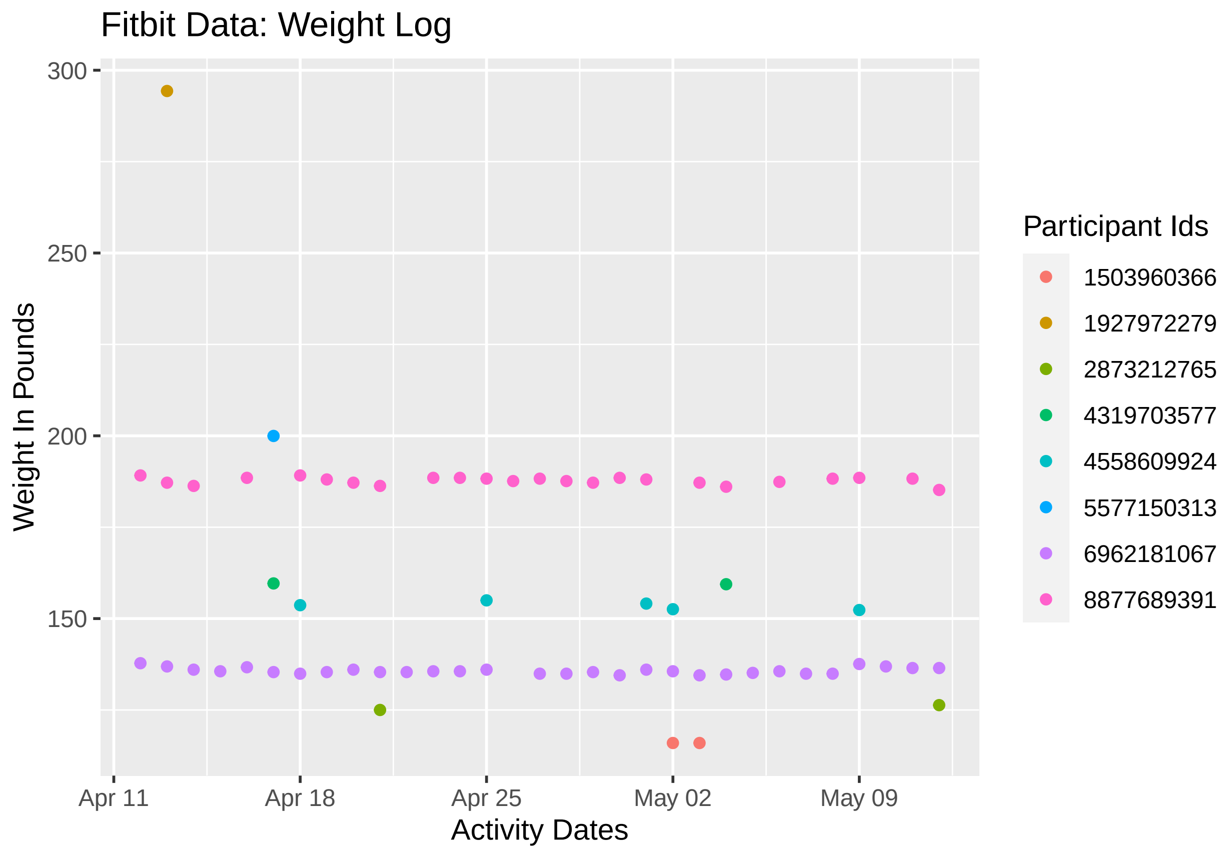
Then looking at the data by day of the week we can see which day of the week averages more sleep. Sunday with the most sleep followed by Wednesday and then Saturday. The weekends seem to be when people catch up on sleep. Only 24 of the 33 participants in the group logged their sleep, probably took off the device at night to charge it or just didn't sleep with the device. A survey of questions to ask specifics about preferences may have helped fill in the gaps here as to why the sleep was not logged as much.

```r
SleepDay %>%
group_by(Day) %>%
  summarise(mean_TotalHoursAsleep = mean(TotalHoursAsleep)) %>%
  ggplot(mapping = aes(x = Day, y = mean_TotalHoursAsleep, fill = Day)) +
  geom_col() +
  labs(title = "Fitbit Data: Average Sleep Hours", x = "Days Of The Week", y = "Hours Asleep", fill = "
```



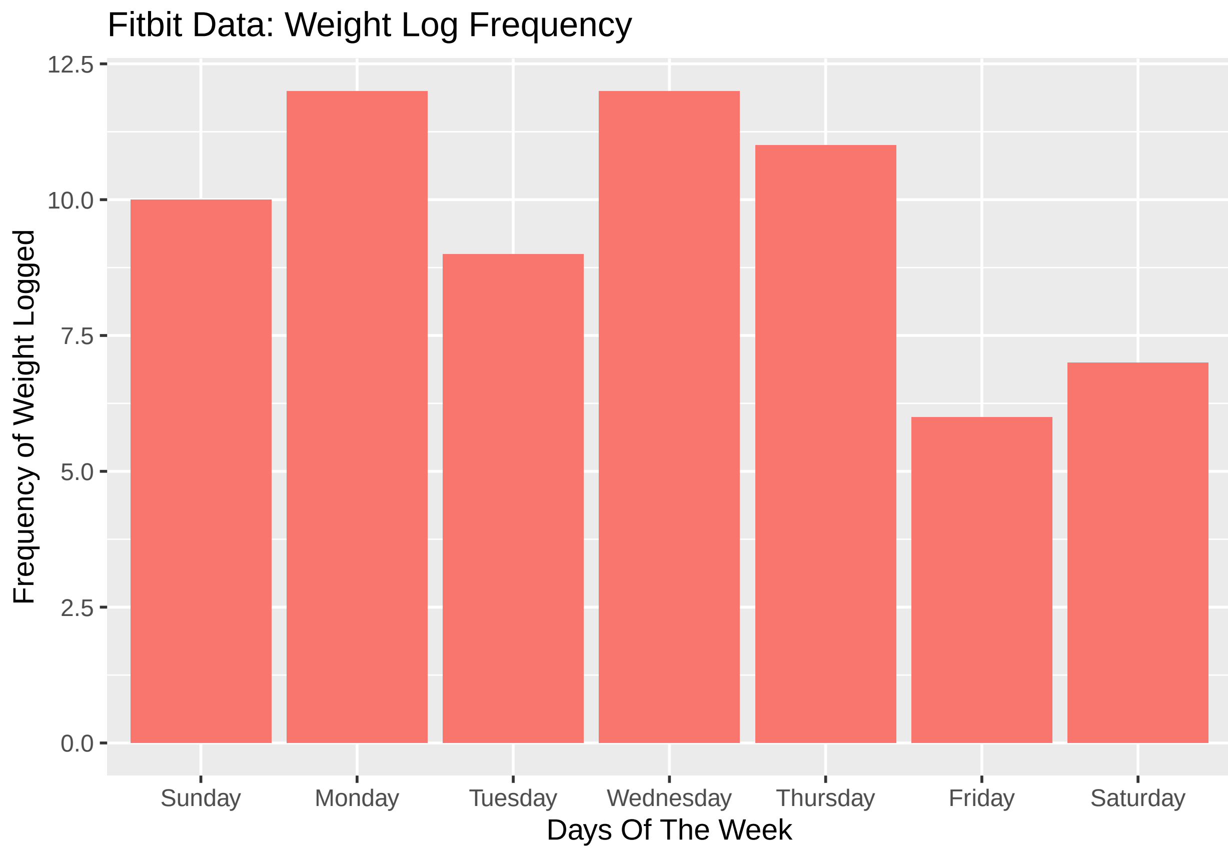
Only two participants regularly logged their weight. Not necessarily a bad thing since recommendations are said that one should weight themselves either daily, weekly or monthly. Once a week is best option of the three since weight fluctuates a lot in a day and if you are trying to stay conscious of your weight only once a month checks may be too long of a wait especially when dieting. Doesn't look like weight loss was a focus for this group, just maintaining weight if anything.Source: Biointelligent Wellness site

```
ggplot(data = WeightLog, mapping = aes(x=Date, y=WeightPounds, color=Id)) +
  geom_point() +
  guides(color = guide_legend(title = "Participant Ids")) +
  labs(title = "Fitbit Data: Weight Log", x = "Activity Dates", y = "Weight In Pounds")
```



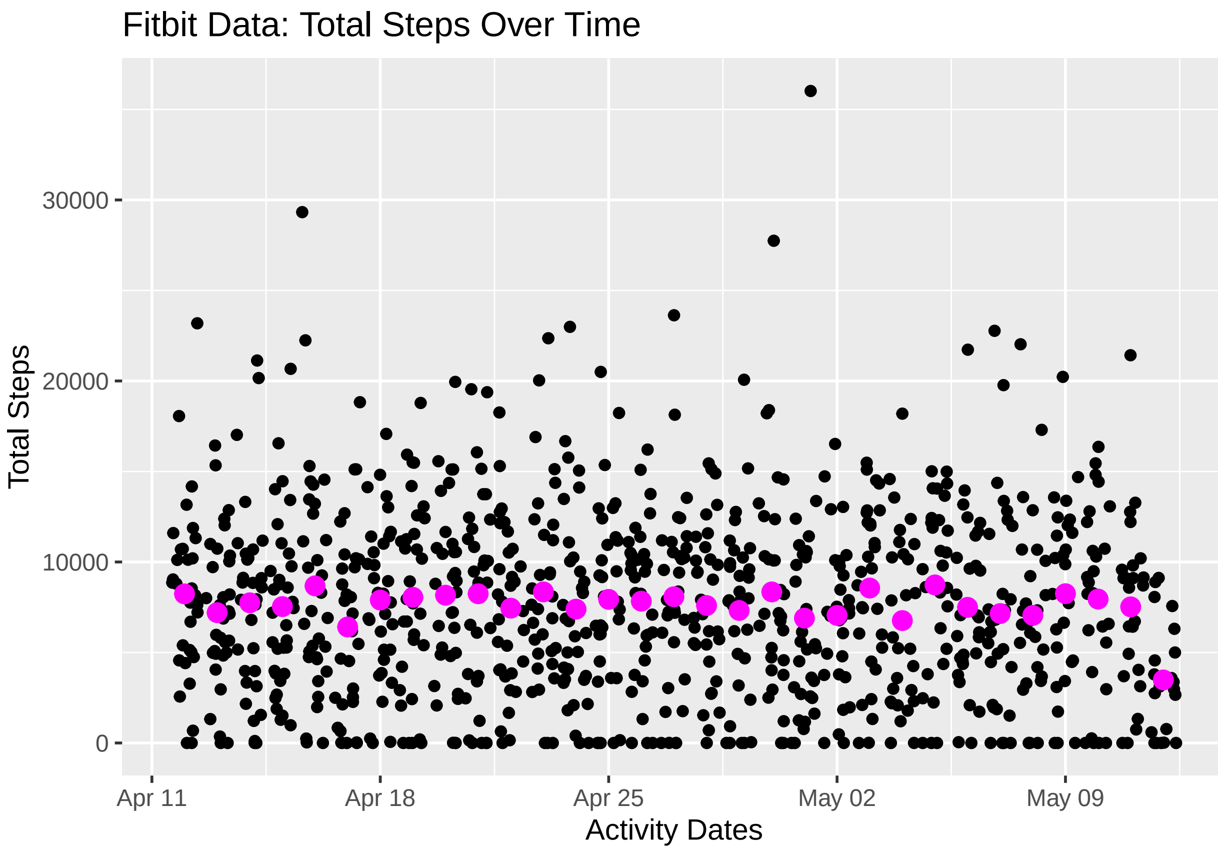
Participants seem to prefer logging their weight on Mondays and Wednesdays the most.

```
WeightLog %>%
  group_by(Day) %>%
  ggplot(mapping = aes(x = Day, fill = "orange")) +
  geom_bar() +
  theme(legend.position = "none") +
  labs(title = "Fitbit Data: Weight Log Frequency", x = "Days Of The Week", y = "Frequency of Weight Log
```



From the DailyActivity file we can see a steady rate of average daily steps for the length of the data set. The group is getting a good amount of daily steps in consistently.

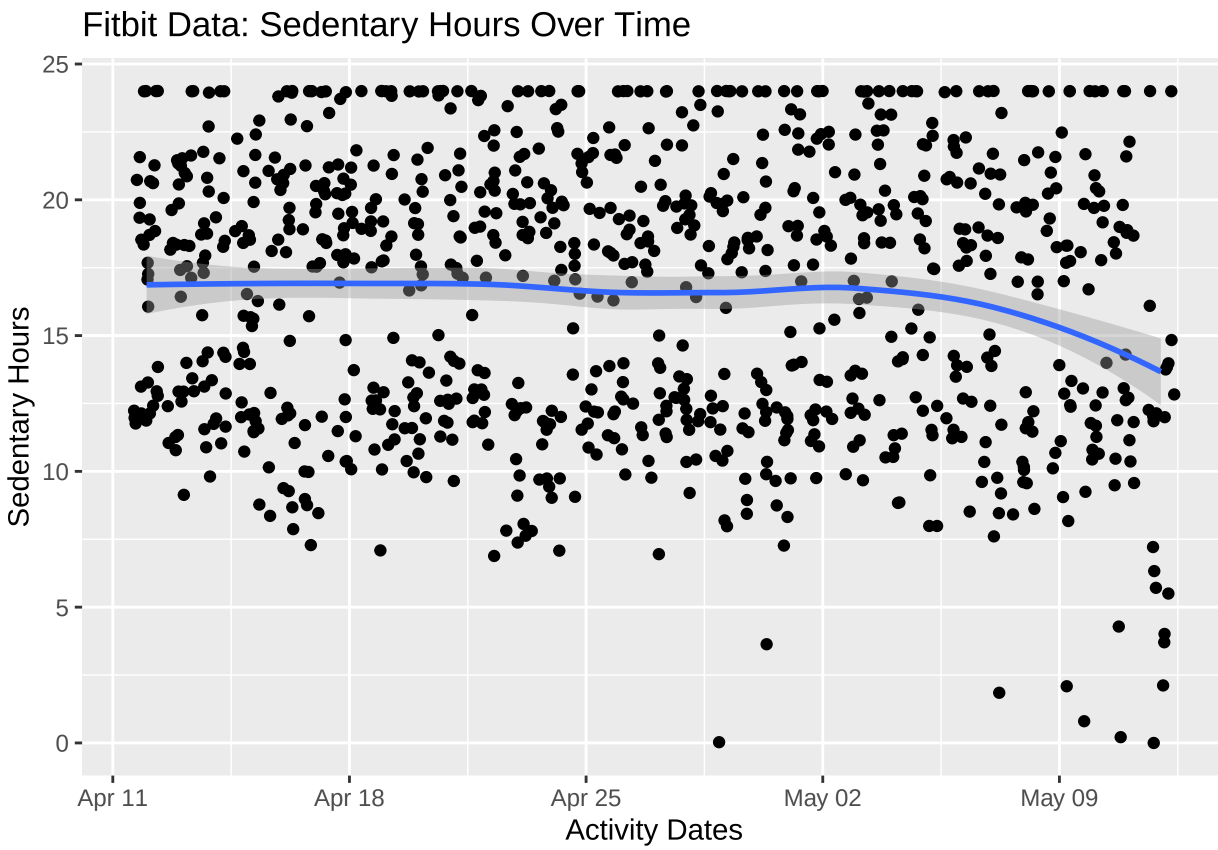
```
ggplot(data = DailyActivity, mapping = aes(ActivityDate,TotalSteps)) +
  geom_jitter() +
  stat_summary(geom = "point", fun = "mean", color= "magenta", size = 3) +
  labs(title = "Fitbit Data: Total Steps Over Time", x = "Activity Dates", y = "Total Steps")
```



The group spends the majority of their day sedentary when you look a the data. The group averages just over 16 hours daily sedentary. If we account for sleep for a portion of this time, sleep does not equal even half this time period. The group is still spending almost 10 hours a day sedentary and that's a lot of time to spend not doing any activity.

```
ggplot(data = DailyActivity, mapping = aes(x = ActivityDate, y = SedentaryHours)) +
  geom_jitter() +
  geom_smooth() +
  labs(title = "Fitbit Data: Sedentary Hours Over Time", x = "Activity Dates", y = "Sedentary Hours")
```

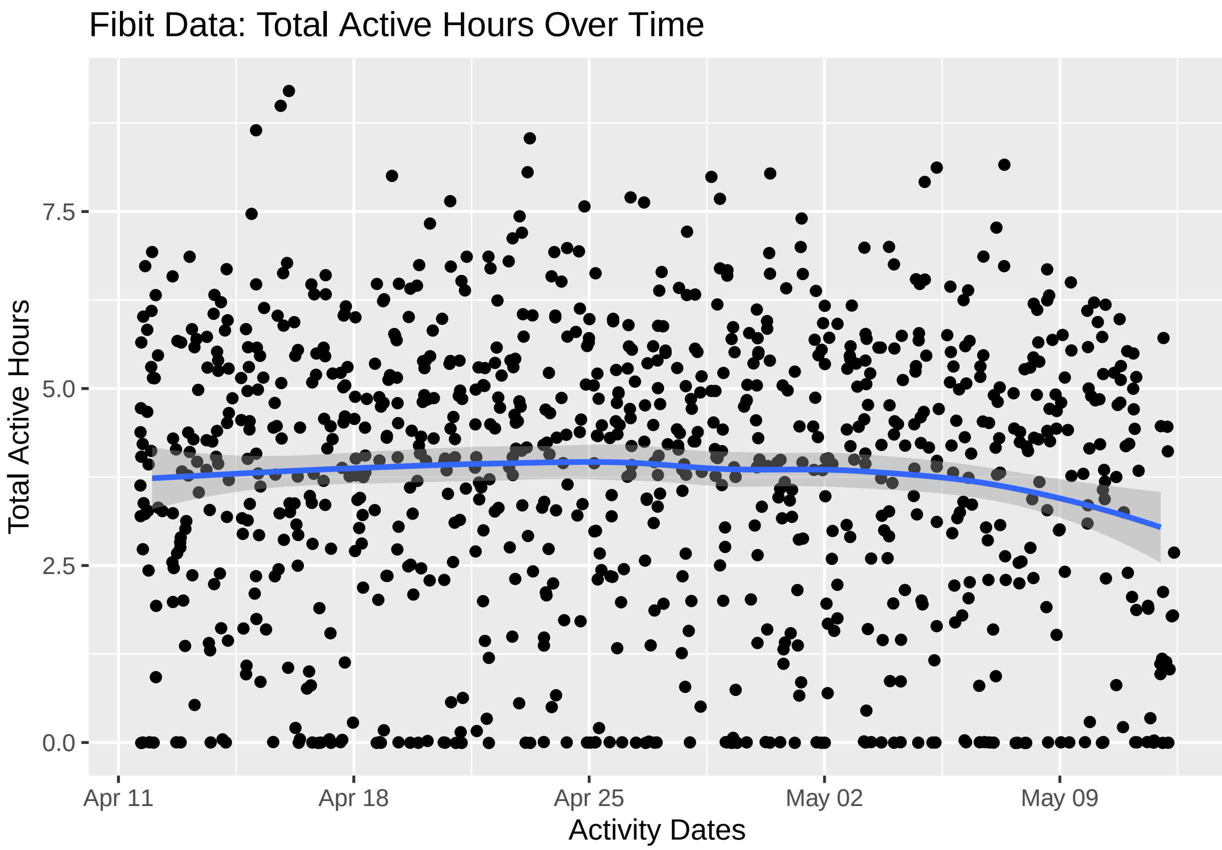
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



When you look at the activity times of the group through out the time frame, the trend line shows a steady pattern. A consistent average rate of time over 3 hours daily.

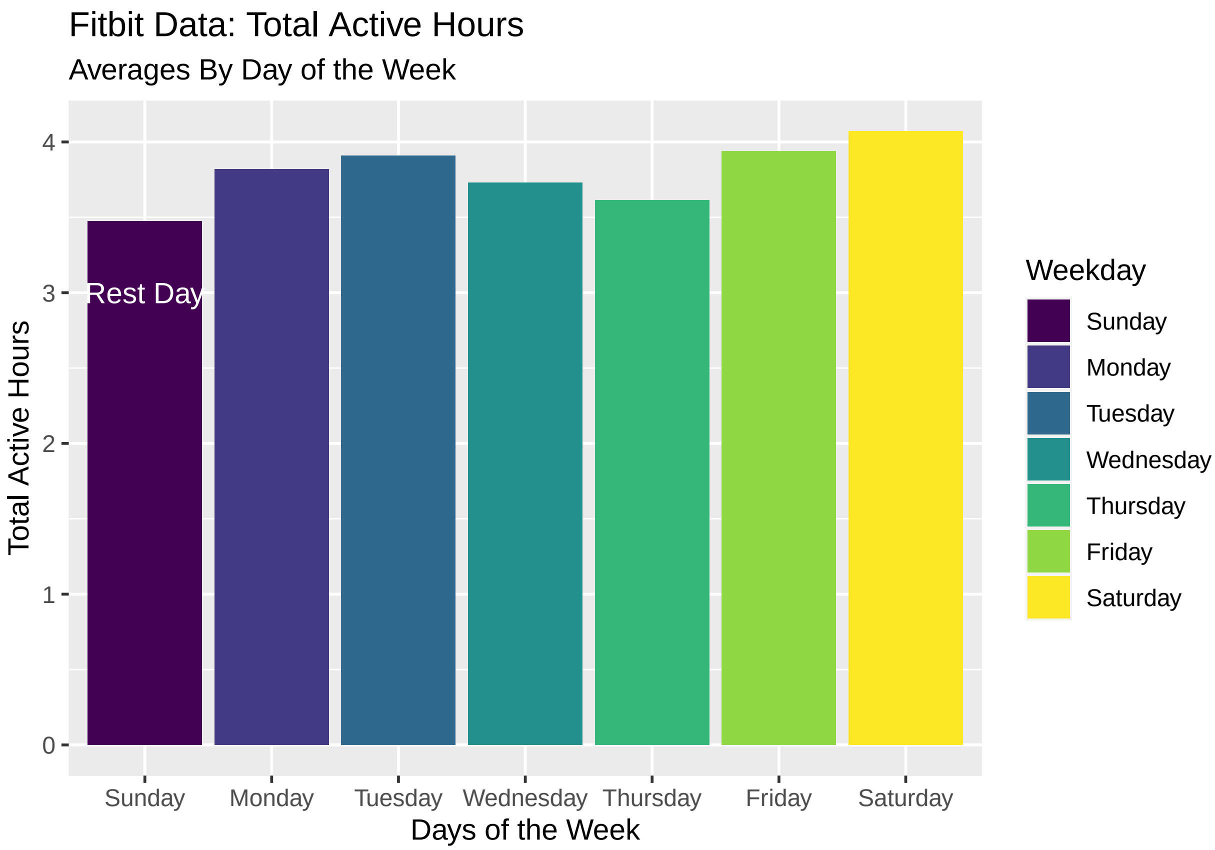
```
ggplot(data = DailyActivity, mapping = aes(x = ActivityDate, y = TotalActiveHours)) +
  geom_jitter() +
  geom_smooth() +
  labs(title = "Fibit Data: Total Active Hours Over Time", x = "Activity Dates", y = "Total Active Hours
```

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



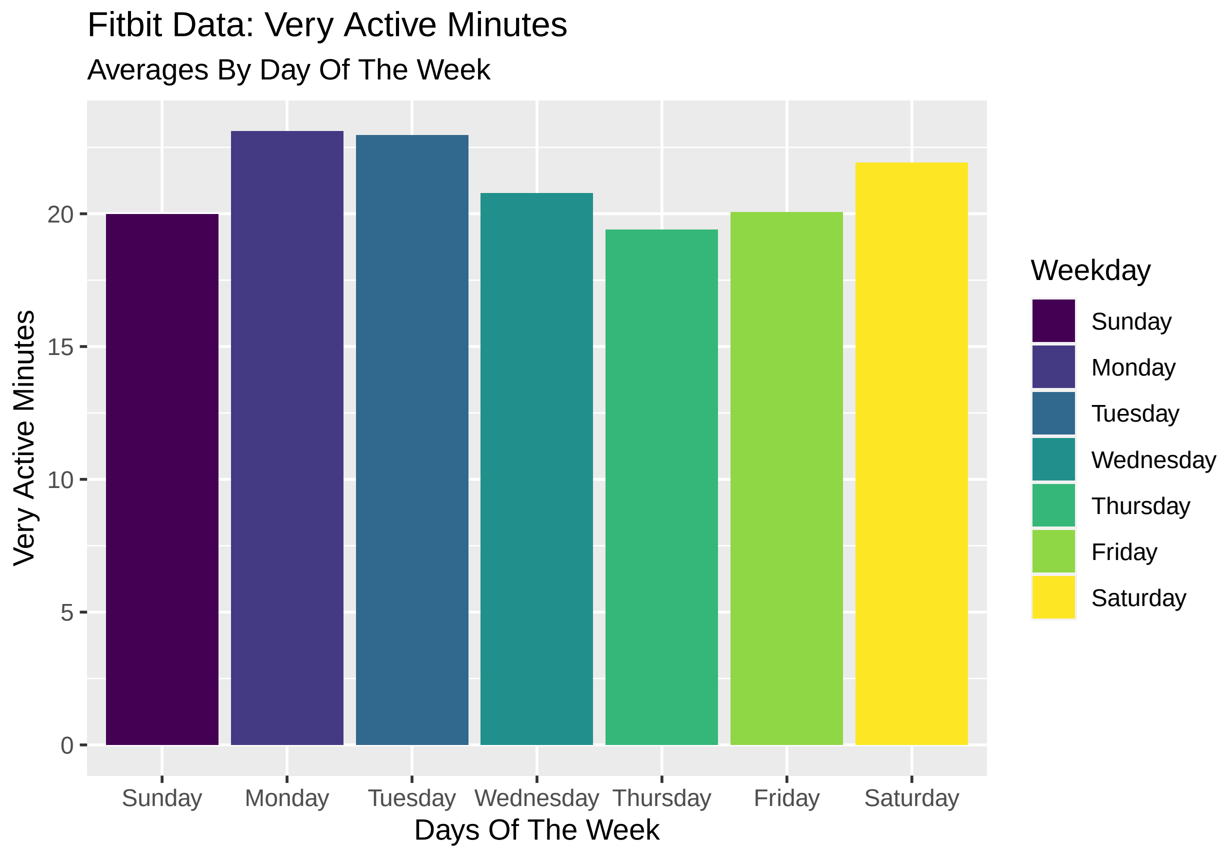
Looking at the breakout of the activity by day of the week. You can see the group spends a total average of 3 hours daily doing some form of activity. Saturday is the only day over 4 hours of activity with Friday pulling up a close second. Sunday seems to be a rest day for the group as that seems to be the day with the lowest amount of active minutes compared to the other days

```
DailyActivity %>%
  group_by(Day) %>%
  summarise(mean_TotalActiveHours = mean(TotalActiveHours)) %>%
  ggplot(mapping = aes(x = Day, y = mean_TotalActiveHours, fill = Day)) +
  geom_col() +
  labs(title = "Fitbit Data: Total Active Hours", subtitle = "Averages By Day of the Week", x = "Days o
  annotate("text", x = 1, y = 3, label = "Rest Day", color = "white")
```



When you just look at the very active minutes for the group, Monday and Tuesday are the days for the group to get the higher impact time of activity. Saturday pull in third for very active minutes.

```
DailyActivity %>%
  group_by(Day) %>%
  summarise(mean_VeryActiveMinutes = mean(VeryActiveMinutes)) %>%
  ggplot(mapping = aes(x = Day, y = mean_VeryActiveMinutes, fill = Day)) +
  geom_col() +
  labs(title = "Fitbit Data: Very Active Minutes", subtitle = "Averages By Day Of The Week", x = "Days (
```



ACT

My findings on the trends in smart device usage based on my analysis is that consumers use none Bellabeat smart devices to mostly track their routines. The participants tracked their steps and exercise mostly and daily calories. From going through the data it looks like mainly how the consumers used the devices was whatever the device automatically tracks daily while they are wearing the devices. Things that need more entry like weight was not so much of a focus.

The business could use this information to prompt healthy daily habits for it's users.

1) Reminding the users to get the proper amount of sleep daily. If they are not meeting the recommended healthy sleep total then prompts on their devices to help them get into a better sleep routine would help I think. Although this may be part of the subscription-based membership we still want to help consumer reach minimum health goals. If the consumer wants to learn more insights into their sleep pattern then they would be encouraged to sign up for the membership.

2) As much as a person needs sleep they need a good amount of moderate exercise daily. So from this data we saw a lot of time spent sedentary meaning the group may have consisted of mostly office workers who spend a good amount of time daily siting. I think prompts would help users remember to get up and move after an hour of being in sedentary mode. 2 Minutes of moderate exercise through out the day would help get the users to the 30 minutes of daily moderate exercise. Even if the user is not on the paid membership some help or encouragement prompts on the device would help.

I also think a long battery life would help users miss less data as they would be able to wear their devices for much longer periods.