BellaBeat Case-Study

Richard Yang

6/2/2022

Project Brief

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. 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 discovered will then help guide marketing strategy for the company. Present your analysis to the Bellabeat executive team along with a high-level recommendations for Bellabeat's marketing strategy.

Ask Phase:

Sršen asks you to analyze smart device usage data in order to gain insights into how consumers use non-Bellabeat smart devices. She then wants you to select one Bellabeat product to apply these insights to in your presentation. Consider these products and questions as you build your analysis:

Products

- **Bellabeat app**: The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products.
- **Leaf**: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
- **Time**: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.
- **Spring**: This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.

Questions

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

Prepare Phase:

Key Objectives:

1. **Store data sets appropriately.**
2. **Identify how the data is organized**
3. **Determine the credibility of the data.**

The data set FitBit Fitness Tracker Data by Mobius was provided. This data set was collected through a survey via Amazon Mechanical Turk from 3/12/2016 to 5/12/2016. Survey data are typically reliable.

Let's determine if the data is ROCCC:

- **Reliable:** This data source was accumulated through a survey posted on Amazon Mechanical Turk. Questions in the survey could be a concern that may cause bias, but otherwise data through a survey are reliable.
- **Original:** According to the description of this data set. This was done by consent of 30 Fitbit users through a survey as previously mentioned.This data set would be considered 3rd party data accumulated through Amazon Mechanical Turk. After cross-examining this data set with the cited source, this data set is not original. It is however, a subset of the data set.
- **Comprehensive:** This data set is comprehensive. It has some limitations in the overall data set, but there is sufficient data that will help with analyzing smart device usage. More descriptive, qualitative data would have been helpful for a deeper analysis.
- **Current:** The data is not current, the data as previously mention is from 2016.
- **Cited:** The data set is cited with reference to individuals involved in this survey research. The website that hosts this original dataset is on Zenodo.org, which seems to be an open-sourced research data repository.

When observing the data set initially, we can see that the data sets are a mix of both long and wide data. There are 18 total data sets, but a lot of the data are duplicates of each other. After thoroughly investigating the data set, I have chosen 2 of the data sets to focus on. Some columns could have used metadata to explain more about measurements, but I don't believe the lack in metadata will deter the analysis. My focus will be on daily activity usages rather than the hourly performances from the users.

We will consider using the following libraries in R:

```r
# install libraries
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
#load libraries for use.
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(skimr)
```

Once the packages are loaded, I want to set up the data to check its integrity. We'll start allocating the data sets into dataframes. Then, we'll use functions skim_without_charts and summary on the dataframes to check integrity of the data.

```
# load datasets in to dataframes for analysis.
daily_activity <- 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.
```

```
daily_sleep <-  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.
```

```
# Check summary of data for integrity.
skim_without_charts(daily_activity)
```

Table 1: Data summary

Name	daily_activity
Number of rows	940
Number of columns	15
Column type frequency:	
character	1
numeric	14
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDate	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480000e+00
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440000e+03
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03

```
skim_without_charts(daily_sleep)
```

Table 4: Data summary

Name	daily_sleep
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
SleepDay	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.000979e+09	2.060360e+09	1503960366	3977333714	4702921684	6962181067	8792009665
TotalSleepRecords	0	1	1.120000e+00	3.500000e-01	1	1	1	1	3
TotalMinutesAsleep	0	1	4.194700e+02	1.183400e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.271000e+02	61	403	463	526	961

```
## Summary of all columns in the data frame.
summary(daily_activity)
```

```
##        Id              ActivityDate         TotalSteps     TotalDistance
##  Min.   :1.504e+09   Length:940         Min.   :    0   Min.   : 0.000  
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 3790   1st Qu.: 2.620  
##  Median :4.445e+09   Mode  :character   Median : 7406   Median : 5.245  
##  Mean   :4.855e+09                      Mean   : 7638   Mean   : 5.490  
##  3rd Qu.:6.962e+09                      3rd Qu.:10727   3rd Qu.: 7.713  
##  Max.   :8.878e+09                      Max.   :36019   Max.   :28.030  
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000           Min.   : 0.000    
##  1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000    
##  Median : 5.245   Median :0.0000           Median : 0.210    
##  Mean   : 5.475   Mean   :0.1082           Mean   : 1.503    
##  3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053    
##  Max.   :28.030   Max.   :4.9421           Max.   :21.920    
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000       
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000       
##  Median :0.2400           Median : 3.365      Median :0.000000       
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606       
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000       
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000       
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0  
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8  
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5  
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2  
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5  
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0  
##     Calories   
##  Min.   :   0  
##  1st Qu.:1828  
##  Median :2134  
##  Mean   :2304  
##  3rd Qu.:2793  
##  Max.   :4900
```

```
summary(daily_sleep)
```

```
##        Id              SleepDay         TotalSleepRecords TotalMinutesAsleep
##  Min.   :1.504e+09   Length:413         Min.   :1.000     Min.   : 58.0     
##  1st Qu.:3.977e+09   Class :character   1st Qu.:1.000     1st Qu.:361.0     
##  Median :4.703e+09   Mode  :character   Median :1.000     Median :433.0     
##  Mean   :5.001e+09                      Mean   :1.119     Mean   :419.5     
##  3rd Qu.:6.962e+09                      3rd Qu.:1.000     3rd Qu.:490.0     
##  Max.   :8.792e+09                      Max.   :3.000     Max.   :796.0     
##  TotalTimeInBed 
##  Min.   : 61.0  
##  1st Qu.:403.0  
##  Median :463.0  
##  Mean   :458.6  
##  3rd Qu.:526.0  
##  Max.   :961.0
```

We observe see that in both data sets there are no missing values per column. The dataset also comes from Zenodo.org, a research data repository. This data was gathered through a survey from volunteers who were willing to share their Fitbit data with the researchers. Considering that this is relevant Fitbit data gather through a survey, we can use this data as smart device data to find insights to smart device usage. The data set seems valid and we can proceed with organizing the data set so it can be usable for analysis.

Process Phase:

Key Tasks

1. **Check the data for errors.**
2. **Choose your tools.**
3. **Transform the data so you can work with it effectively**
4. **Document the cleaning process.**

Let's begin by verifying the unique IDs in this data set, and see if it aligns with the description of this data set.

```
# Verifying unique IDs in the data set to see if it aligns with description of data.
n_distinct(daily_activity$Id)
```

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

```
n_distinct(daily_sleep$Id)
```

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

According to the data description, we were suppose to have 30 individuals that were surveyed for their Fit bit data. We can see that our daily_activity data set has 33, 3 more than describe. The daily_sleep data set only has 24 IDs, but I believe we can still make an analysis with the existing data.

We'll continue our cleaning/process phase using R. I believe using R will allow me to manipulate and clean my data efficiently. Afterwards, I can transition easily into visualizing.

I want to create a more complete data set from my two dataframes so it will be more efficient to work with. I will look to merge the data set based on IDs so I can work with all my data within one dataframe.

We'll create another dataframe to use as well. This dataframe will rotate some columns to long data. This long data will help with making great insights. The following codes will provide details:

```
## Convert daily_activity dates in data frame to Date Objects.
daily_activity$ActivityDate <- as.POSIXct(daily_activity$ActivityDate, format = "%m/%d/%Y")

## Create and format new date column for daily_activity data set.
daily_activity$Date <- format(daily_activity$ActivityDate, format = "%m/%d/%Y")

## Convert daily_sleep dates in data frame to Date Objects.
daily_sleep$SleepDay <- as.POSIXct(daily_sleep$SleepDay, format = "%m/%d/%Y")

## Create and format new date column for daily_sleep data set.
daily_sleep$Date <- format(daily_sleep$SleepDay, format = "%m/%d/%Y")
```

```
# Merge daily_activity and daily sleep data frames, then piped to...
# Drop any rows with missing values, then piped to...
# Select columns to NOT include int he final data set.
daily_data <- merge(x=daily_activity, y=daily_sleep, by = c('Id', 'Date')) %>%
  drop_na() %>%
  select(-ActivityDate, -TrackerDistance, -LoggedActivitiesDistance, -SleepDay)

# Convert Character Date column to Data Object
daily_data$Date <- as.POSIXct(daily_data$Date, format ="%m/%d/%Y")

# Create DaysOfWeek column for analysis
```

```
daily_data$DaysOfWeek <- weekdays(daily_data$Date)

# Convert Data Object to Character Date column
daily_data$Date <- format(daily_data$Date, format = "%m/%d/%Y")

# Preview sample of data set
head(daily_data)
```

```
##           Id       Date TotalSteps TotalDistance VeryActiveDistance
## 1 1503960366 04/12/2016      13162          8.50               1.88
## 2 1503960366 04/13/2016      10735          6.97               1.57
## 3 1503960366 04/15/2016       9762          6.28               2.14
## 4 1503960366 04/16/2016      12669          8.16               2.71
## 5 1503960366 04/17/2016       9705          6.48               3.19
## 6 1503960366 04/19/2016      15506          9.88               3.53
##   ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
## 1                     0.55                6.06                       0
## 2                     0.69                4.71                       0
## 3                     1.26                2.83                       0
## 4                     0.41                5.04                       0
## 5                     0.78                2.51                       0
## 6                     1.32                5.03                       0
##   VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
## 1                25                  13                  328              728
## 2                21                  19                  217              776
## 3                29                  34                  209              726
## 4                36                  10                  221              773
## 5                38                  20                  164              539
## 6                50                  31                  264              775
##   Calories TotalSleepRecords TotalMinutesAsleep TotalTimeInBed DaysOfWeek
## 1     1985                 1                327            346    Tuesday
## 2     1797                 2                384            407  Wednesday
## 3     1745                 1                412            442     Friday
## 4     1863                 2                340            367   Saturday
## 5     1728                 1                700            712     Sunday
## 6     2035                 1                304            320    Tuesday
```

```
# Observe unique ID to verify data integrity.
n_distinct(daily_data$Id)
```

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

```
# Create a data frame that converts some columns to long data.
# Columns to convert are All the "ActivityMinutes" columns.
activity_minutes <- daily_data %>%
  select(Id, Date,DaysOfWeek,TotalSteps, VeryActiveMinutes,
         FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes) %>%
  pivot_longer(cols = ends_with("Minutes"),
               names_to = "Activity", values_to = "ActivityLevels")

head(activity_minutes)
```

```
## # A tibble: 6 x 6
##           Id Date       DaysOfWeek TotalSteps Activity             ActivityLevels
##        <dbl> <chr>      <chr>           <dbl> <chr>                         <dbl>
## 1 1503960366 04/12/2016 Tuesday         13162 VeryActiveMinutes              25
## 2 1503960366 04/12/2016 Tuesday         13162 FairlyActiveMinutes            13
## 3 1503960366 04/12/2016 Tuesday         13162 LightlyActiveMinut~           328
## 4 1503960366 04/12/2016 Tuesday         13162 SedentaryMinutes              728
## 5 1503960366 04/13/2016 Wednesday       10735 VeryActiveMinutes              21
## 6 1503960366 04/13/2016 Wednesday       10735 FairlyActiveMinutes            19
```

In the codes above, we recreated the columns with dates into an "Date Object" after realizing that possibly some of the dates may be formatted as text. Once we create the date objects, we could merge the data sets as an INNER JOIN based on "Id" and "Date" column from both data sets. Once we had the data sets merged, we also used the "Weekend()" function to create a column that tells us what dates are what days of the week.

We verified the data merged correctly by previewing a sample of it, as well as, validating that the # of Unique IDs is the same as before. We see that the unique ID count is 24, which is the same as the unique ID from the daily_sleep data set.

The activity_minutes data frame was created using the full data set we just created. All we did here was pivot some columns, so we can get insights from shifting the data.

We now have two very good data frames to work with. It may take some more manipulation of these two data frames to get some insights, but we have a good data set foundation now. We will only be analyzing data from individuals who have contributed survey data to both the daily_activities and daily_sleep.

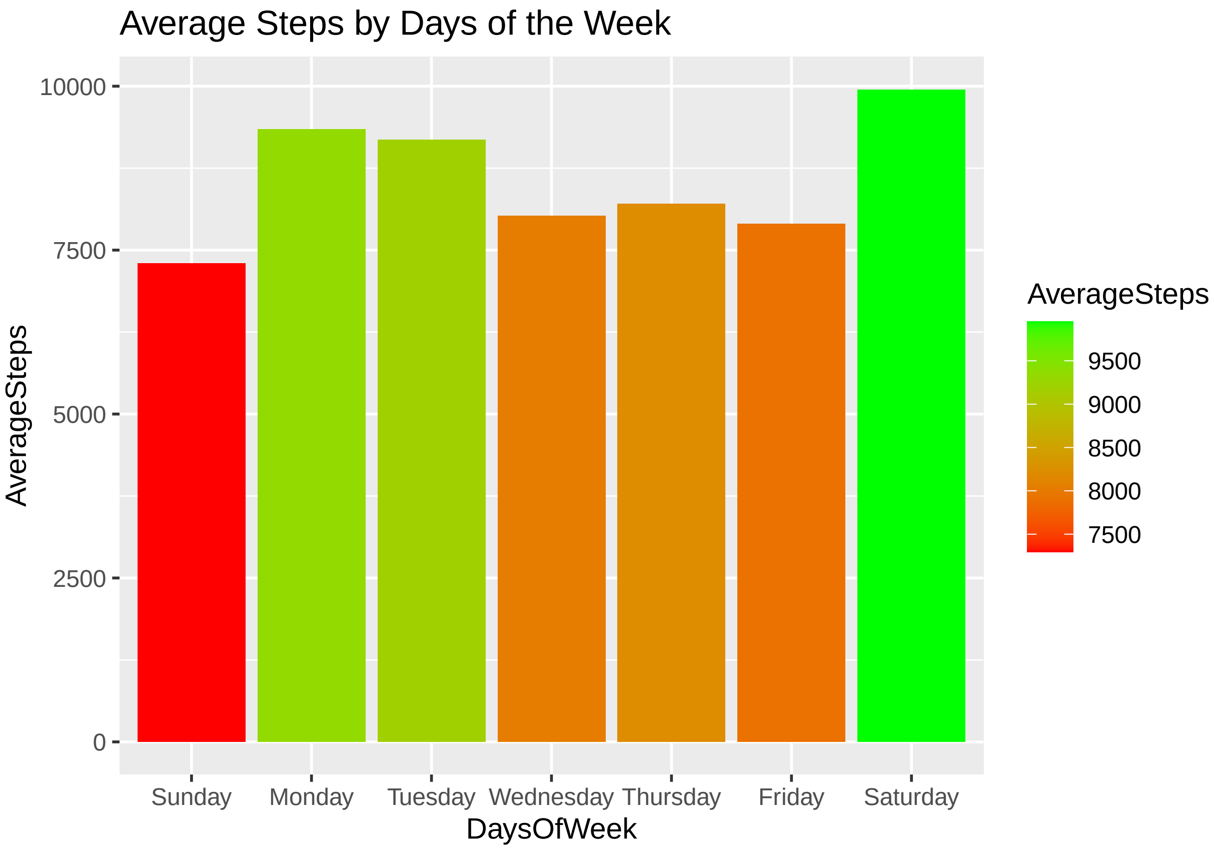
Analyze Phase:

Key Tasks

1. **Aggregate your data so it's useful and accessible**
2. **Organize and format your data**
3. **Perform calculations**
4. **How will these insights help answer your business questions?**

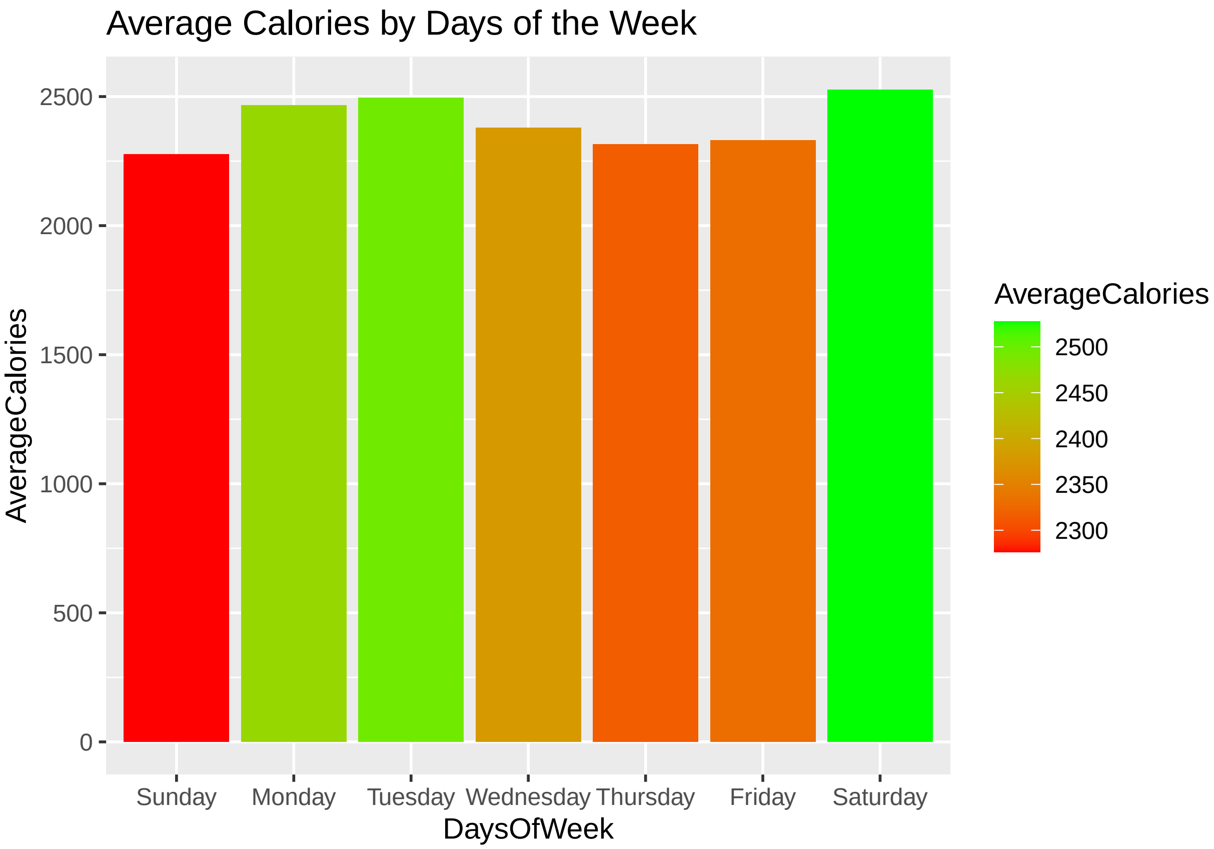
In the previous phase, we were able to aggregate, organize, and format the data set in to a usable forms. We'll begin exploring the data set to see what insights we can find in regards to smart device usage among these 24 candidates.

```r
# Create a average steps by day data frame to graph
avg_dist_week <- daily_data %>%
  group_by(DaysOfWeek) %>%
  summarize(mean(TotalSteps)) %>%
  rename(AverageSteps = "mean(TotalSteps)")

# Reorganize data frame by days of the week.
avg_dist_week$DaysOfWeek <-  ordered(avg_dist_week$DaysOfWeek,
           levels=c("Sunday", "Monday", "Tuesday", "Wednesday",
                    "Thursday", "Friday", "Saturday"))

# Plot Avg. Steps Vs DayOfWeek.
# Add color to see which days were highest/lowest
ggplot(data=avg_dist_week) +
  geom_col(mapping = aes(x=DaysOfWeek, y=AverageSteps, fill=AverageSteps)) +
  scale_fill_gradient(low='red', high='Green') +
  labs(title = "Average Steps by Days of the Week")
```



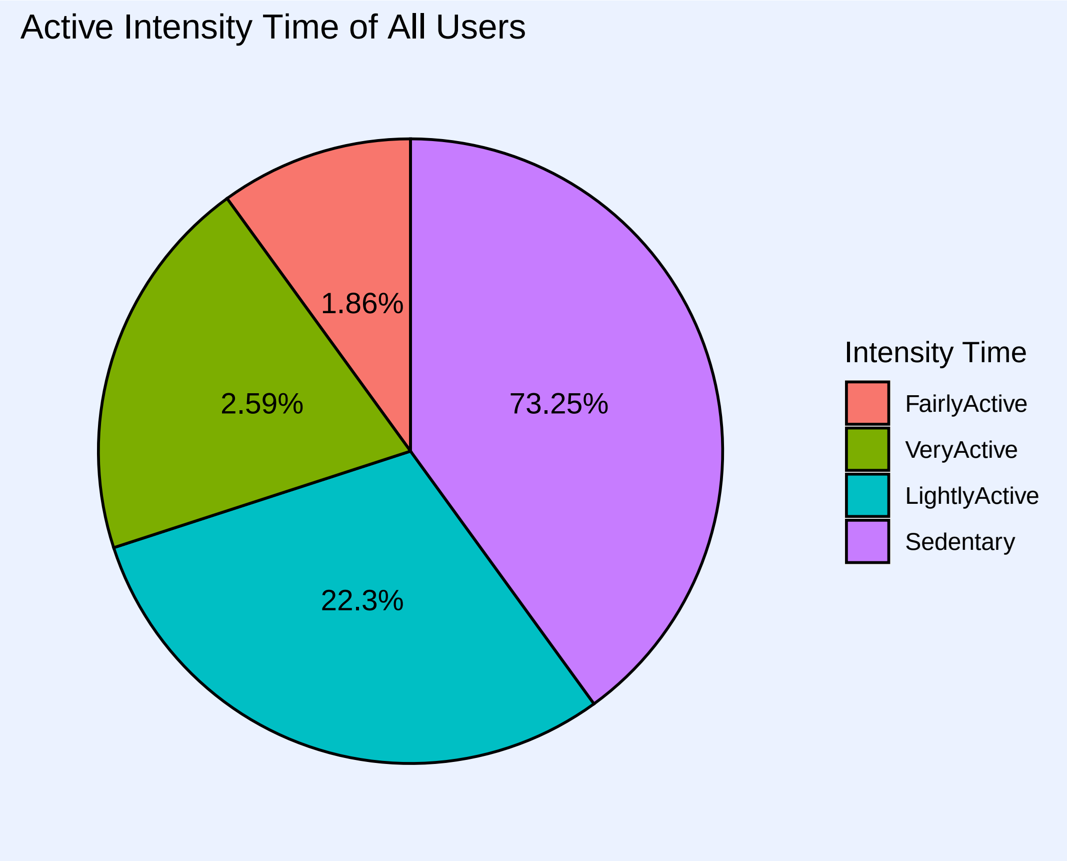
(*Figure 1*)

```
# Create Average calories per day
avg_calorie_week <- daily_data %>%
  group_by(DaysOfWeek) %>%
  summarize(mean(Calories)) %>%
  rename(AverageCalories = "mean(Calories)")

# Reorganize the order of days of the week
avg_calorie_week$DaysOfWeek <-  ordered(avg_dist_week$DaysOfWeek,
                                     levels=c("Sunday", "Monday", "Tuesday",
                                              "Wednesday", "Thursday",
                                              "Friday", "Saturday"))
# Create graph for calories per day
ggplot(data=avg_calorie_week) +
  geom_col(mapping = aes(x=DaysOfWeek, y=AverageCalories,
                      fill=AverageCalories)) +
  scale_fill_gradient(low='red', high='Green') +
  labs(title = "Average Calories by Days of the Week")
```



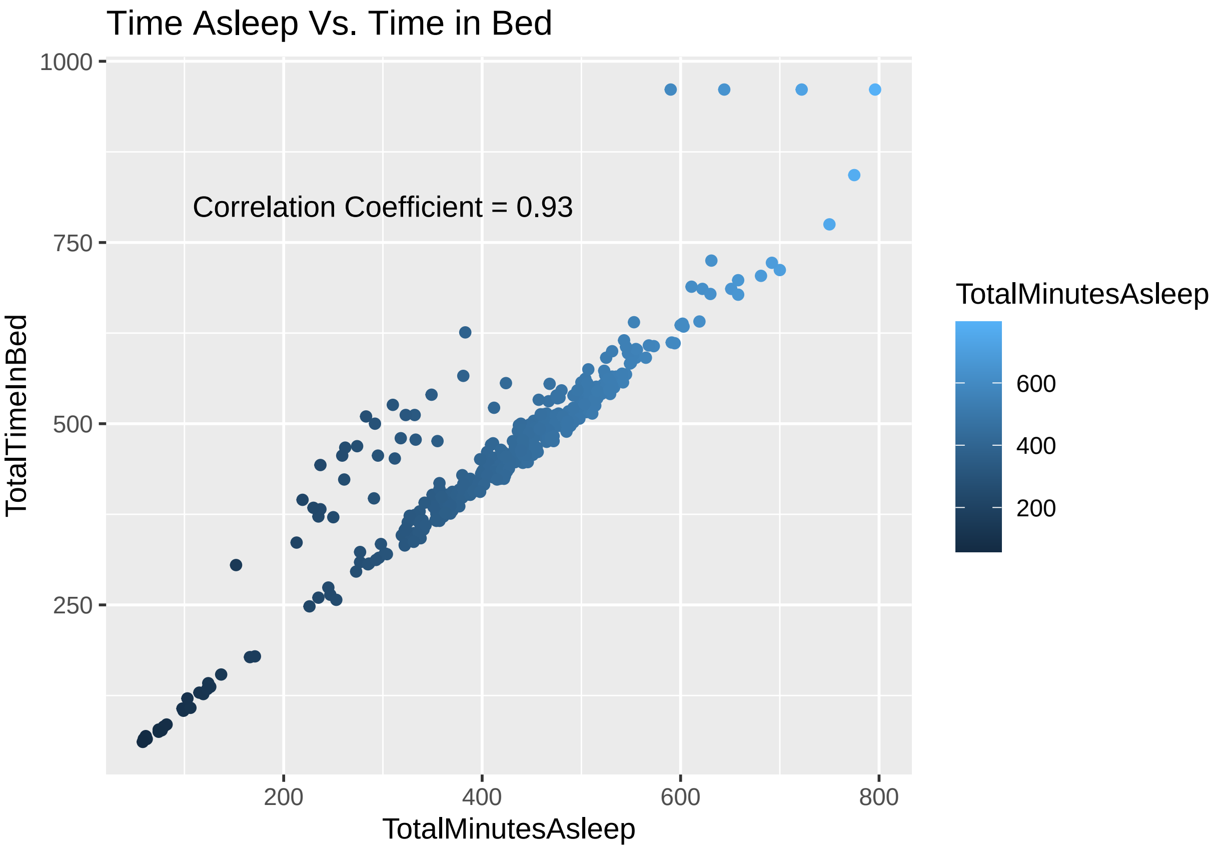
(*Figure 2*)

```
# Create data frame for activity
avg_activity_lvls <- activity_minutes %>%
  group_by(Activity) %>%
  summarize(AvgActivityLevels = mean(ActivityLevels))%>%
  rename(IntensityTime = Activity)

#round the column AvgActivityLevels to 2 decimal points
avg_activity_lvls$AvgActivityLevels <- round(avg_activity_lvls$AvgActivityLevels, digits = 2)

#Derive percentage values from column AvgActivityLevels
avg_activity_lvls$Percent <- round(avg_activity_lvls$AvgActivityLevels / 972.25 * 100, 2)

#Append "%" symbol to percentage values in Percent column
avg_activity_lvls$Percent <- paste0(as.matrix(avg_activity_lvls$Percent), '%')

# Reorganize the order of days of the week
avg_activity_lvls$IntensityTime <-  ordered(avg_activity_lvls$IntensityTime,
                                      levels=c("SedentaryMinutes", "LightlyActiveMinutes", "FairlyActive

# Alternative Graph for these values
# ggplot(data=avg_activity_lvls, aes(x=IntensityTime, y= AvgActivityLevels,
#                                     fill=AvgActivityLevels)) +
#   geom_col() +
#   geom_text(aes(x=IntensityTime, y= AvgActivityLevels,
#                 label=AvgActivityLevels), vjust = -0.5), size = 3) +
#   scale_fill_gradient(low='red', high = 'green') +
#   scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
#   labs(title = "IntensityTime Vs. Average Activity Levels")

ggplot(avg_activity_lvls, aes(x = "", y = as.factor(Percent), fill = as.factor(Percent))) +
  geom_col(color = "black") +
  geom_text(aes(label = Percent),
            position = position_stack(vjust = 0.5)) +
  coord_polar(theta = "y") +
  scale_fill_discrete(labels = c("FairlyActive", "VeryActive" ,"LightlyActive" ,"Sedentary")) +
  theme(axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        panel.background = element_rect(fill = "#ebf2ff"),
        plot.background = element_rect(fill = "#ebf2ff"),
        legend.background = element_rect(fill = "#ebf2ff")) +
  guides(fill = guide_legend(title = "Intensity Time")) +
  labs(title = "Active Intensity Time of All Users")
```



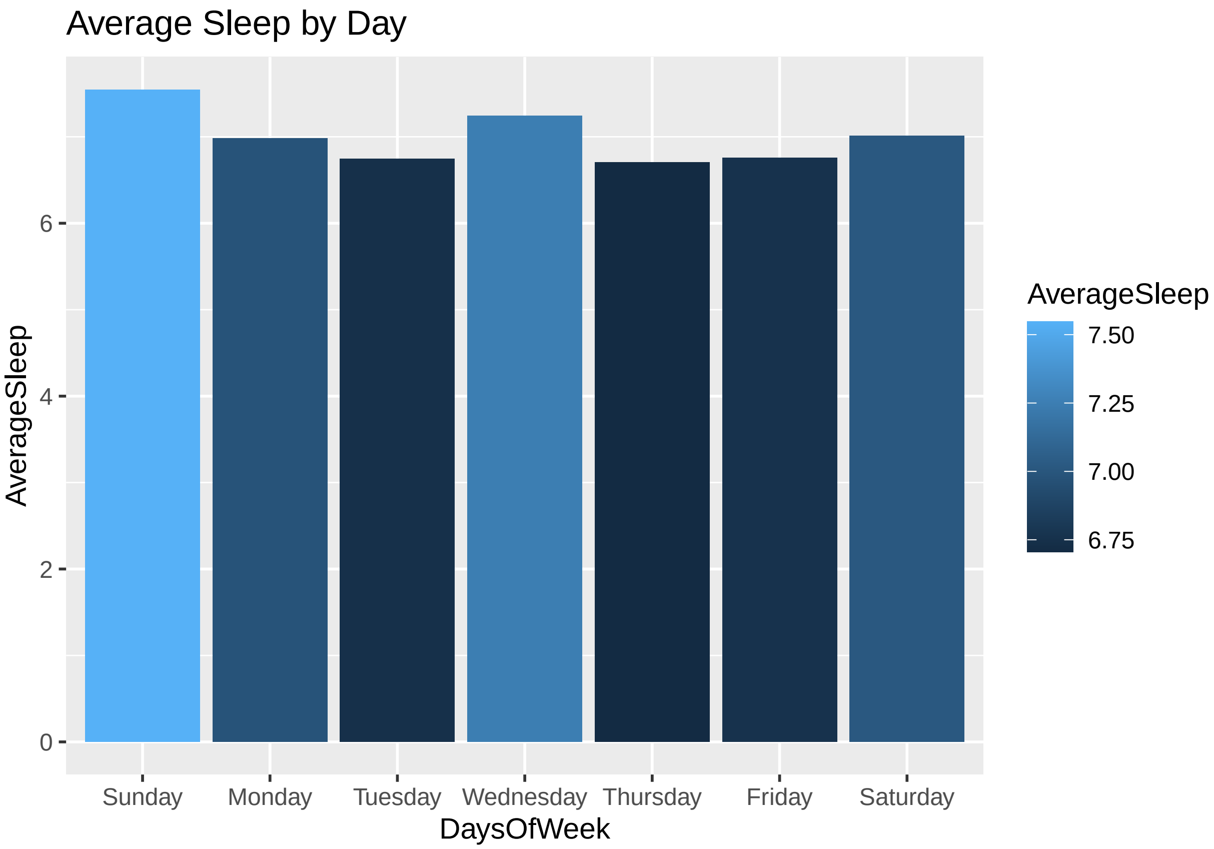
(*Figure 3*)

```
# Create correlation graph of time in bed vs. sleep time
ggplot(data=daily_data) +
  geom_point(mapping = aes(x=TotalMinutesAsleep, y=TotalTimeInBed,
                           color=TotalMinutesAsleep)) +
  annotate('text', x =300,y=800, label= "Correlation Coefficient = 0.93")+
  labs(title = "Time Asleep Vs. Time in Bed")
```



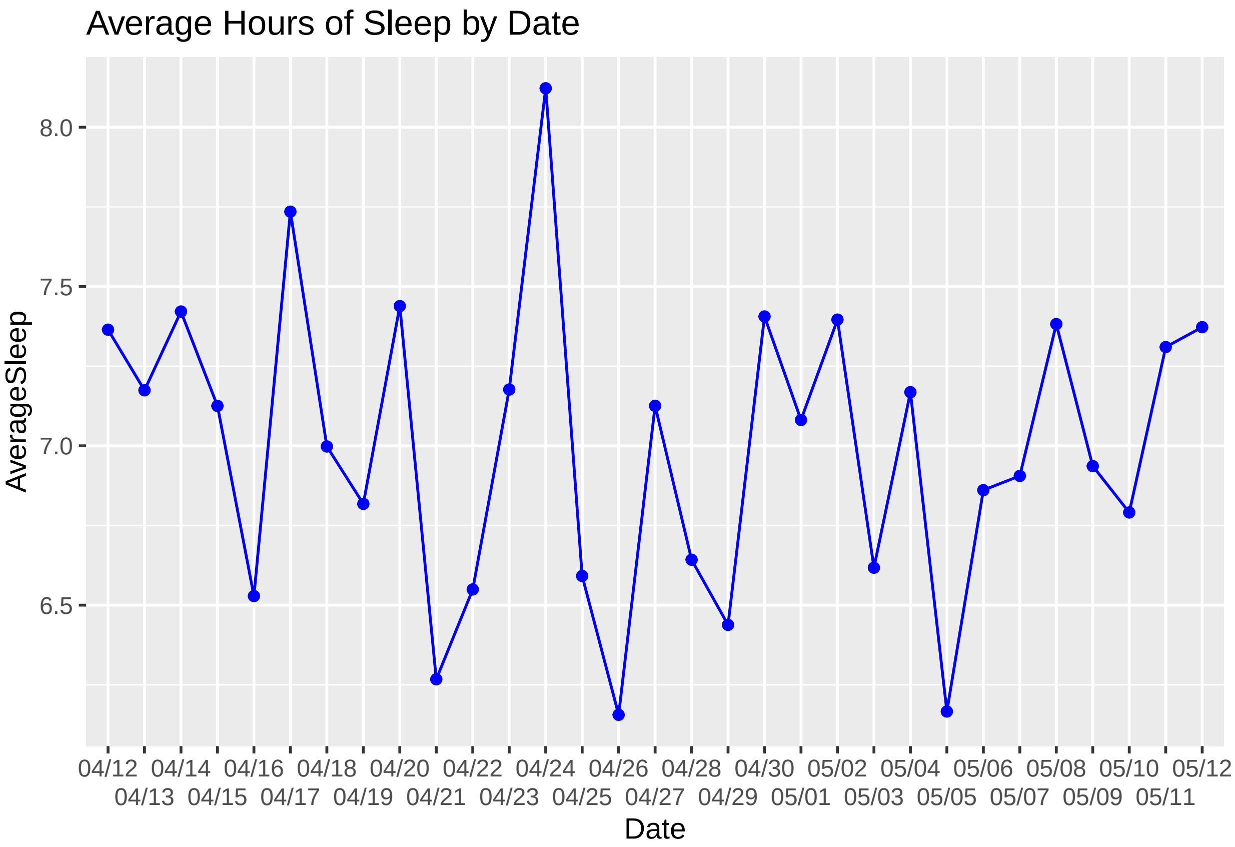
(*Figure 4*)

```
# Create average sleep per day data frame.
sleep_data <- daily_data %>%
  select(Id, Date, DaysOfWeek, TotalMinutesAsleep)

# Convert minutes of sleep to hours.
sleep_data <- mutate(sleep_data, TotalHoursSleep = TotalMinutesAsleep / 60)

# Round hours of sleep column to 2 decimal points
sleep_data$TotalHoursSleep <- round(sleep_data$TotalHoursSleep, 2)

# Create data frame for avg. sleep per day
avg_sleep_days <- sleep_data %>%
  group_by(DaysOfWeek) %>%
  summarize(AverageSleep = mean(TotalHoursSleep))

# Reorganize the order of days of the week
avg_sleep_days$DaysOfWeek <-  ordered(avg_sleep_days$DaysOfWeek,
                                     levels=c("Sunday", "Monday", "Tuesday",
                                              "Wednesday", "Thursday",
                                              "Friday", "Saturday"))

# Create plot for avg. sleep per day
ggplot(data =avg_sleep_days) +
  geom_col(mapping = aes(x=DaysOfWeek, y=AverageSleep, fill=AverageSleep)) +
  labs(title = "Average Sleep by Day")
```



(Figure 5)

```
# Create data frame for avg. sleep per date
avg_sleep_date <- sleep_data %>%
  group_by(Date) %>%
  summarize(AverageSleep = mean(TotalHoursSleep))

# Create Date Object
avg_sleep_date$Date <- as.POSIXct(avg_sleep_date$Date, format = "%m/%d/%Y")
# Date as character to shorten date label for plotting
avg_sleep_date$Date <- format(avg_sleep_date$Date, format = "%m/%d")

# Create plot for avg. sleep per date
ggplot(data = avg_sleep_date) +
  geom_line(mapping = aes(x=Date, y=AverageSleep, group=1,), colour="blue") +
  geom_point(mapping = aes(x=Date, y=AverageSleep), colour="blue") +
  labs(title = "Average Hours of Sleep by Date") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))
```



(*Figure 6*)

We were able to create 6 different graphs to observe insights and trends. Of the 6 graphs, we'll be looking to explore insights and trends based on walking steps, calories, walking intensity, and sleep.

Share

Key Tasks

1. **Determine the best way to share your findings**
2. **Create effective data visualizations**
3. **Present your findings**
4. **Ensure your work is accessible**

The best way to share the results would probably be to print my results in a report or PDF format. This can be done easily through R and it's knitting functionality. Let's dive into our graphs and discuss the insights that were found.

Figure 1: Average Steps by Days of the Week This graph shows the average steps by days of the week from our smart device users. What we see here is that Monday, Tuesday, and Saturday were the highest steps taken throughout the week, which means that the majority of walking exercises occur on those days as well. Out of all the days of the week, on average no days surpass 10,000 steps.

Figure 2: Average Calories by Days of the Week We can observe from this graph that on average the lowest calorie intake is about 2,300, while the highest is about 2,500 calorie intake. The highest calorie intake comes on Saturdays, and lowest on Sundays. Throughout the week the average calorie intake stays within this range.

Figure 3: Intensity Time Vs. Average Activity Levels This is a graph that shows the intensity levels and the average of how long individuals maintained those work outs in those levels. We can see that the average Sedentary (Inactive) minutes have the most at 712.17. Lightly active minutes average 216.85, this could be normal movements throughout the day. We can see that fairly active and very active minutes are the lowest, both being under 30 minutes.

Figure 4: Correlation of Time in Bed and Time Asleep This graph shows a correlation between time in bed and time asleep. There is a high correlation coefficient between these two variables at 0.93. The close to 1.00 means there is a higher correlation, so most people who spend time in bed spend that time sleeping.

Figure 5: Average Sleep by Day We visualized the average hours of sleep by the days of the week in this visualization. We can observe that the range of sleep hours is from lowest, 6.75 to highest, 7.50 hours of sleep. Most people get the highest amount of sleep on Sunday and Wednesday, with the lowest hours of sleep on Tuesday, Thursday, and Friday.

Figure 6: Average Sleep by Date The visualization is the average sleep hours from our individuals by date. The date range is from 4/12/2016 to 5/12/2016. The big takeaway is that the average sleep for our individuals is highly sporadic. There is not a consistent pattern that is noticeable immediately.

Act

Key Tasks

1.**Present your case study to stakeholders**

Let's reflect on the objective. The objective is to analyze how consumers are using smart devices and then identify one Bellabeat product to apply these insights. The insights that were completed can be applied to the Bellabeat App.

Based on the insights we were able to gather, we believe the best way to utilize these insights would be to adapt the Bellabeat App to assist individuals to healthier lifestyles.

According to an article on mayoclinic.org, the recommendation of steps should be 10,000 steps a day, and we see that according to the data many fall short of that number. In that same article, many fall short of the 150 minutes of moderate intensity exercises as well that can help reduce the risk of heart disease, obesity, diabetes, high blood pressure, and depression.

Another area to focus on for the Bellabeat app is individual sleep hours. The CDC suggest for adults 18-60 get 7 or more hours of sleep per night. We saw from our data that the sleep through the month was highly sporadic. The app can assist and try to help individuals get more consistent sleep throughout the week.

The last insight we were able to find in the data is the amount of calorie intake. We were able to see that on average individuals were maintaining the correct amount of calories according to eatright.org.

The Bellabeat app allows for user to understand their health data already. What we can do is have the app offer recommendations based off of this data to assist them in helping them make better decision about their health and lifestyle. We can improve the Bellabeat app by including education information to our users so they can make more efficient health and lifestyle adjustments to their needs. We can also look to run a campaign to bundle our products together that supports overall health and well-being, and when customers buy it in a bundle we can offer a discount for bundling.

The next steps would be to bring these insights to the marketing and development teams to see if there is an area where they can align on to apply these insights to the Bellabeat app and our products.