Checking for duplicate values

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(sleep_day))
```

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

```
sum(duplicated(weight_data))
```

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

Observations

From a quick scan of the loaded datasets the following quick observation were made

The id column is common in all 3 datasets, and can be used to merge the datasets The data type of the Date variable in the 3 datasets are currently character variables and needs to be converted to Date format. * The sleep_data and the weight_data have both date and time merged in one column and need to be be separated, as only the date variable will be used for the analysis. * There were 33 unique users who logged in their daily activities, however, only 24 and 8 unique users logged sleep data and weight data respectively. This implies that most of these users used the device to log their daily activities, but not all of the users track their weight and sleeping habits with the device. * There appears to be no duplicate data in the daily_activity and weight_data, however the sleep_data has 3 duplicates, which need to be removed

Data Cleaning Steps

Will format the column names to lowercase for consistency, and change some of the column names as well

Removing duplicate date from sleep_day

```
# Removing duplicate date from the sleep_day
sleep_day <- distinct(sleep_day)

# Confirm that the duplicate was removed
sum(duplicated(sleep_day))
```

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

```
# Clean column names to lower case
daily_activity <- clean_names(daily_activity)
sleep_day <- clean_names(sleep_day)
weight_data <- clean_names(weight_data)

# Change the activity_date column name to 'date' in the daily_activity dataset
daily_activity <- daily_activity %>%
  dplyr::rename(date = activity_date)
```

Examine the column names

```
colnames(daily_activity)
```

```
##  [1] "id"                       "date"
##  [3] "total_steps"              "total_distance"
##  [5] "tracker_distance"         "logged_activities_distance"
##  [7] "very_active_distance"     "moderately_active_distance"
##  [9] "light_active_distance"    "sedentary_active_distance"
## [11] "very_active_minutes"      "fairly_active_minutes"
## [13] "lightly_active_minutes"   "sedentary_minutes"
```

```
## [15] "calories"
```

```
colnames(weight_data)
```

```
## [1] "id"               "date"             "weight_kg"        "weight_pounds"
## [5] "fat"              "bmi"              "is_manual_report" "log_id"
```

```
colnames(sleep_day)
```

```
## [1] "id"                   "sleep_day"            "total_sleep_records"
## [4] "total_minutes_asleep" "total_time_in_bed"
```

Checking for null values

```
sum(is.na(sleep_day))
```

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

```
sum(is.na(daily_activity))
```

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

```
sum(is.na(weight_data))
```

```
## [1] 65
```

Since there are 65 missing values from the column below, I will remove this due to insufficient data.

```
# remove the 'fat' column with the missing data in the weight_data and the log_id column
weight_data <- select(weight_data, -fat)
weight_data <- select(weight_data, -log_id)
```

```
# view columns
head(weight_data)
```

```
##           id                  date weight_kg weight_pounds   bmi
## 1 1503960366  5/2/2016 11:59:59 PM      52.6      115.9631 22.65
## 2 1503960366  5/3/2016 11:59:59 PM      52.6      115.9631 22.65
## 3 1927972279  4/13/2016 1:08:52 AM     133.5      294.3171 47.54
## 4 2873212765 4/21/2016 11:59:59 PM      56.7      125.0021 21.45
## 5 2873212765 5/12/2016 11:59:59 PM      57.3      126.3249 21.69
## 6 4319703577 4/17/2016 11:59:59 PM      72.4      159.6147 27.45
##   is_manual_report
## 1             True
## 2             True
## 3            False
## 4             True
## 5             True
## 6             True
```

I noticed that some of the date values were chr type, so I will convert the data type from character to a date variable

```
daily_activity$date <- lubridate::mdy(daily_activity$date)
daily_activity <-  mutate(daily_activity, weekday = weekdays(date))

# confirm that the data type is changed from character to date
head(daily_activity)
```

```
## # A tibble: 6 x 16
##       id date       total_steps total_distance tracker_distance logged_activiti~
```

```
##    <dbl> <date>           <dbl>          <dbl>            <dbl>            <dbl>
## 1 1.50e9 2016-04-12       13162           8.5              8.5                0
## 2 1.50e9 2016-04-13       10735           6.97             6.97               0
## 3 1.50e9 2016-04-14       10460           6.74             6.74               0
## 4 1.50e9 2016-04-15        9762           6.28             6.28               0
## 5 1.50e9 2016-04-16       12669           8.16             8.16               0
## 6 1.50e9 2016-04-17        9705           6.48             6.48               0
## # ... with 10 more variables: very_active_distance <dbl>,
## #   moderately_active_distance <dbl>, light_active_distance <dbl>,
## #   sedentary_active_distance <dbl>, very_active_minutes <dbl>,
## #   fairly_active_minutes <dbl>, lightly_active_minutes <dbl>,
## #   sedentary_minutes <dbl>, calories <dbl>, weekday <chr>
```

Will convert the date from chr to date format and add a weekday column

```
# sleep_data cleaning: separate sleep_day column to date and time column, convert the date from charact
sleep_day <- sleep_day %>%
    separate(sleep_day,c("date","time"), sep=" ") %>%
    mutate(date = mdy(date), weekday = weekdays(date)) %>%
    select(-"time")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 410 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
#

sleep_day$weekday <- factor(sleep_day$weekday,
                                levels = c("Monday", "Tuesday", "Wednesday",
                                           "Thursday", "Friday", "Saturday",
                                           "Sunday"))

# confirm that the data type is changed from character to date format
head(sleep_day)
```

```
##           id       date total_sleep_records total_minutes_asleep
## 1 1503960366 2016-04-12                   1                  327
## 2 1503960366 2016-04-13                   2                  384
## 3 1503960366 2016-04-15                   1                  412
## 4 1503960366 2016-04-16                   2                  340
## 5 1503960366 2016-04-17                   1                  700
## 6 1503960366 2016-04-19                   1                  304
##   total_time_in_bed   weekday
## 1               346   Tuesday
## 2               407 Wednesday
## 3               442    Friday
## 4               367  Saturday
## 5               712    Sunday
## 6               320   Tuesday
```

Weight data cleaning: separate date column to date and time column, convert the date from character variable to date format

```
weight_data <- weight_data %>%
  separate(date, c("date", "time"), sep = " ")%>%
  select(-"time")%>%
  mutate(date = mdy(date), weekday = weekdays(date))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 67 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
# confirm that the data type is changed from character to date
head(weight_data)
```

```
##           id       date weight_kg weight_pounds   bmi is_manual_report
## 1 1503960366 2016-05-02      52.6      115.9631 22.65             True
## 2 1503960366 2016-05-03      52.6      115.9631 22.65             True
## 3 1927972279 2016-04-13     133.5      294.3171 47.54            False
## 4 2873212765 2016-04-21      56.7      125.0021 21.45             True
## 5 2873212765 2016-05-12      57.3      126.3249 21.69             True
## 6 4319703577 2016-04-17      72.4      159.6147 27.45             True
##     weekday
## 1    Monday
## 2   Tuesday
## 3 Wednesday
## 4  Thursday
## 5  Thursday
## 6    Sunday
```

Merge Data

I will combine the daily_activity and sleep_day in order to do visualizations

```
combined_data <- merge(daily_activity, sleep_day, by=c ('id', 'date'), all = TRUE)
head(combined_data)
```

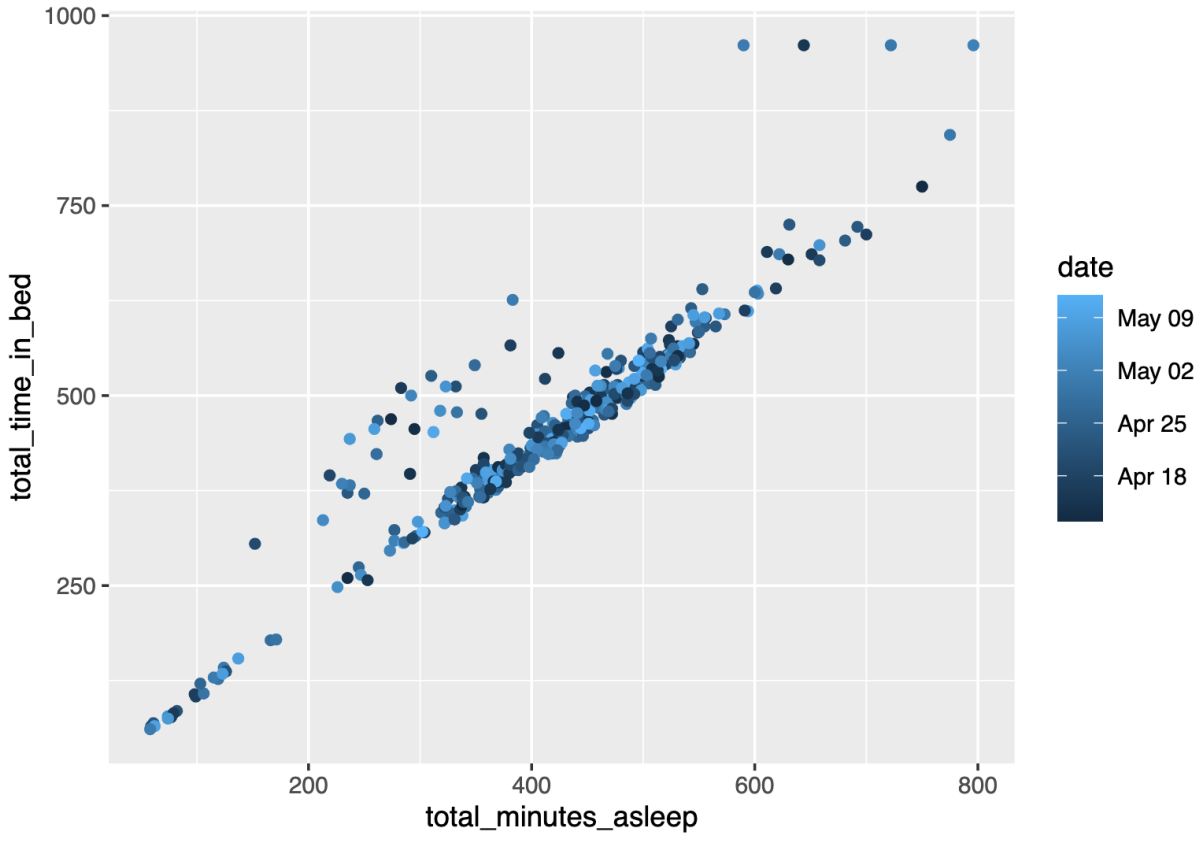
```
##           id       date total_steps total_distance tracker_distance
## 1 1503960366 2016-04-12       13162           8.50             8.50
## 2 1503960366 2016-04-13       10735           6.97             6.97
## 3 1503960366 2016-04-14       10460           6.74             6.74
## 4 1503960366 2016-04-15        9762           6.28             6.28
## 5 1503960366 2016-04-16       12669           8.16             8.16
## 6 1503960366 2016-04-17        9705           6.48             6.48
##   logged_activities_distance very_active_distance moderately_active_distance
## 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
##   light_active_distance sedentary_active_distance very_active_minutes
## 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
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    13                   328               728     1985
## 2                    19                   217               776     1797
## 3                    11                   181              1218     1776
## 4                    34                   209               726     1745
## 5                    10                   221               773     1863
## 6                    20                   164               539     1728
```

```
##   weekday.x total_sleep_records total_minutes_asleep total_time_in_bed
## 1   Tuesday                   1                  327                346
## 2 Wednesday                   2                  384                407
## 3  Thursday                  NA                   NA                 NA
## 4    Friday                   1                  412                442
## 5  Saturday                   2                  340                367
## 6    Sunday                   1                  700                712
##   weekday.y
## 1   Tuesday
## 2 Wednesday
## 3      <NA>
## 4    Friday
## 5  Saturday
## 6    Sunday
```

Share

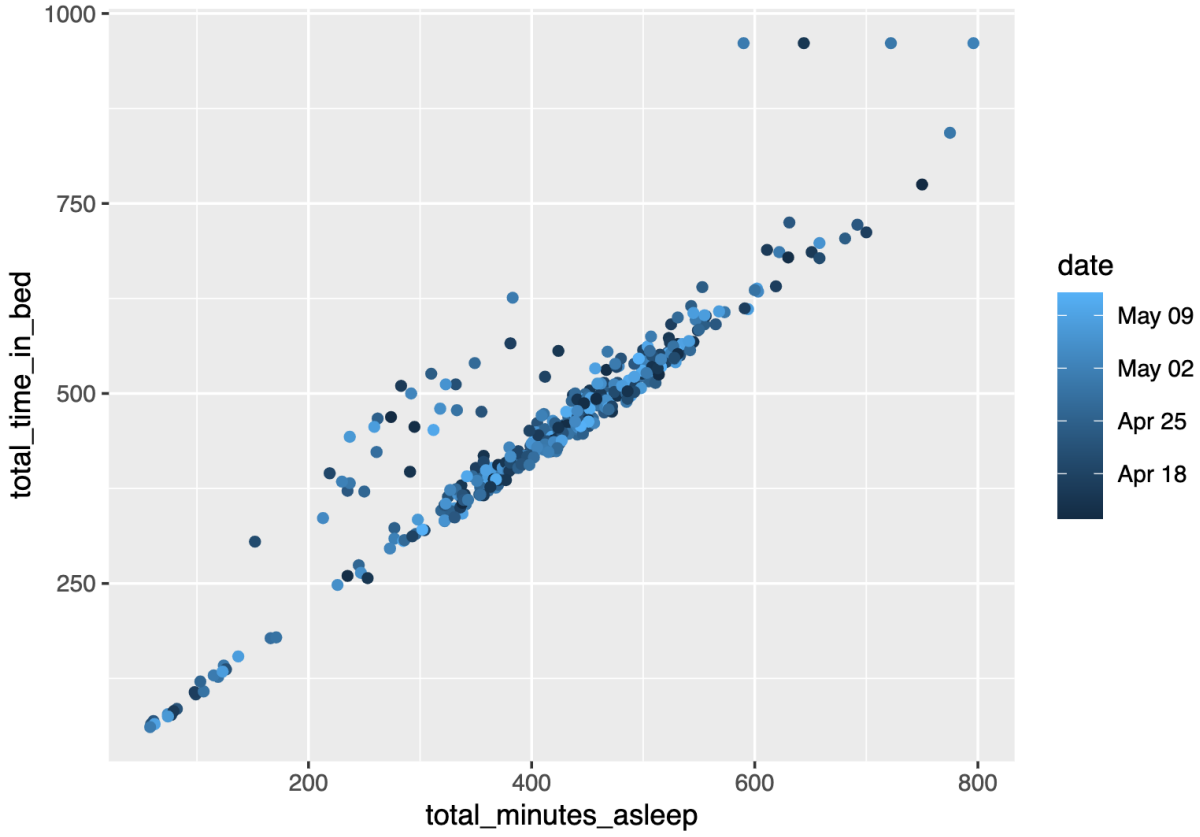
Scatter plot to show the time in bed vs time alseep

```
ggplot(data=sleep_day, aes(x=total_minutes_asleep, y=total_time_in_bed)) + geom_point(aes(color=date))
```



Scatter plot to show the sedentary minutes vs total steps

```
ggplot(data=sleep_day, aes(x=total_minutes_asleep, y=total_time_in_bed)) + geom_point(aes(color=date))
```




```
daily_average <- combined_data %>%
  group_by(id) %>%
  summarise (mean_daily_steps = mean(total_steps), mean_daily_calories = mean(calories), mean_daily_sle

head(daily_average)
```

```
## # A tibble: 6 x 4
##           id mean_daily_steps mean_daily_calories mean_daily_sleep
##        <dbl>            <dbl>               <dbl>            <dbl>
## 1 1503960366           12117.               1816.               NA
## 2 1624580081            5744.               1483.               NA
## 3 1644430081            7283.               2811.               NA
## 4 1844505072            2580.               1573.               NA
## 5 1927972279             916.               2173.               NA
## 6 2022484408           11371.               2510.               NA
```

The dataset did not include any demographic information about the users. We can classify the users by activity considering the daily amount of steps. We can categorize users as follows: I used this article as a reference to determine how to categorize them.

Sedentary - Less than 5000 steps a day. Lightly active - Between 5000 and 7499 steps a day. Fairly active - Between 7500 and 9999 steps a day. Very active - More than 10000 steps a day.

```
user_category <- daily_average %>%
  mutate(user_category = case_when(
```

```
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7499 ~ "lightly active",
    mean_daily_steps >= 7500 & mean_daily_steps < 9999 ~ "fairly active",
    mean_daily_steps >= 10000 ~ "very active"
  ))

head(user_category)
```

```
## # A tibble: 6 x 5
##           id mean_daily_steps mean_daily_calories mean_daily_sleep user_category
##        <dbl>            <dbl>               <dbl>            <dbl> <chr>
## 1 1503960366           12117.               1816.               NA very active
## 2 1624580081            5744.               1483.               NA lightly acti~
## 3 1644430081            7283.               2811.               NA lightly acti~
## 4 1844505072            2580.               1573.               NA sedentary
## 5 1927972279             916.               2173.               NA sedentary
## 6 2022484408           11371.               2510.               NA very active
```

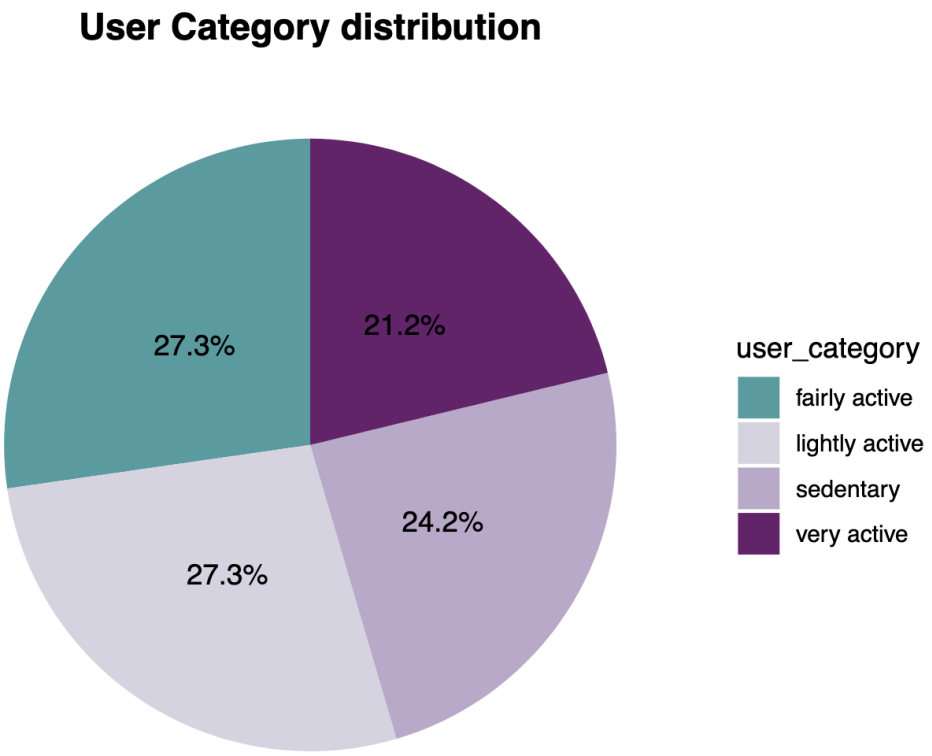
```
user_category_percent <- user_category %>%
  group_by(user_category) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_category) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

user_category_percent$user_type <- factor(user_category_percent$user_category , levels = c("very active


head(user_category_percent)
```

```
## # A tibble: 4 x 4
##   user_category  total_percent labels user_type
##   <chr>                  <dbl> <chr>  <fct>
## 1 fairly active          0.273 27.3%  fairly active
## 2 lightly active         0.273 27.3%  lightly active
## 3 sedentary              0.242 24.2%  sedentary
## 4 very active            0.212 21.2%  very active
```

```
user_category_percent %>%
  ggplot(aes(x="",y=total_percent, fill=user_category)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  scale_fill_manual(values = c("#5b9aa0","#d6d4e0", "#b8a9c9", "#622569")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  labs(title="User Category distribution")
```



Verifying the combined data

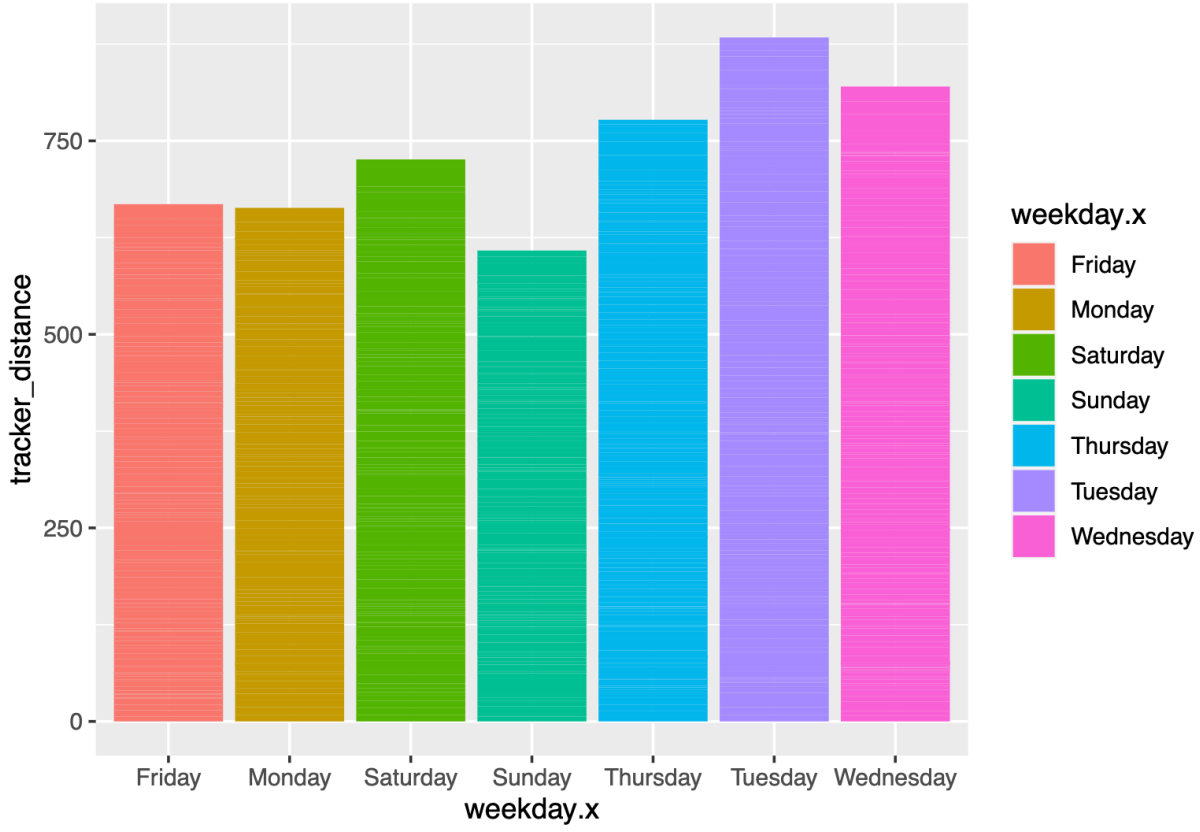
```
head(combined_data)
```

```
##           id       date total_steps total_distance tracker_distance
## 1 1503960366 2016-04-12       13162           8.50             8.50
## 2 1503960366 2016-04-13       10735           6.97             6.97
## 3 1503960366 2016-04-14       10460           6.74             6.74
## 4 1503960366 2016-04-15        9762           6.28             6.28
## 5 1503960366 2016-04-16       12669           8.16             8.16
## 6 1503960366 2016-04-17        9705           6.48             6.48
##   logged_activities_distance very_active_distance moderately_active_distance
## 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
##   light_active_distance sedentary_active_distance very_active_minutes
## 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
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    13                   328               728     1985
## 2                    19                   217               776     1797
## 3                    11                   181              1218     1776
## 4                    34                   209               726     1745
```

```
## 5                    10                   221              773     1863
## 6                    20                   164              539     1728
##   weekday.x total_sleep_records total_minutes_asleep total_time_in_bed
## 1   Tuesday                   1                  327               346
## 2 Wednesday                   2                  384               407
## 3  Thursday                  NA                   NA                NA
## 4    Friday                   1                  412               442
## 5  Saturday                   2                  340               367
## 6    Sunday                   1                  700               712
##   weekday.y
## 1   Tuesday
## 2 Wednesday
## 3      <NA>
## 4    Friday
## 5  Saturday
## 6    Sunday
```

Graph to show the tracker distance usage for each day of the week

```
ggplot(data = combined_data, aes( x = weekday.x, y = tracker_distance, fill = weekday.x)) +
  geom_bar(stat = "identity")
```



Act

After reviewing the data, I will present my recommendations on how Bellabeat can use these insights to improve their Marketing stategies.

Observations

The data we were working with is from 2016, from an Amazon Mechanical Turk survey. It might be best to gather more recent data to make sure our findings are current. It appears that many FitBit users do not wear their device consistently, so having a reminder might help users to remmember to wear them more often. They could have rewards similar to the Apple watch, that helps motivate you to reach milestones. There was a lot of sedentary time in this group, The CDC recommends 30 minutes of activity each day. That is why having a reminder on the device would help motivate users to stay more active. It appears that many users were wearing the device in bed, but not asleep. This is most likely due to using their phone before going to bed. So, they might want to set screen time limitations, so their sleep is not negatively effected by blue light before going to sleep. Marketing Suggestions for Bellabeat Bellabeat can create a podcast or blog that talks about healthy lifestyle and the importance of daily exercise. They could have monthly incentives for meeting milestones for their users, such as free swag, or a discount towards one of their products.