Google Analytics Case Study

David Cohn

6/21/2021

Part 1: Ask

The business task of this case study is to analyze publicly available wearable device data from a cohort of Fitbit users (who agreed to provide their data), and use the conclusions from that analysis to advise the startup Bellabeat regarding their health-related products. Bellabeat has a series of health wellness products focusing on sleep, stress, activity, hydration and more. As such, insights drawn from a dataset covering comparable focuses could provide Bellabeat with characteristics to look for in their customer cohort.

```
# R Packages Used in Pre-Processing and Analysis
library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)
library(scales)
```

Part 2: Prepare

The Fitbit dataset comprises publicly available data (CC0: Public Domain) for thirty subjects (who agreed to provide their data), covering roughly a two month period. The dataset has day, hour, minute and second level data spanning activity, heart rate, steps, calories, sleep and more. For my analysis, I focused on the following data tables within the data set: day-level activity and sleep data, second-level heartrate data, and hour-level calories, intensities and steps data. The activity data table contains a subject identifier, the date of the activity, as well as total steps, distance and calories. In addition, each subject's activity is broken down according to four categories, from sedentary (lowest level of activity) to very active (highest level of activity). The day-level sleep data contains an identifier, the date, the total number of minutes asleep and the total time in bed. The second level heart rate data details heart rate values for the cohort on five second intervals. The hour level calorie data documents calories burned each hour, along with the date and an identifier, while the hour level steps data has the number of steps covered each hour. Finally, the intensity data mirrors the aforementioned four-tiered system from the activity data, where the intensity of activity over the entire hour is collated and averaged to produce a score from 0 (full sedentary) to 3 (Very Active).

```
# Data Tables Used in Analysis
activity.data = read.csv('dailyActivity_merged.csv')
heart.rate.data = read.csv('heartrate_seconds_merged.csv')
hour.calories.data = read.csv('hourlyCalories_merged.csv')
hour.intensities.data = read.csv('hourlyIntensities_merged.csv')
hour.steps.data = read.csv('hourlySteps_merged.csv')
day.sleep.data = read.csv('sleepDay_merged.csv')
```

```
activity.data = rename(activity.data, Date = ActivityDate)

# Convert Date to Date-Time Format
activity.data$Date = as.Date(activity.data$Date, format = '%m/%d/%Y')
```

Part 3: Process

As part of the data cleaning process, I examined each data table (using the View pane in R Studio) for outliers, noisy or missing data. I also considered the summary statistics shown below in helping me to identify these characteristics in the data. Ultimately, I encountered 80 data entries out of 940 total entries (across days and subjects) with either missing total steps data (Total Steps = 0), Calories data (Calories burned = 0), or activity minutes data (where the total number of activity minutes is less than 4 hours, indicating the Fitbit was worn for less than 4 hours of a 24 hour day). Removing these entries resulted in 860 rows of data. To ensure that this missing data was not included in other wearable device tables, I made sure to remove the identifier-date pairs that had missing data in the activity table from the other corresponding tables.

```
activity.data.features = select(activity.data, -Id, -Date)

# Summary Statistics For Activity Data
summary(activity.data.features)
```

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

```
remove(activity.data.features)
```

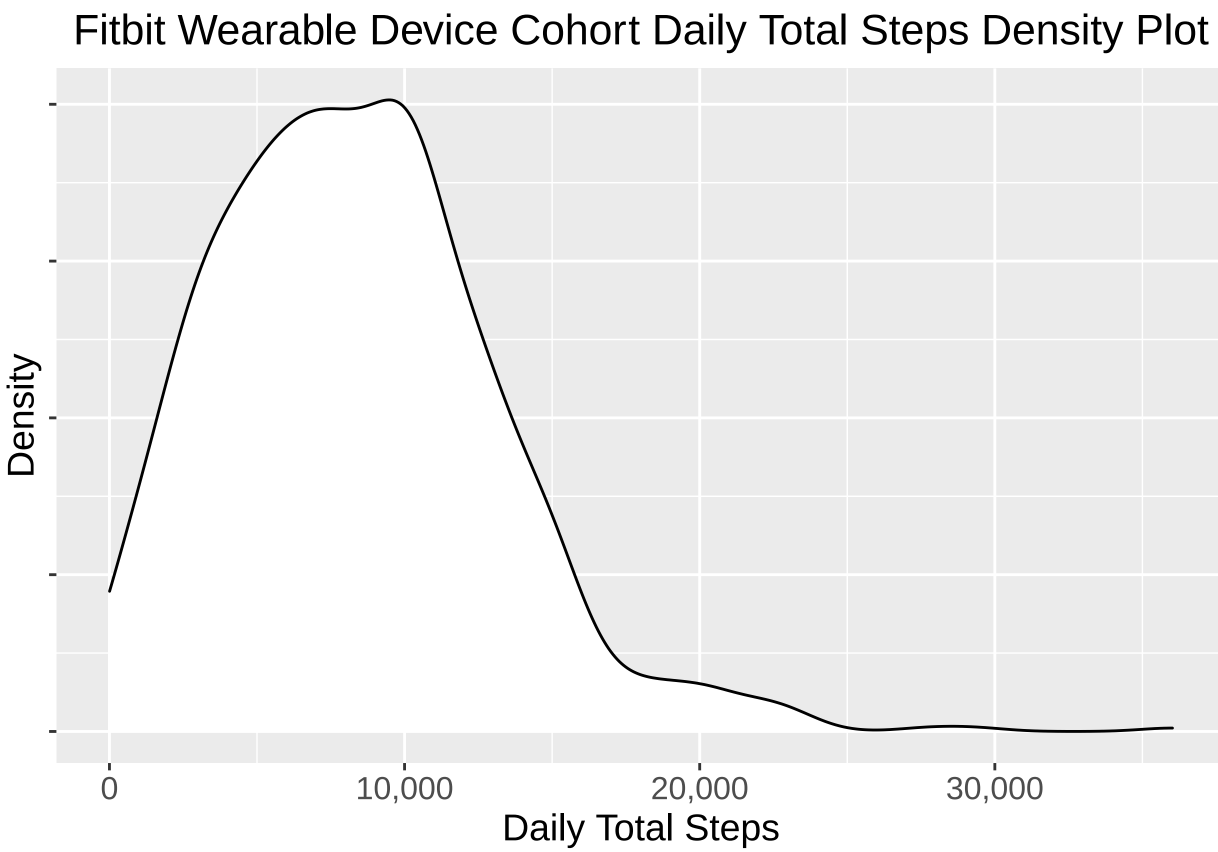
```
filtered.activity.data = activity.data %>%
  mutate(Total.Activity.Minutes = VeryActiveMinutes + FairlyActiveMinutes +
         LightlyActiveMinutes + SedentaryMinutes) %>%
  filter(TotalSteps > 0) %>%
  filter(Calories > 0) %>%
  filter(Total.Activity.Minutes >= 240)
```

Part 4: Analyze

After removing the missing data described in the previous section, I proceeded to analyze my data, starting with the activity table.

I first produced a density plot of the daily total steps by the cohort. As shown below, the data is left-skewed, with far more instances of lower levels of activity (corresponding to a smaller number of steps) than high levels of activity. Taken together, this suggests that this cohort (based on this two month period) may not be particularly active.

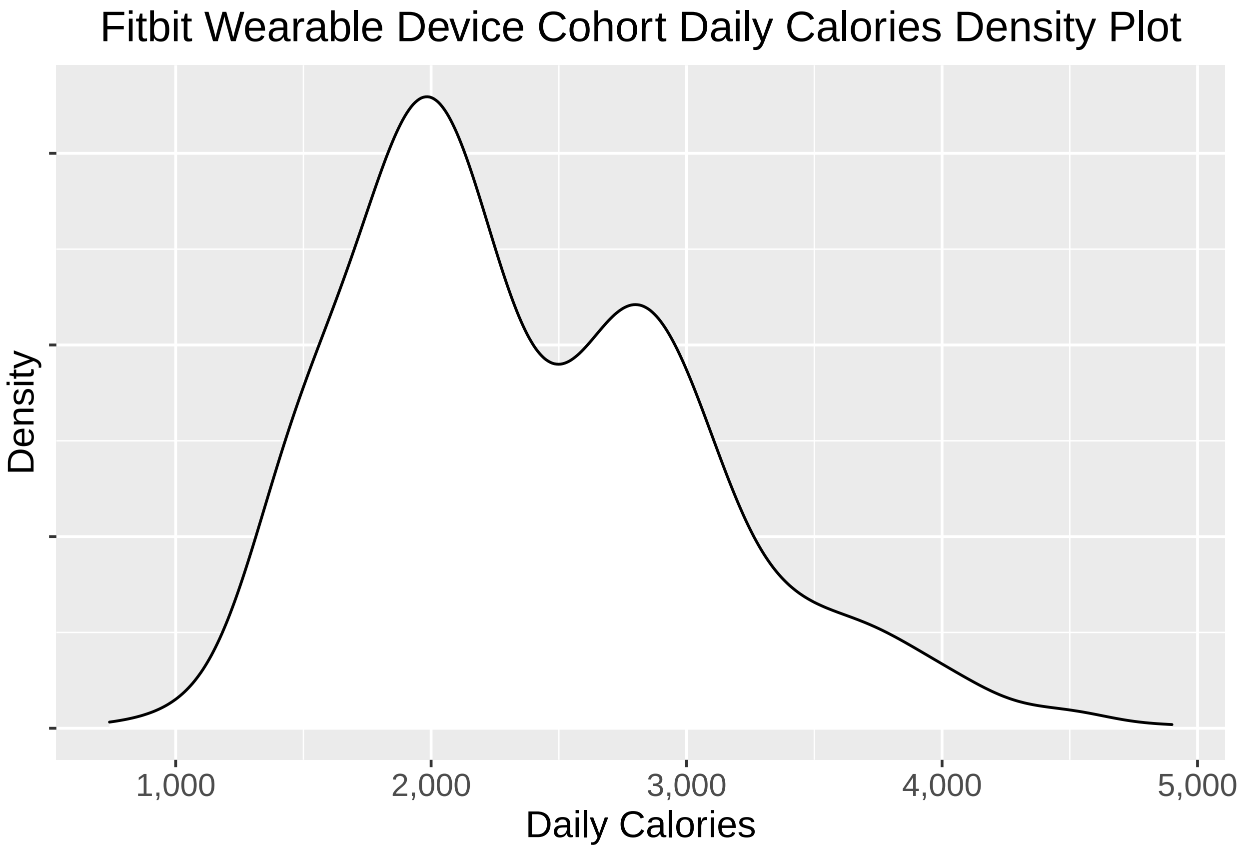
```
total.steps.density.plot = ggplot(filtered.activity.data, aes(x = TotalSteps)) +
  geom_density(colour = "black", fill = "white") + scale_x_continuous(labels = comma) +
  labs(title = "Fitbit Wearable Device Cohort Daily Total Steps Density Plot",
       x = "Daily Total Steps", y = "Density") + theme(axis.text.y = element_blank(),
       axis.text = element_text(size = 12), axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16))

print(total.steps.density.plot)
```



Next, I analyzed the daily calories for the Fitbit cohort. The Calorie density plot is also left-skewed, with a modal daily calories burned of ~2000 calories. As such, in terms of calories burned, this cohort is comparable to the broader public, where a 2000 calorie diet is considered a standard for assessing and labeling food for their ingredient composition.

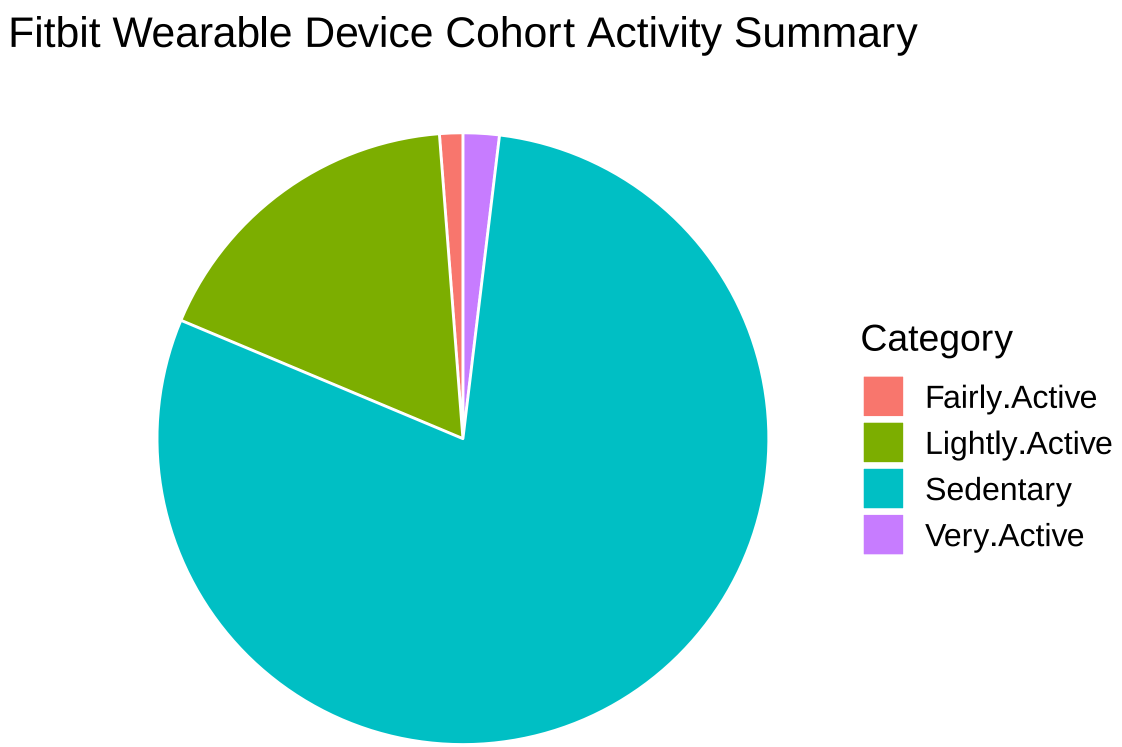
```
calories.density.plot = ggplot(filtered.activity.data, aes(x = Calories)) +
  geom_density(colour = "black", fill = "white") + scale_x_continuous(labels = comma) +
  labs(title = "Fitbit Wearable Device Cohort Daily Calories Density Plot",
       x = "Daily Calories", y = "Density") + theme(axis.text.y = element_blank(),
       axis.text = element_text(size = 12), axis.title = element_text(size = 14),
```

```
      plot.title = element_text(hjust = 0.5, size = 16))

print(calories.density.plot)
```



I subsequently analyzed the percentage of time spent for each activity category (sedentary, lightly active, fairly active, and very active), which was summarized in a pie chart. As shown below, the vast majority of time spent by the cohort was in sedentary (low to no activity) "mode", further supporting the hypothesis above that this cohort of people is not as active.

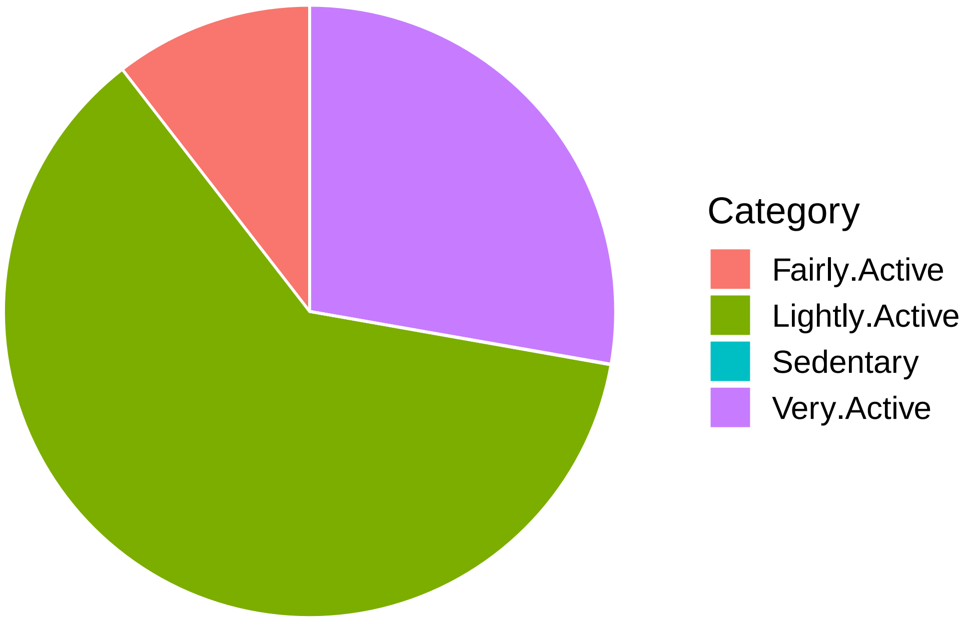
```
activity.duration.cohort.summary = filtered.activity.data %>%
  summarise(Very.Active = sum(VeryActiveMinutes) / sum(Total.Activity.Minutes) * 100,
            Fairly.Active = sum(FairlyActiveMinutes) / sum(Total.Activity.Minutes) * 100,
            Lightly.Active = sum(LightlyActiveMinutes) / sum(Total.Activity.Minutes) * 100,
            Sedentary = sum(SedentaryMinutes) / sum(Total.Activity.Minutes) * 100) %>%
  gather(Category, Percentage)

activity.duration.summary.pie.chart = ggplot(activity.duration.cohort.summary,
  aes(x = "", y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", color = "white") + coord_polar("y", start = 0) +
  theme_void() + labs(title = "Fitbit Wearable Device Cohort Activity Summary",
      caption = "Percentage of Time Spent by Activity Category") +
      theme(plot.title = element_text(hjust = 0.5, size = 16),
            legend.text = element_text(size = 12),
      legend.title = element_text(size = 14),
      plot.caption = element_text(size = 12, hjust = 0.5))

print(activity.duration.summary.pie.chart)
```



Percentage of Time Spent by Activity Category

```
activity.distance.cohort.summary = filtered.activity.data %>%
  summarise(Very.Active = sum(VeryActiveDistance) / sum(Total.Activity.Minutes) * 100,
            Fairly.Active = sum(ModeratelyActiveDistance) / sum(Total.Activity.Minutes) * 100,
            Lightly.Active = sum(LightActiveDistance) / sum(Total.Activity.Minutes) * 100,
            Sedentary = sum(SedentaryActiveDistance) / sum(Total.Activity.Minutes) * 100) %>%
  gather(Category, Percentage)

activity.distance.summary.pie.chart = ggplot(activity.distance.cohort.summary,
  aes(x = "", y = Percentage, fill = Category)) +
  geom_bar(stat = "identity", color = "white") + coord_polar("y", start = 0) +
  theme_void() + labs(title = "Fitbit Wearable Device Cohort Activity Summary",
       caption = "Percentage of Distance Traveled by Activity Category") +
       theme(plot.title = element_text(hjust = 0.5, size = 16),
             legend.text = element_text(size = 12),
       legend.title = element_text(size = 14),
       plot.caption = element_text(size = 12, hjust = 0.5))

print(activity.distance.summary.pie.chart)
```

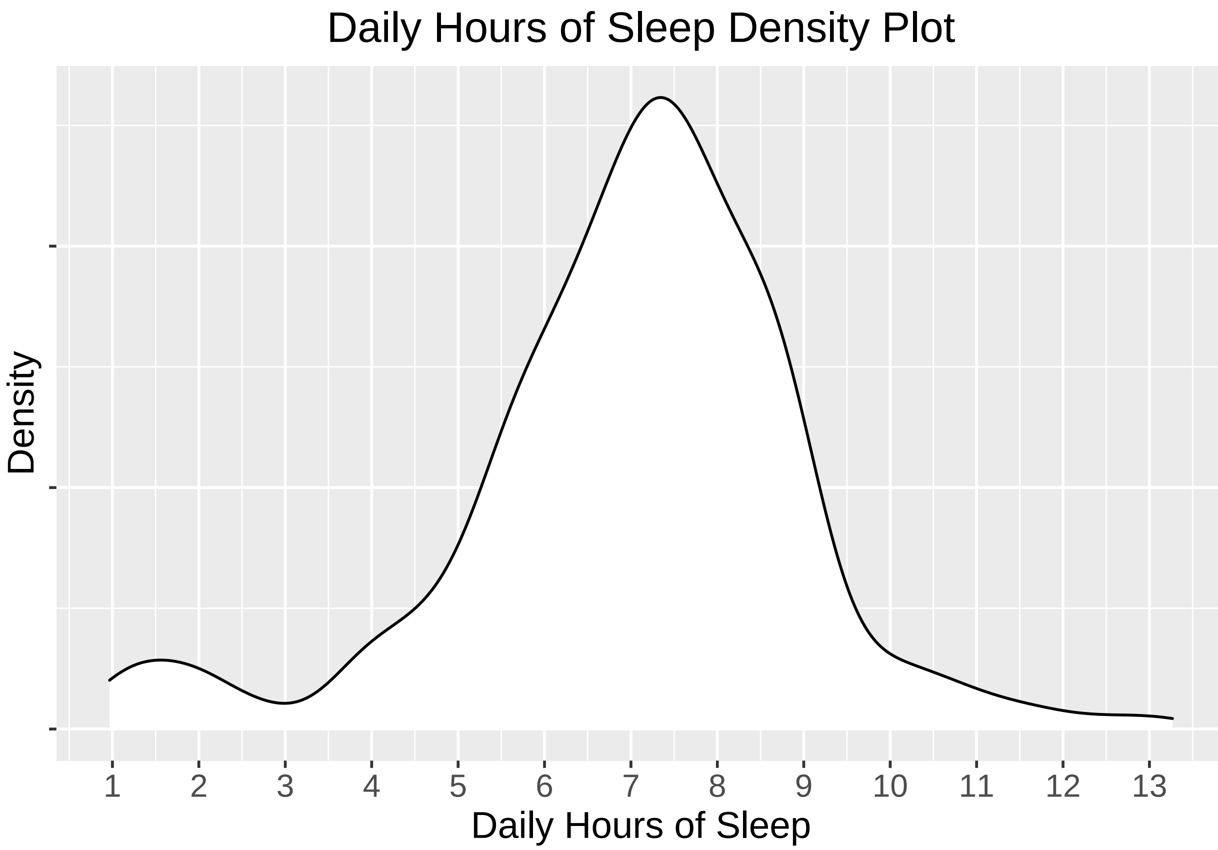
Fitbit Wearable Device Cohort Activity Summary



Percentage of Distance Traveled by Activity Category

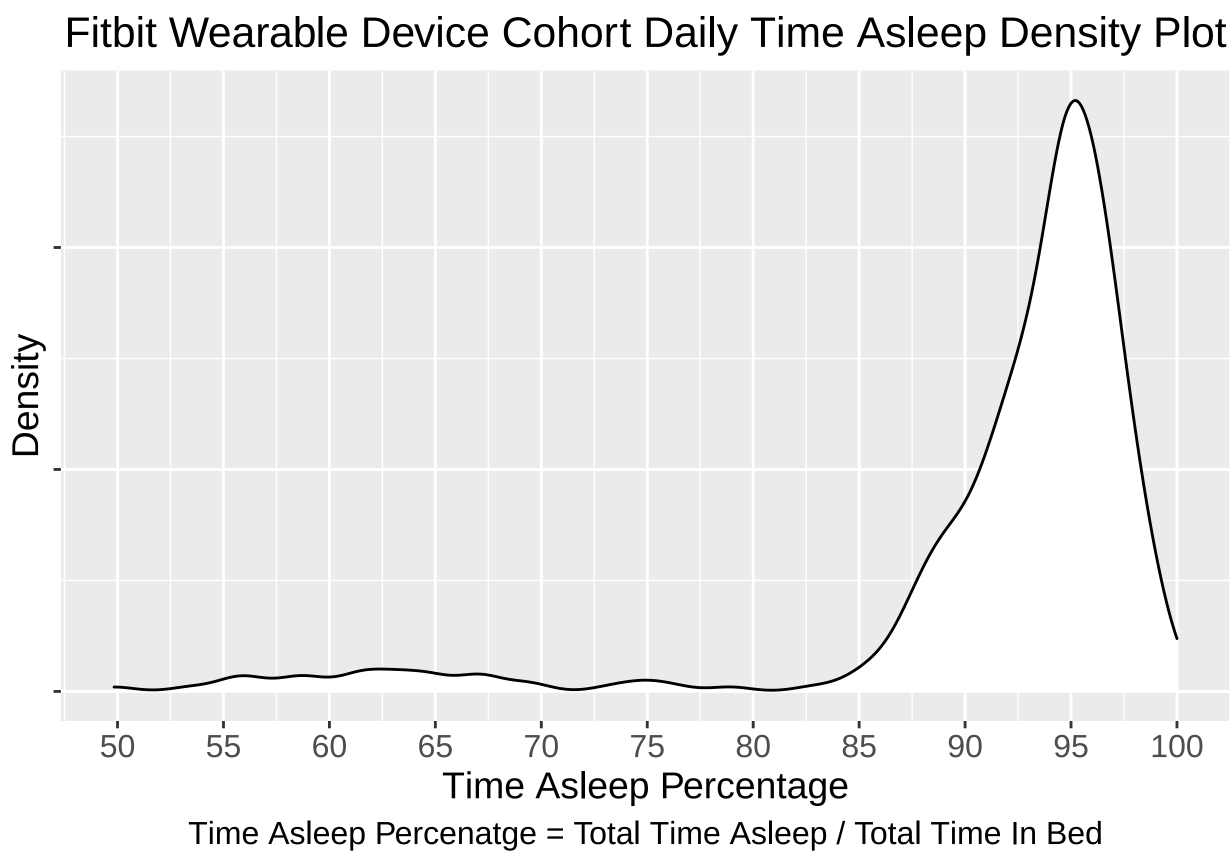
After examining the activity data, I looked at the sleep data for the group. The density plot below shows the modal amount of sleep is slightly more than 7 hours a night, with the vast majority of the cohort sleeping between 4 and 10 hours a night. This finding is comparable to current sleep guidelines, which recommend at least 7 hours of sleep per night.

```
day.sleep.data$SleepDay = mdy_hms(day.sleep.data$SleepDay)

day.sleep.data = day.sleep.data %>%
  mutate(Date = as.Date(SleepDay)) %>%
  mutate(Day.of.Week = wday(Date, label = TRUE, abbr = FALSE)) %>%
  select(-SleepDay) %>%
  mutate(Sleep.Hours = TotalMinutesAsleep / 60) %>%
  select(Id, Date, TotalMinutesAsleep, TotalTimeInBed, Sleep.Hours, Day.of.Week) %>%
  mutate(Asleep.Percentage = TotalMinutesAsleep / TotalTimeInBed * 100)

sleep.trends.density.plot = ggplot(day.sleep.data, aes(x = Sleep.Hours)) +
  geom_density(colour = "black", fill = "white") + scale_x_continuous(breaks = seq(0, 13)) +
  labs(title = "Daily Hours of Sleep Density Plot",
       x = "Daily Hours of Sleep", y = "Density") + theme(axis.text.y = element_blank(),
       axis.text = element_text(size = 12), axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16))

print(sleep.trends.density.plot)
```



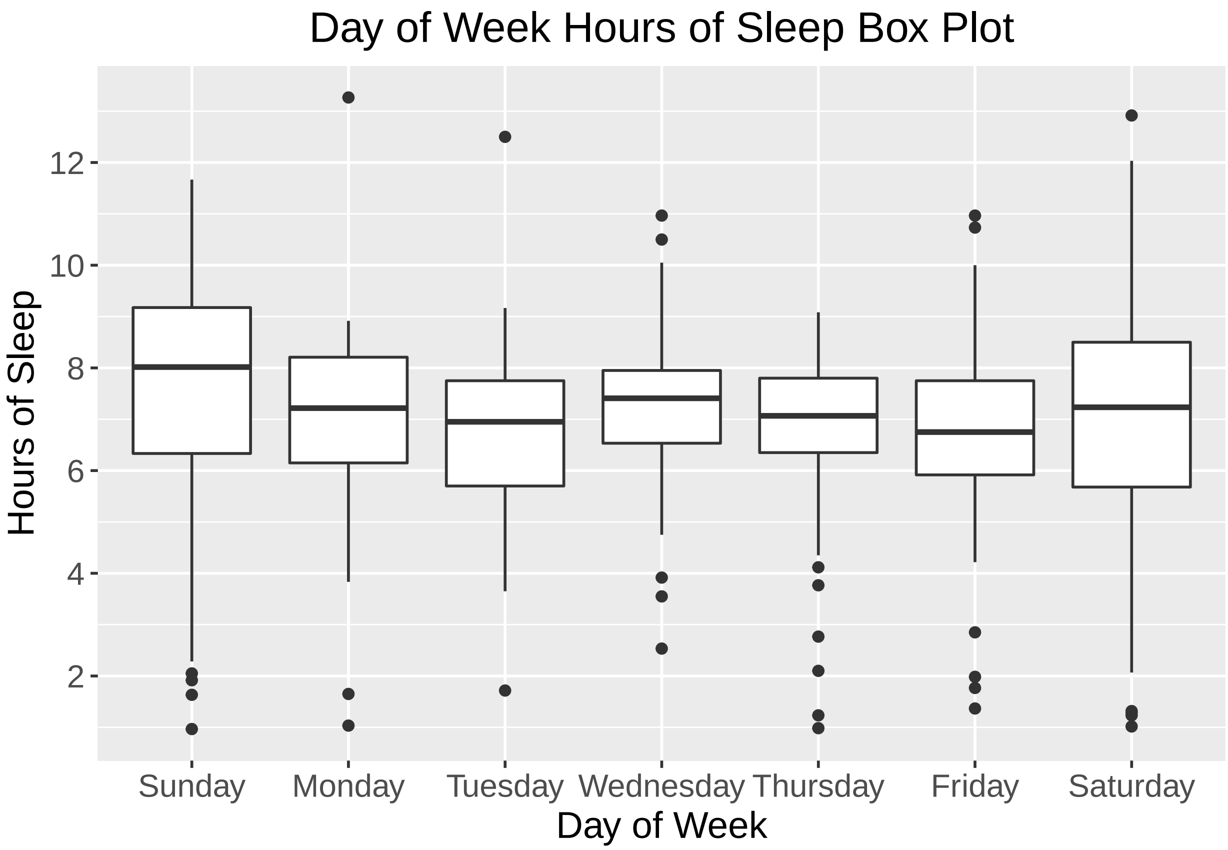
In order to assess possible sleep restlessness, I calculated the percentage of time the cohort was sleeping out of the total time spent in bed. Visualized below, the vast majority of time spent in bed was defined by sleeping (modal result >95%). As such, it seems most of the cohort on most nights did not appear to suffer from sleeplessness. However, there is a noticable left tail in the density plot distribution, indicating evidence of sleeplessness in a small portion of the cohort on at least some nights.

```
asleep.percentage.density.plot = ggplot(day.sleep.data, aes(x = Asleep.Percentage)) +
  geom_density(colour = "black", fill = "white") + scale_x_continuous(breaks = seq(50, 100, 5)) +
  labs(title = "Fitbit Wearable Device Cohort Daily Time Asleep Density Plot",
       caption = "Time Asleep Percenatge = Total Time Asleep / Total Time In Bed",
       x = "Time Asleep Percentage", y = "Density") + theme(axis.text.y = element_blank(),
       axis.text = element_text(size = 12), axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16),
       plot.caption = element_text(size = 12, hjust = 0.5))

print(asleep.percentage.density.plot)
```



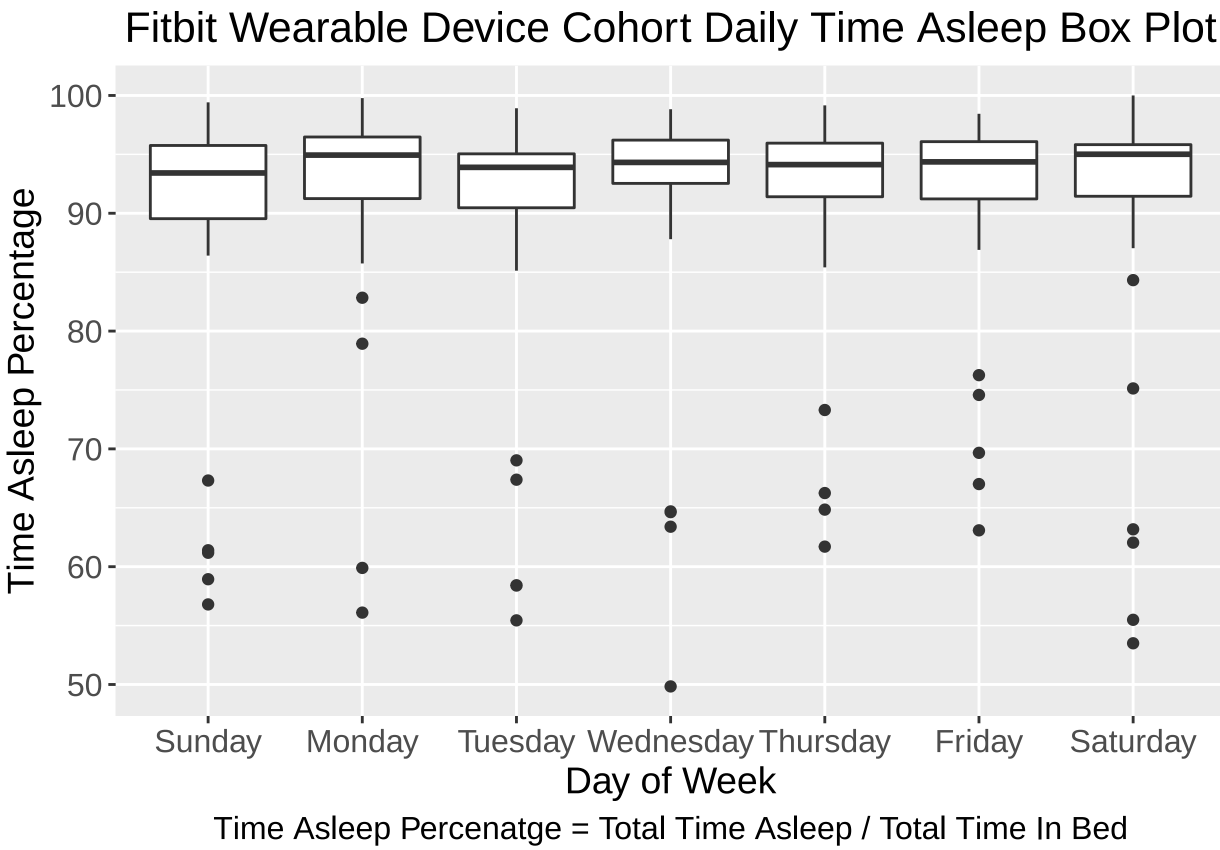
Next, I examined whether sleep/restlessness was defined by any possible day of the week effect. As shown below, the median amount of sleep (black bar in each box) was slightly higher on Sunday than the other days of the week; this presumably makes sense, as the cohort would potentially have more time on the weekend/"non-workdays" for sleep. There was also greater variation (as measured by 25%-75% spread, which corresponds to the height of the box) on the weekend. This also seemingly makes sense, as some people might use the weekend to catch up on sleep, while others might use the free time to engage in leisure activities.

```
day.of.week.hours.sleep.boxplot = ggplot(day.sleep.data,
  aes(x = Day.of.Week, y = Sleep.Hours)) + geom_boxplot() +
  labs(title = "Day of Week Hours of Sleep Box Plot",
       x = "Day of Week", y = "Hours of Sleep") + theme(axis.text = element_text(size = 12),
       axis.title = element_text(size = 14), plot.title = element_text(hjust = 0.5, size = 16),
       plot.caption = element_text(size = 12, hjust = 0.5)) +
  scale_y_continuous(breaks = seq(0, 14, 2))

print(day.of.week.hours.sleep.boxplot)
```



However, there did not seem to be any noticable day-of-week effect on sleep restlessness; "work days" (week days) and "non-work" days (weekends) had similar percentage time asleep.

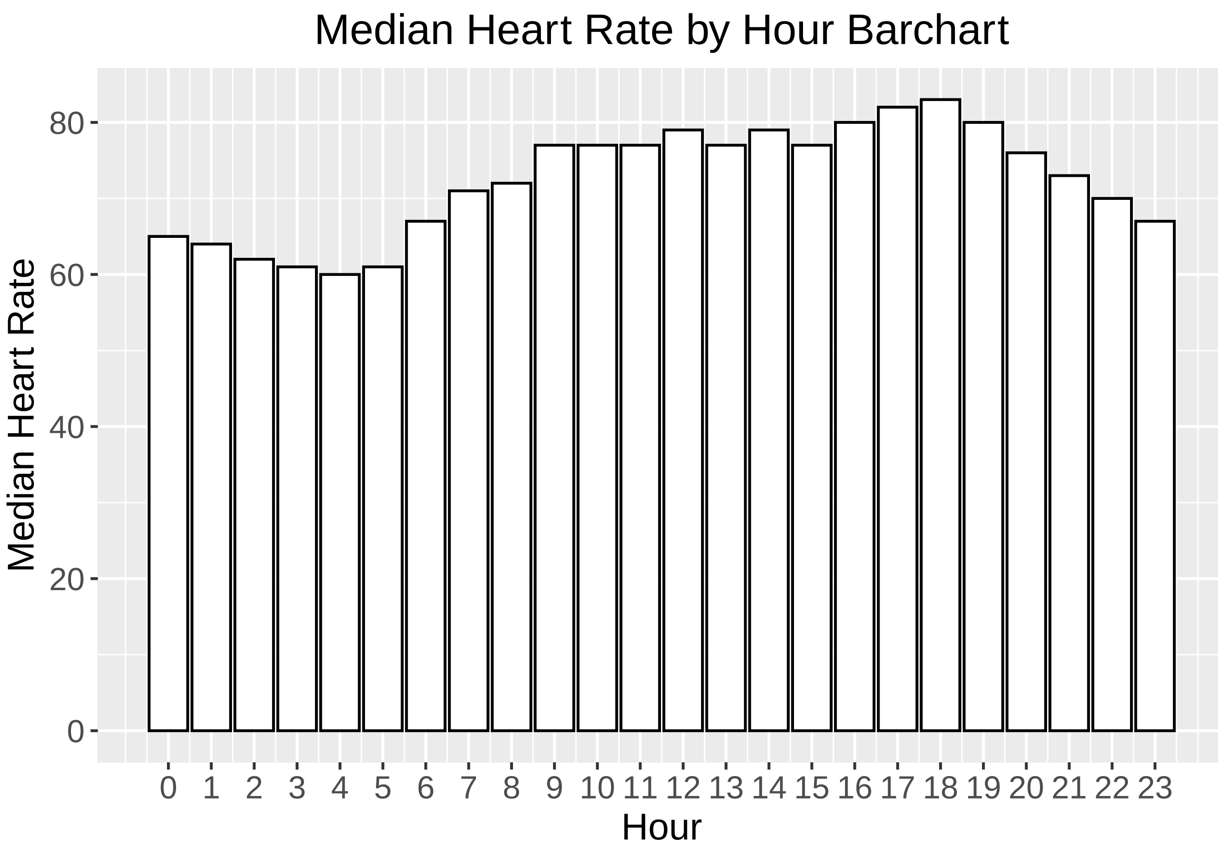
```
day.of.week.time.asleep.boxplot = ggplot(day.sleep.data,
  aes(x = Day.of.Week, y = Asleep.Percentage)) + geom_boxplot() +
  labs(title = "Fitbit Wearable Device Cohort Daily Time Asleep Box Plot",
       caption = "Time Asleep Percenatge = Total Time Asleep / Total Time In Bed",
       x = "Day of Week", y = "Time Asleep Percentage") +
  theme(axis.text = element_text(size = 12),
       axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16),
       plot.caption = element_text(size = 12, hjust = 0.5))

print(day.of.week.time.asleep.boxplot)
```



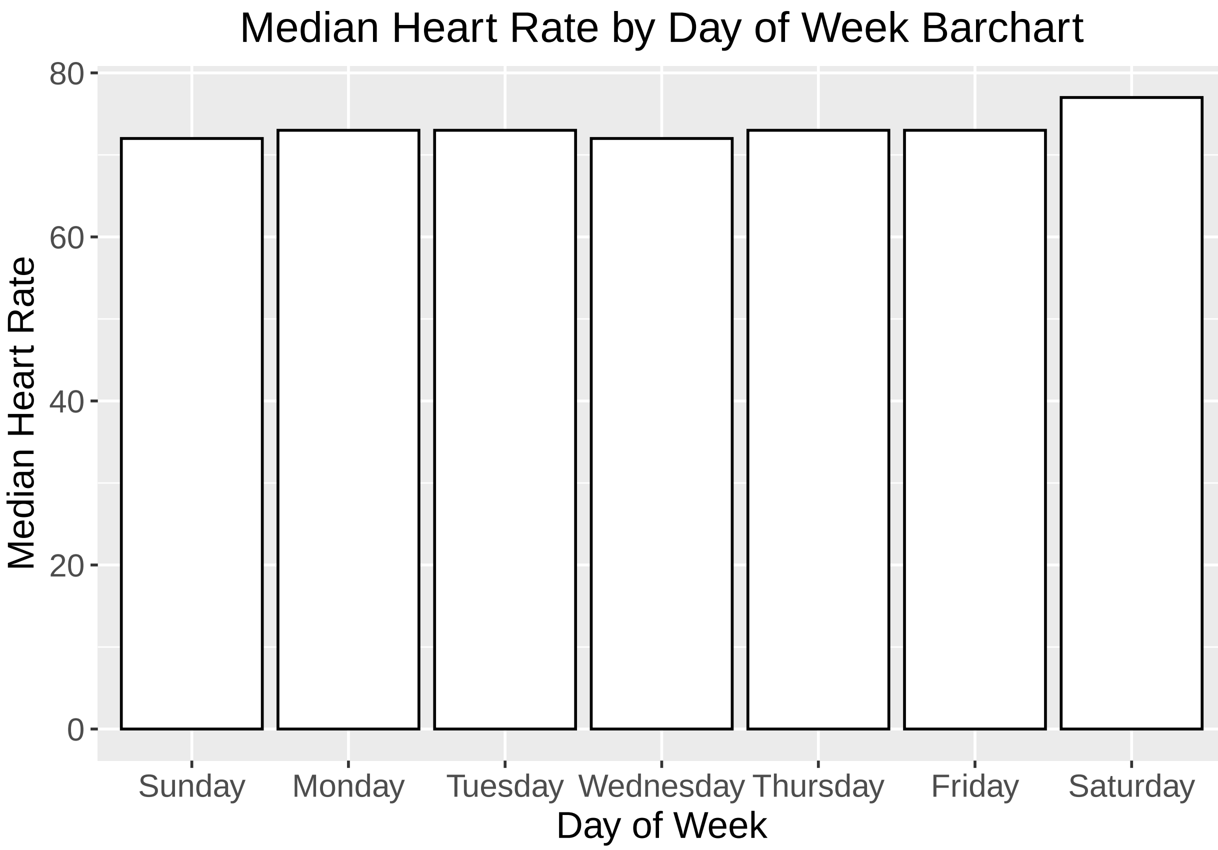
Third, I analyzed hourly heart rate trends through the use of a barplot. The median heart rate for the cohort was highest during the 6:00pm hour, which would place that hour presumably after the 9:00am-5:00pm "work day". Hours in the late evening/early morning, presumably when people would be sleeping, had the lowest median heart rates. There was no noticable day-of-the week effect in median heart rates.

```
heart.rate.data$Time = mdy_hms(heart.rate.data$Time)

heart.rate.data = heart.rate.data %>%
  rename(Date.Time = Time) %>%
  mutate(Date = as.Date(Date.Time)) %>%
  mutate(Hour = hour(Date.Time)) %>%
  mutate(Day.of.Week = wday(Date.Time, label = TRUE, abbr = FALSE))

heart.rate.data = filtered.activity.data %>%
  select(Id, Date) %>%
  inner_join(heart.rate.data, by = c("Id", "Date"))

median.heart.rate.hour.summary = heart.rate.data %>%
  group_by(Hour) %>%
  summarise(Median.Heart.Rate = median(Value))

median.heart.rate.hour.barplot = ggplot(median.heart.rate.hour.summary,
  aes(x = Hour, y = Median.Heart.Rate)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 16)) +
  scale_x_continuous(breaks = seq(0, 23)) + scale_y_continuous(breaks = seq(0, 100, 20)) +
```

```r
  labs(title = "Median Heart Rate by Hour Barchart",
       x = "Hour", y = "Median Heart Rate")

print(median.heart.rate.hour.barplot)
```



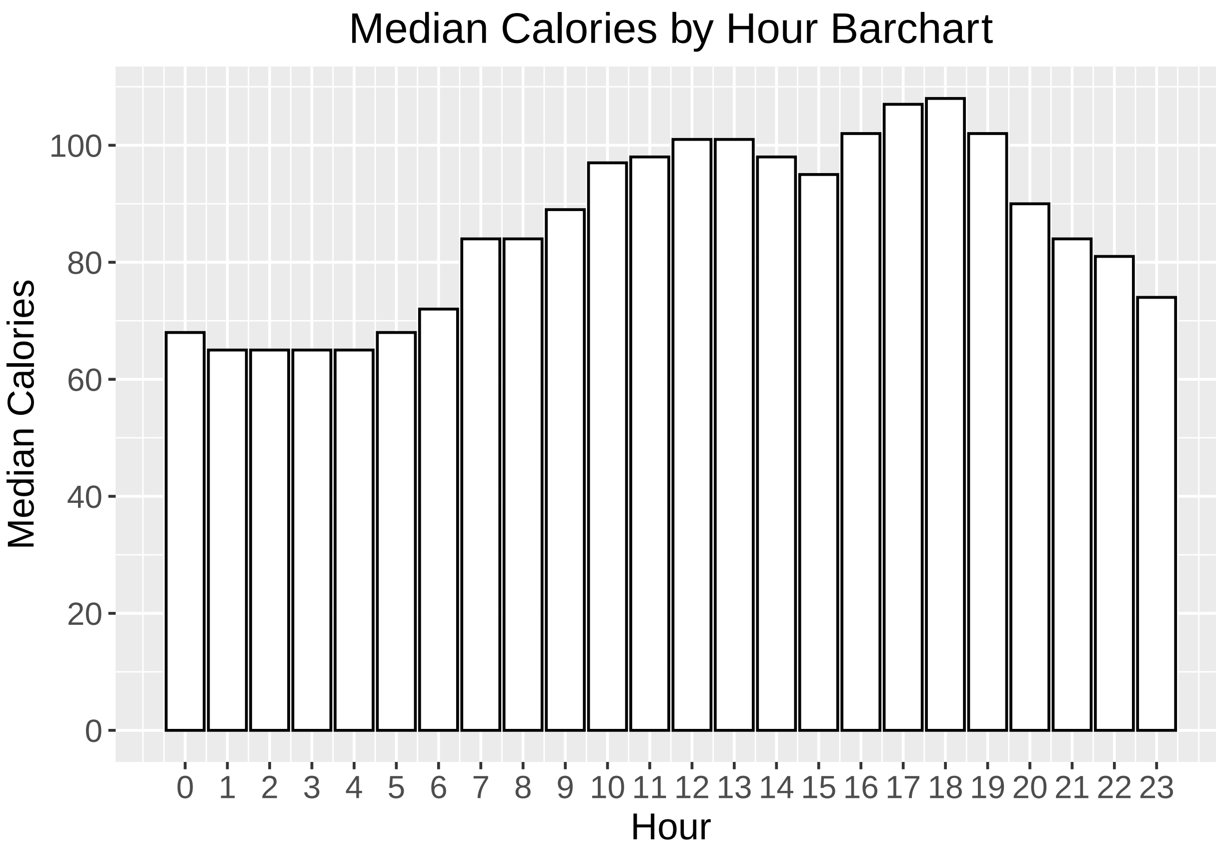

```r
median.heart.rate.day.of.week.summary = heart.rate.data %>%
  group_by(Day.of.Week) %>%
  summarise(Median.Heart.Rate = median(Value))

median.heart.rate.day.of.week.barplot = ggplot(median.heart.rate.day.of.week.summary,
                                               aes(x = Day.of.Week, y = Median.Heart.Rate)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 16)) +
  scale_y_continuous(breaks = seq(0, 100, 20)) +
  labs(title = "Median Heart Rate by Day of Week Barchart",
       x = "Day of Week", y = "Median Heart Rate")

print(median.heart.rate.day.of.week.barplot)
```



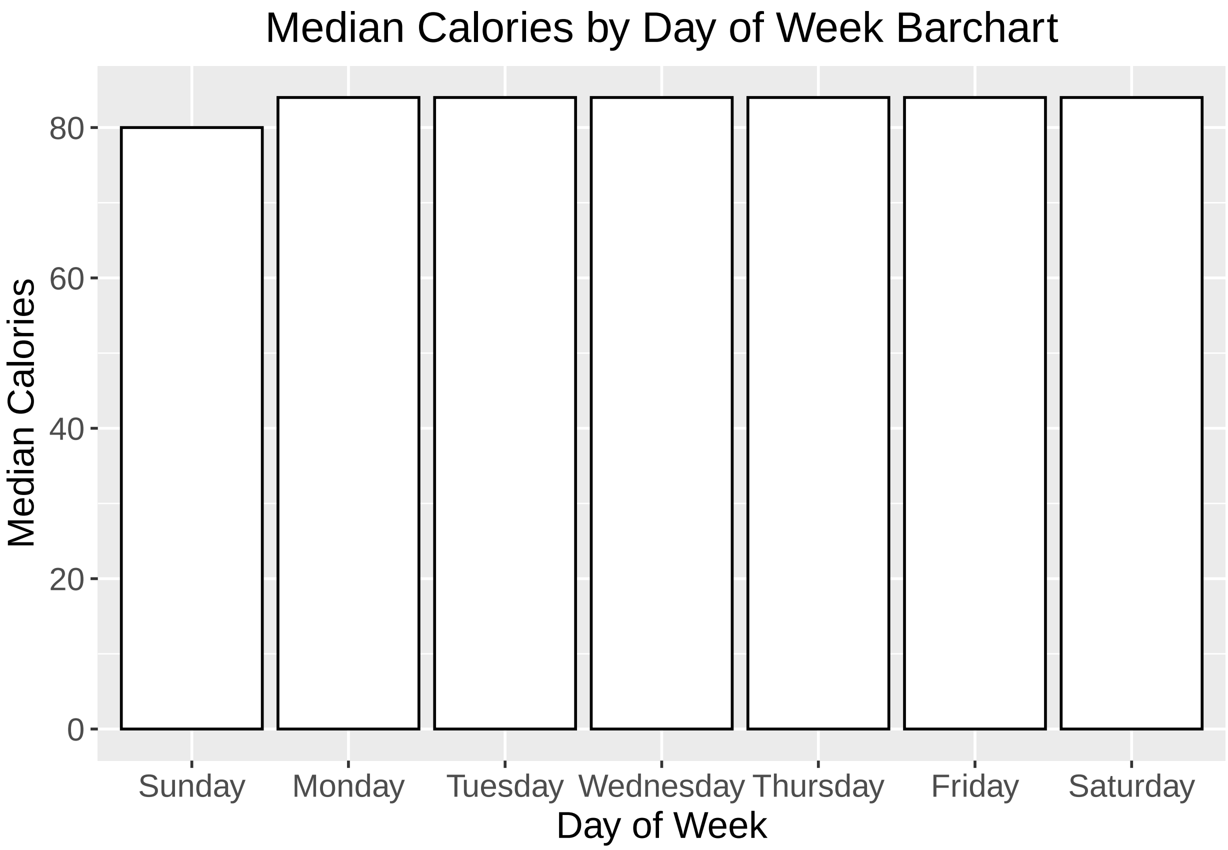
A possible indicator of stress is a high heart rate with a low activity rate (as the elevated heart rate is not being caused by exercise/activity, but stress/anxiety). As such, I performed comparable hour and day-of-the week analyses for calories consumed, intensity, and total steps. As shown below, the highest calories burned, intensity and total step hours correspond to the periods with the highest heart rates (i.e. 6:00pm hour etc.). As such, the elevated heart rates in the 6:00pm hour seem to be correlated with exercise, rather than periods of non-exercise (i.e. stress/anxiety).

```
hour.calories.data$ActivityHour = mdy_hms(hour.calories.data$ActivityHour)

hour.calories.data = hour.calories.data %>%
  rename(Date.Time = ActivityHour) %>%
  mutate(Date = as.Date(Date.Time)) %>%
  mutate(Hour = hour(Date.Time)) %>%
  mutate(Day.of.Week = wday(Date.Time, label = TRUE, abbr = FALSE))

hour.calories.data = filtered.activity.data %>%
  select(Id, Date) %>%
  inner_join(hour.calories.data, by = c("Id", "Date"))

median.calories.hour.summary = hour.calories.data %>%
  group_by(Hour) %>%
  summarise(Median.Calories = median(Calories))

median.calories.hour.barplot = ggplot(median.calories.hour.summary,
  aes(x = Hour, y = Median.Calories)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
```

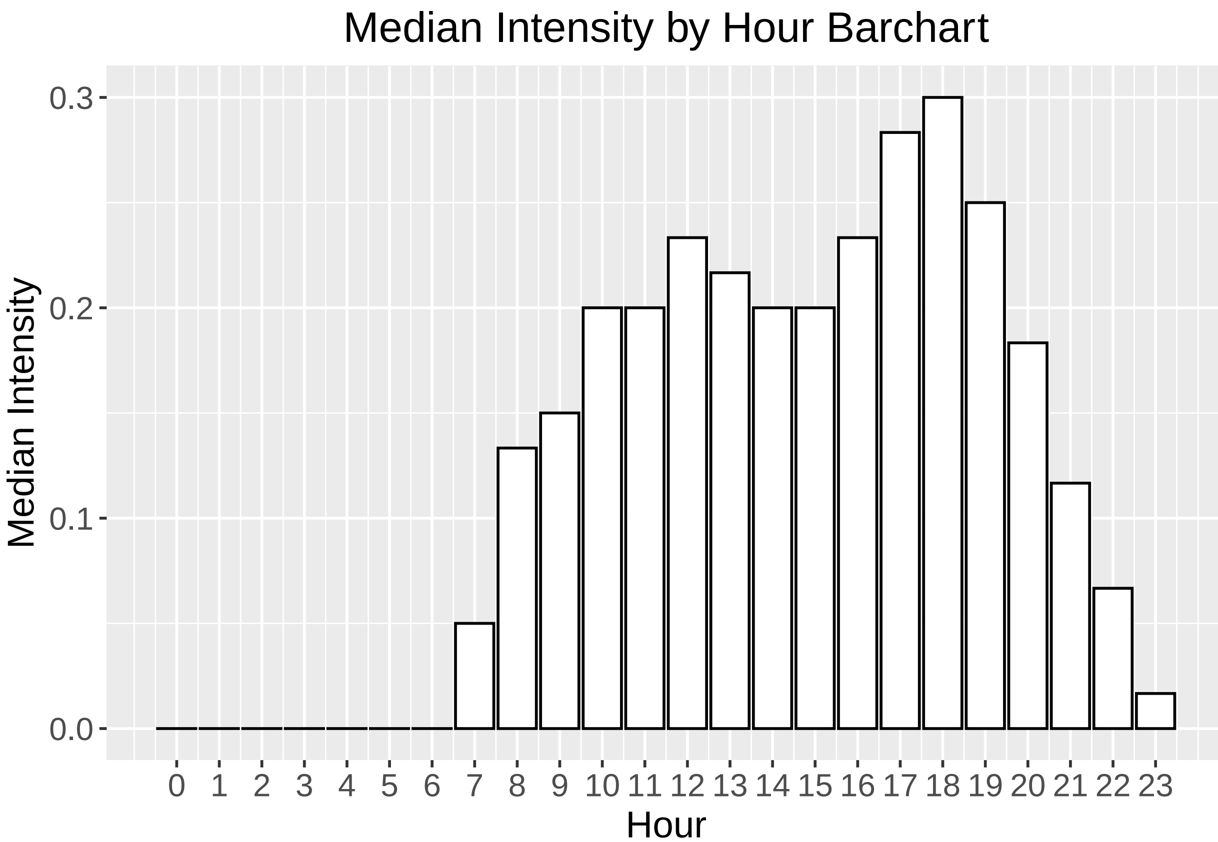
```r
      plot.title = element_text(hjust = 0.5, size = 16)) +
  scale_x_continuous(breaks = seq(0, 23)) +
  labs(title = "Median Calories by Hour Barchart",
      x = "Hour", y = "Median Calories") + scale_y_continuous(breaks = seq(0, 100, 20))

print(median.calories.hour.barplot)
```



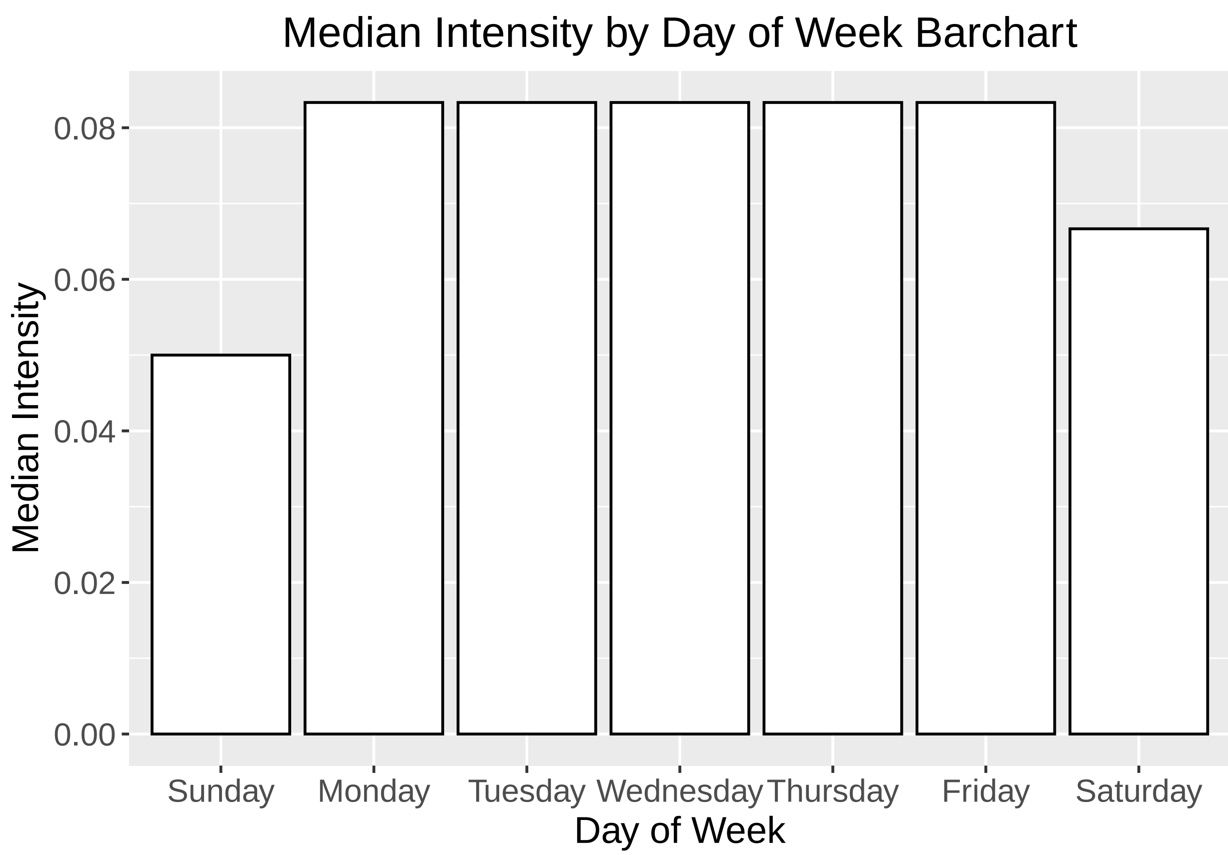

```r
median.calories.day.of.week.summary = hour.calories.data %>%
  group_by(Day.of.Week) %>%
  summarise(Median.Calories = median(Calories))

median.calories.day.of.week.barplot = ggplot(median.calories.day.of.week.summary,
                                             aes(x = Day.of.Week, y = Median.Calories)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
      plot.title = element_text(hjust = 0.5, size = 16)) +
  labs(title = "Median Calories by Day of Week Barchart",
      x = "Day of Week", y = "Median Calories") + scale_y_continuous(breaks = seq(0, 100, 20))

print(median.calories.day.of.week.barplot)
```



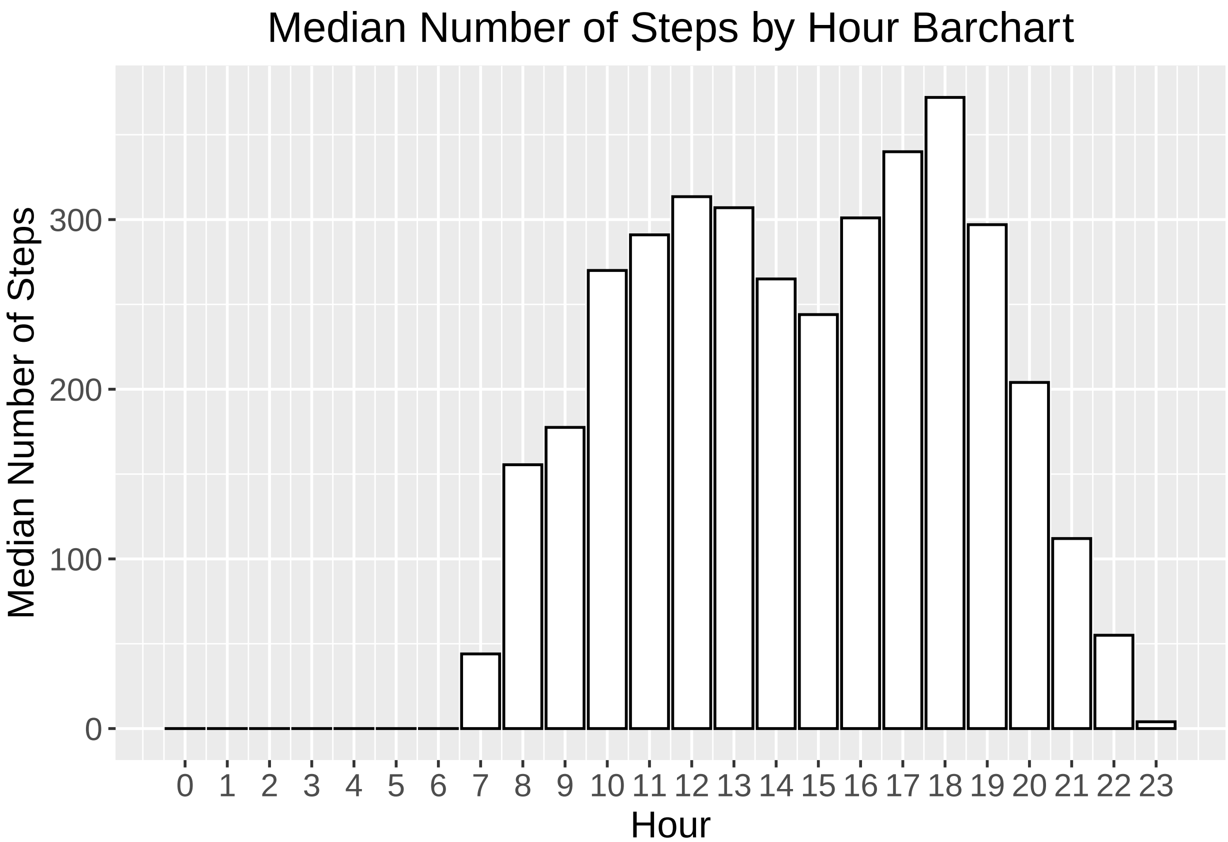

```
hour.intensities.data$ActivityHour = mdy_hms(hour.intensities.data$ActivityHour)

hour.intensities.data = hour.intensities.data %>%
  rename(Date.Time = ActivityHour) %>%
  mutate(Date = as.Date(Date.Time)) %>%
  mutate(Hour = hour(Date.Time)) %>%
  mutate(Day.of.Week = wday(Date.Time, label = TRUE, abbr = FALSE))

hour.intensities.data = filtered.activity.data %>%
  select(Id, Date) %>%
  inner_join(hour.intensities.data, by = c("Id", "Date"))

median.intensities.hour.summary = hour.intensities.data %>%
  group_by(Hour) %>%
  summarise(Median.Intensity = median(AverageIntensity))

median.intensities.hour.barplot = ggplot(median.intensities.hour.summary,
                                         aes(x = Hour, y = Median.Intensity)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16)) +
  scale_x_continuous(breaks = seq(0, 23)) +
  labs(title = "Median Intensity by Hour Barchart",
       x = "Hour", y = "Median Intensity")

print(median.intensities.hour.barplot)
```



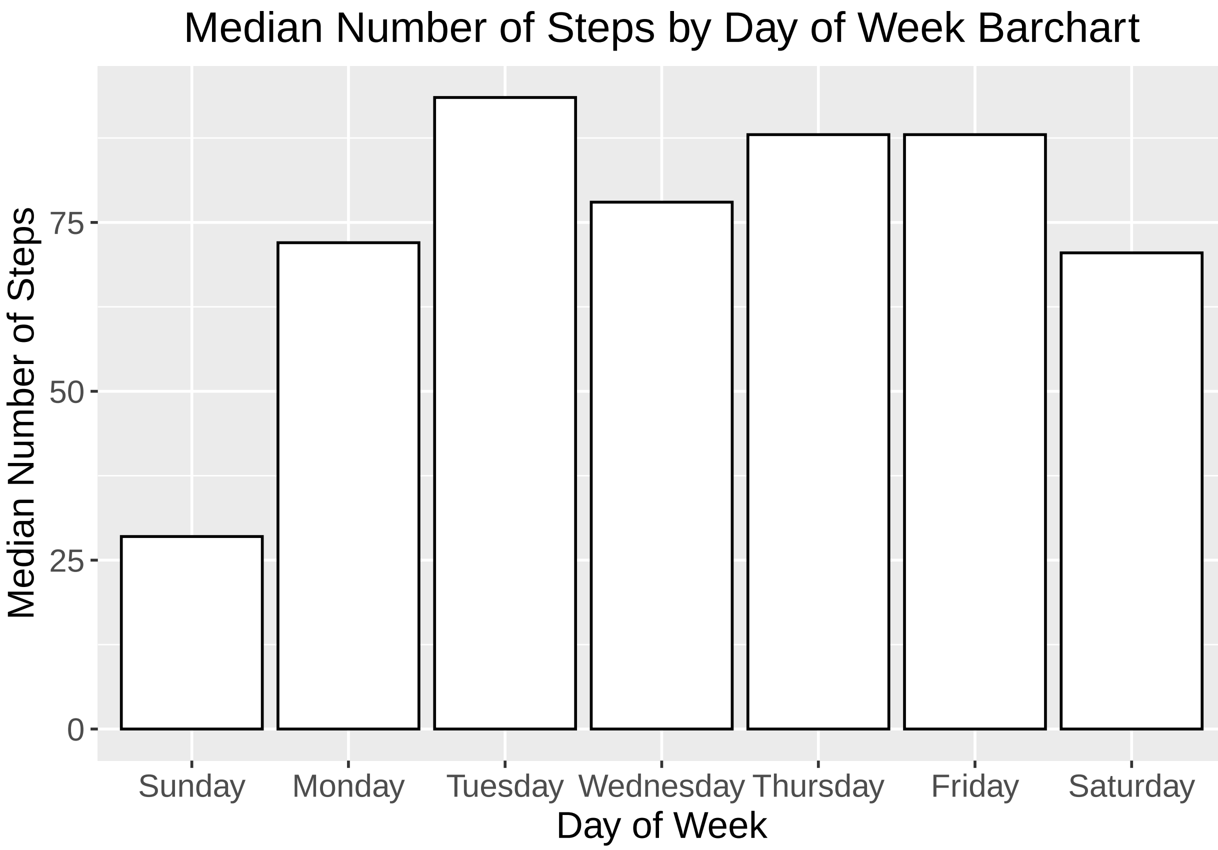

```
median.intensities.day.of.week.summary = hour.intensities.data %>%
  group_by(Day.of.Week) %>%
  summarise(Median.Intensity = median(AverageIntensity))

median.intensities.day.of.week.barplot = ggplot(median.intensities.day.of.week.summary,
                                                aes(x = Day.of.Week, y = Median.Intensity)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 16)) +
  labs(title = "Median Intensity by Day of Week Barchart",
        x = "Day of Week", y = "Median Intensity")

print(median.intensities.day.of.week.barplot)
```




```
hour.steps.data$ActivityHour = mdy_hms(hour.steps.data$ActivityHour)

hour.steps.data = hour.steps.data %>%
  rename(Date.Time = ActivityHour) %>%
  mutate(Date = as.Date(Date.Time)) %>%
  mutate(Hour = hour(Date.Time)) %>%
  mutate(Day.of.Week = wday(Date.Time, label = TRUE, abbr = FALSE))

hour.steps.data = filtered.activity.data %>%
  select(Id, Date) %>%
  inner_join(hour.steps.data, by = c("Id", "Date"))

median.steps.hour.summary = hour.steps.data %>%
  group_by(Hour) %>%
  summarise(Median.Steps = median(StepTotal))

median.steps.hour.barplot = ggplot(median.steps.hour.summary,
  aes(x = Hour, y = Median.Steps)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
       plot.title = element_text(hjust = 0.5, size = 16)) +
  scale_x_continuous(breaks = seq(0, 23)) +
  labs(title = "Median Number of Steps by Hour Barchart",
       x = "Hour", y = "Median Number of Steps")

print(median.steps.hour.barplot)
```




```
median.steps.day.of.week.summary = hour.steps.data %>%
  group_by(Day.of.Week) %>%
  summarise(Median.Steps = median(StepTotal))

median.steps.day.of.week.barplot = ggplot(median.steps.day.of.week.summary,
                                          aes(x = Day.of.Week, y = Median.Steps)) +
  geom_bar(stat = "identity", fill = "white", colour = "black") +
  theme(axis.text = element_text(size = 12), axis.title = element_text(size = 14),
      plot.title = element_text(hjust = 0.5, size = 16)) +
  labs(title = "Median Number of Steps by Day of Week Barchart",
      x = "Day of Week", y = "Median Number of Steps")

print(median.steps.day.of.week.barplot)
```



Part 5: Share

In summarizing my findings above, the Fitbit cohort:

- is mostly sedentary (low activity)
- has comparable calorie and sleep trends to the broader population
- has some evidence of sleeplessness, but most of the cohort does not appear to suffer from sleeplessness
- has wider variation in sleep patterns on weekends vs weekdays
- has higher heart rates connected to higher levels of activity during non-work hours, and the lowest heart rates and activity levels during typical sleep hours.

Part 6: Act

Based on the summary above, I would advise Bellabeat to compare their user cohort to this Fitbit cohort (particularly the lower levels of activity in the Fitbit cohort) to see if the conclusions drawn from this Fitbit cohort apply to their Bellabeat userbase. While there was not widespread sleeplessness in the Fitbit cohort, it does appear that sleeplessness is a problem for some; as such, Bellabeat could help their users with sleeplessness by identifying possible periods of restlessness. Finally, based on the higher variation in sleep on the weekends, Bellabeat could use its products to help its users ensure that they are getting enough rest on the weekends, while helping them balance sleep with their other plans.