Fitbit Data Project

Load Required Packages

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

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

```
library(dplyr)
library(gghighlight)
dailyActivity_merged <- read.csv(file = "C:\\Users\\Wes\\Desktop\\FitBit Data\\dailyActivity_merged.csv
```

Cleaning Data

Start by looking data points where there are missing values. Since it appears that the most common use of the fitness tracker is to track steps, we will begin there.

```
dailyActivity_merged %>%
 filter(TotalSteps == 0) %>%
 view
```

There 77 of the 940 observation points have to data at all, we will remove these entries.

```
dailyActivity <- dailyActivity_merged %>%
  filter(TotalSteps != 0)
```

Additionally we see that of the remaining users, some have very few observations. These users are not likely to provide insight for our purpose

```
dailyActivity %>%
  count(Id) %>% view
```

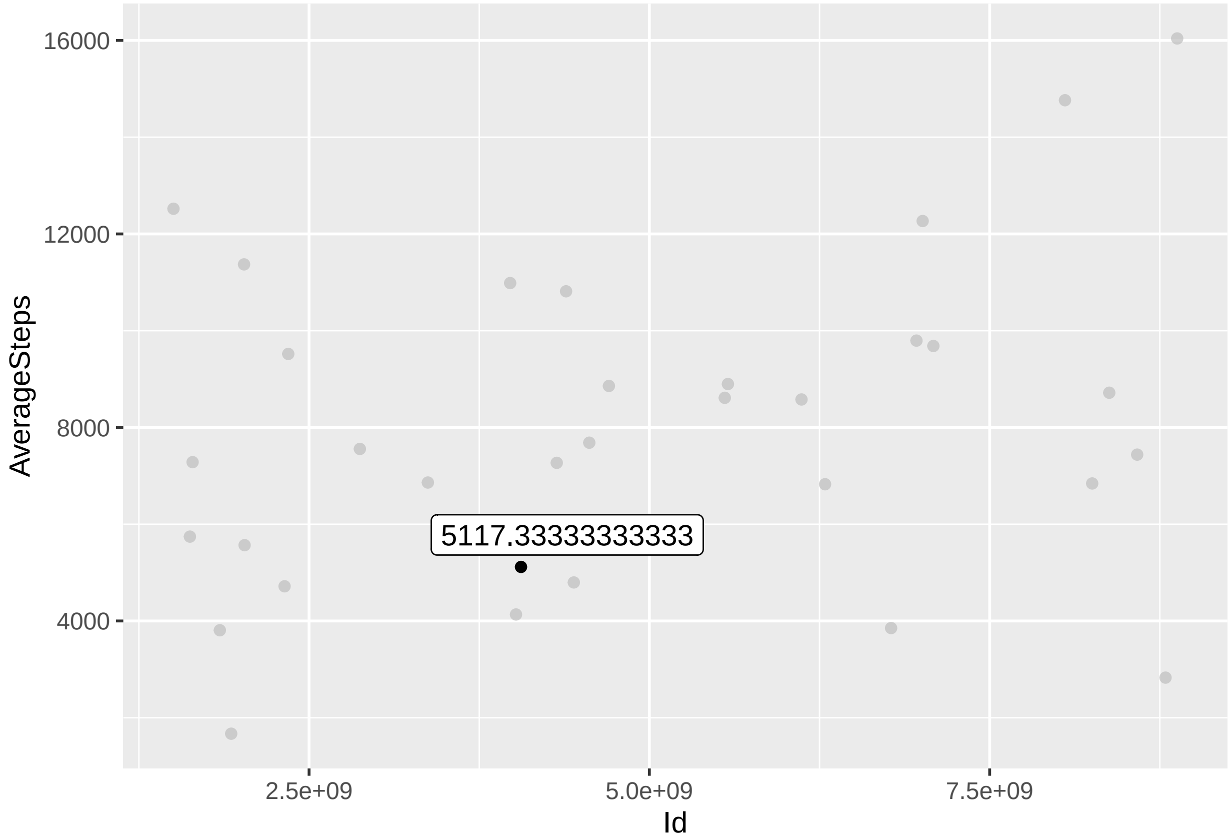
Pulling up the particular user with only 3 observations, we see that the user only used the device for 3 days, and then stopped. Of those 3 days, the user was not particularly active when compared to other users.

```
dailyActivity %>%
  filter( Id == 4057192912) %>%
  view
```

We can then further confirm this by creating a new table for average steps based on user Id, and then graphing that data against the rest of the observations, highlighting the user in question.

```
 AverageSteps <- aggregate(TotalSteps ~ Id, dailyActivity, mean)
AverageSteps <- rename(AverageSteps, "AverageSteps" = "TotalSteps")

AverageSteps %>%
  ggplot(mapping=aes(x=Id, y=AverageSteps)) + geom_point() + gghighlight(Id == 4057192912, label_key = A
```



As a result, I opted to drop this user from the data set for the time being. It may be worthwhile to return to these "low activity" users to see if we can glean insight as to for what purpose they use the trackers. I then checked my new table to ensure that the clean was successful.

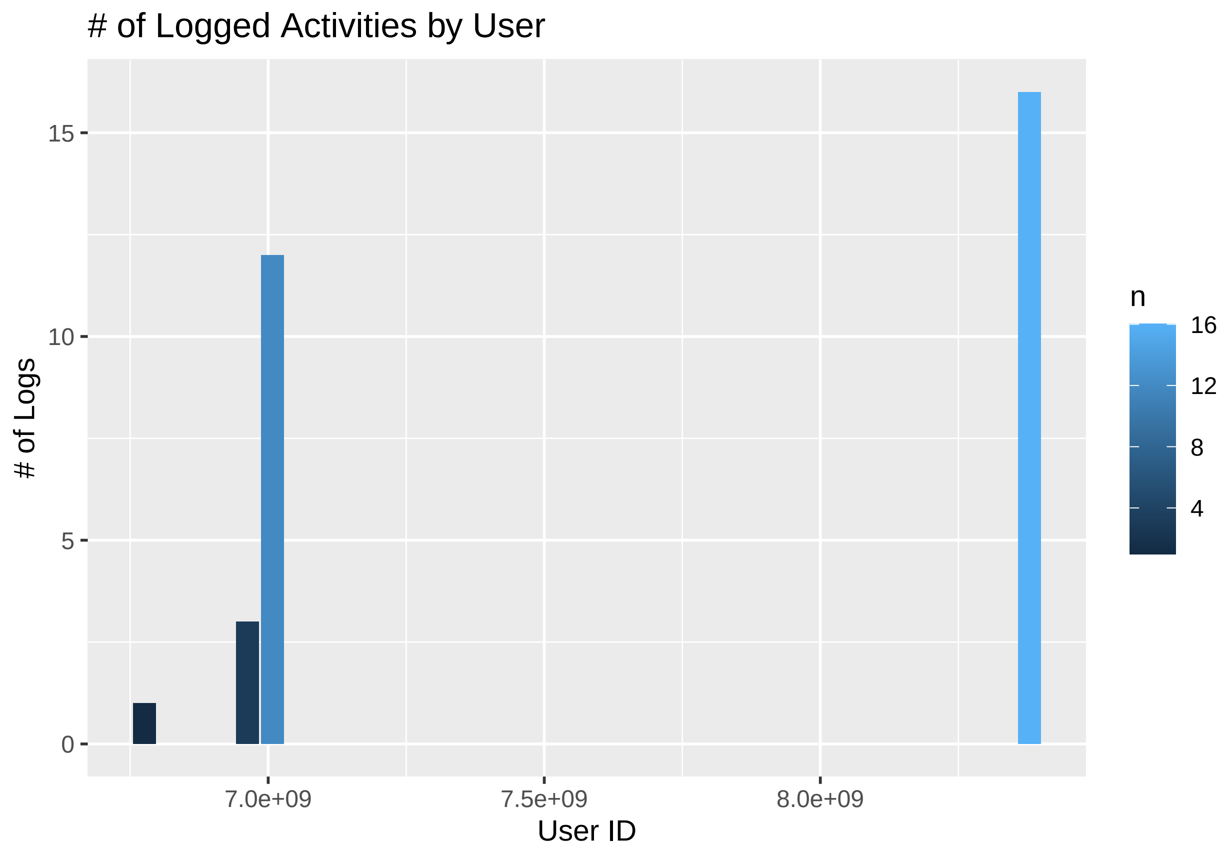
```
dailyActivity_clean <- dailyActivity %>%
  filter(Id != 4057192912)

dailyActivity_clean %>%
  count(Id) %>%
  view
```

Insights

Activity Logs We can first look at what is and is not being tracked. If we examine the user's logged activities, we find that logging activities is not something that the users do often. Of the 32 remaining users, only 4 have logged even a single activity, and only 2 have logged more than 3 activities.

```
dailyActivity %>%
  filter(LoggedActivitiesDistance > 0)  %>%
  count(Id) %>%
  ggplot(mapping=aes(x=Id, y=n, fill=n)) + geom_col() +
  ggtitle("# of Logged Activities by User") + xlab("User ID") + ylab("# of Logs")
```



Given the lack of utilization of the "Logging Activity" feature, future investments would probably be best spent elsewhere.

Active Times By using a column graph we can identify the most active times of day, as well as well as potential 'workout times' for our users.

```
hourlySteps_merged<- read.csv(file = "C:\\Users\\Wes\\Desktop\\FitBit Data\\hourlySteps_merged.csv")
```

First we must mutate the Activity Hour column in order to be useful. We will need the lubridate package for this.

```
library(lubridate)
```

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

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

```
hourlySteps_merged <- hourlySteps_merged %>% mutate(ActivityHour = mdy_hms(ActivityHour))
hourlySteps_merged$Date <- as.Date(hourlySteps_merged$ActivityHour)
hourlySteps_merged$Time <- format(as.POSIXct(hourlySteps_merged$ActivityHour), format="%H:%M:%S")
```

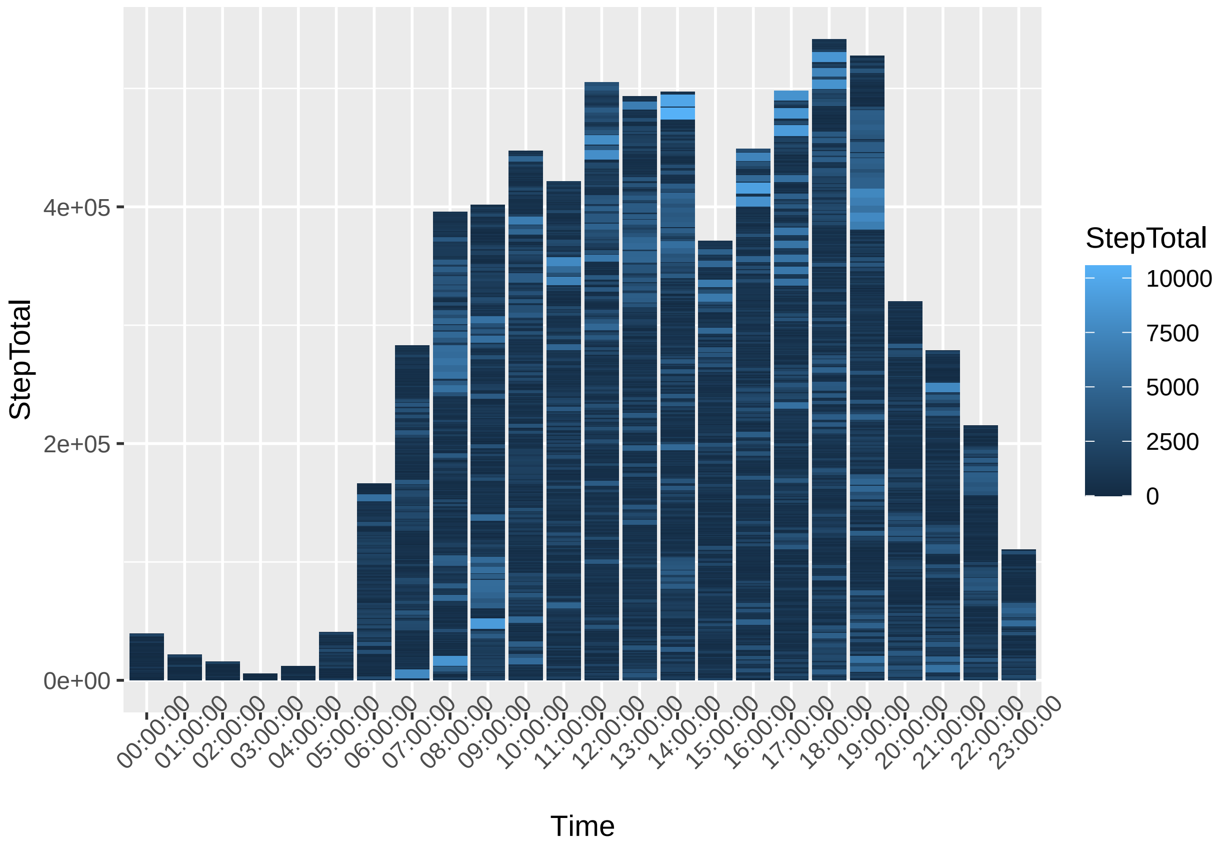
Again we see that user Id '4057192912' lacks datapoints and can be removed

```
hourlySteps_merged %>%
  count(Id) %>% view

hourlySteps_merged <- hourlySteps_merged %>%
  filter(Id != 4057192912)
```

Now we can plot

```
hourlySteps_merged %>%
  ggplot(mapping=aes(x=Time, y=StepTotal, fill=StepTotal)) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```

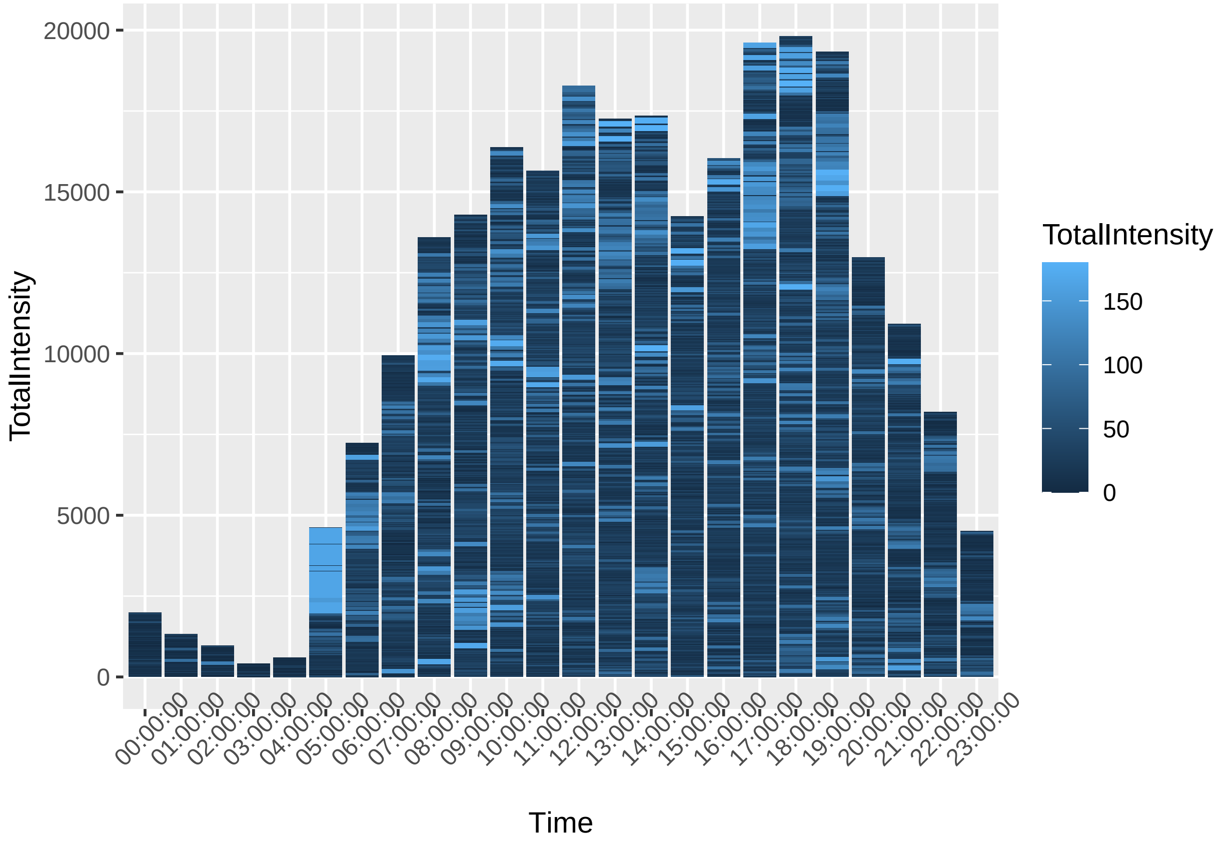


The chart shows us, not only that are the evening hours and lunch hours the busiest, which is to be expected, but also by identifying step totals by user during each hour, with lighter shades of blue being more steps by a single user in a given period. This tells us that these times are likely times in which users will be exercising.

We can confirm this by looking at the intensities tracked throughout the day, specifically highlighting higher intensity readings.

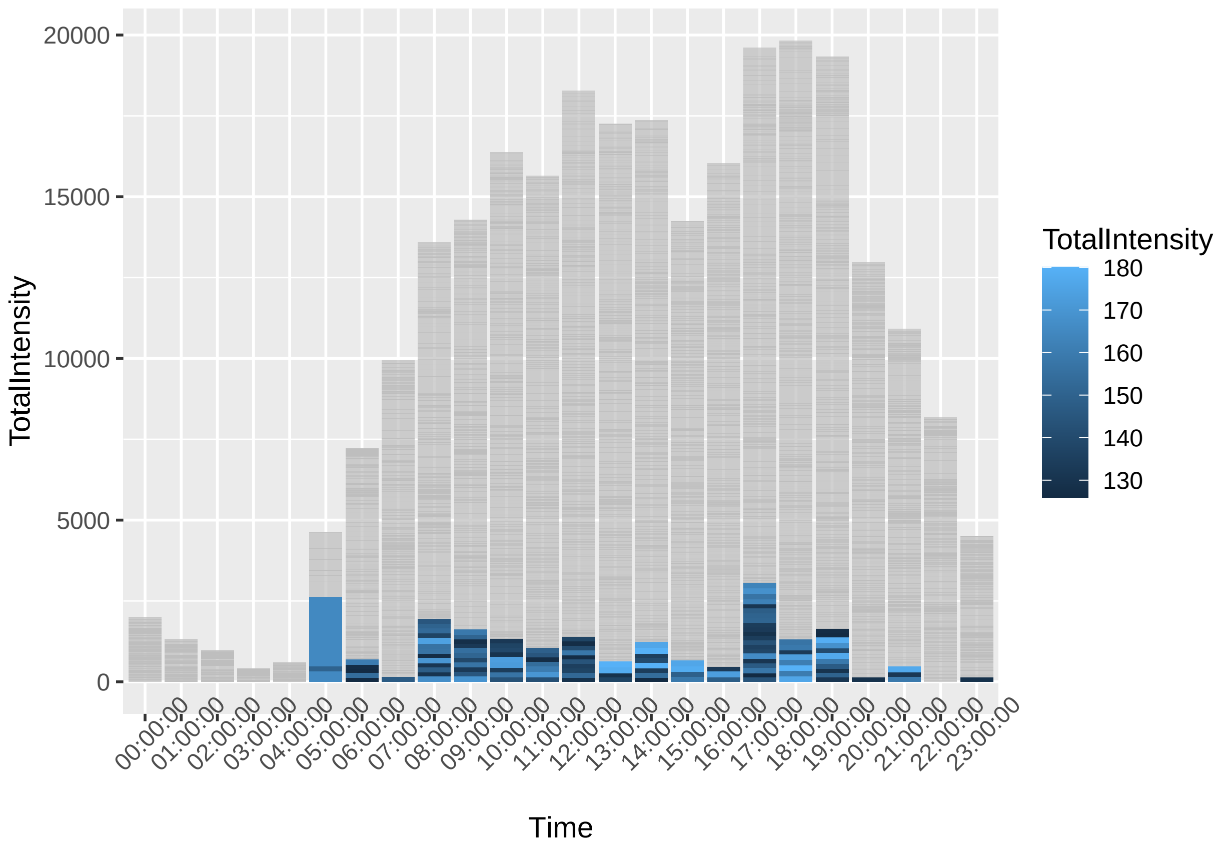
```
hourlyIntensities_merged<- read.csv(file = "C:\\Users\\Wes\\Desktop\\FitBit Data\\hourlyIntensities_merg
```

```
hourlyIntensities_merged <- hourlyIntensities_merged %>%
  filter(Id != 4057192912)
hourlyIntensities_merged <- hourlyIntensities_merged %>%  mutate(ActivityHour = mdy_hms(ActivityHour))
hourlyIntensities_merged$Time <- format(as.POSIXct(hourlyIntensities_merged$ActivityHour), format ="%H:%
hourlyIntensities_merged %>%
  ggplot(mapping=aes(x=Time, y=TotalIntensity, fill=TotalIntensity )) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```



```
hourlyIntensities_merged %>%
  ggplot(mapping=aes(x=Time, y=TotalIntensity, fill=TotalIntensity )) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45)) + gghighlight(TotalIntensity>125)
```

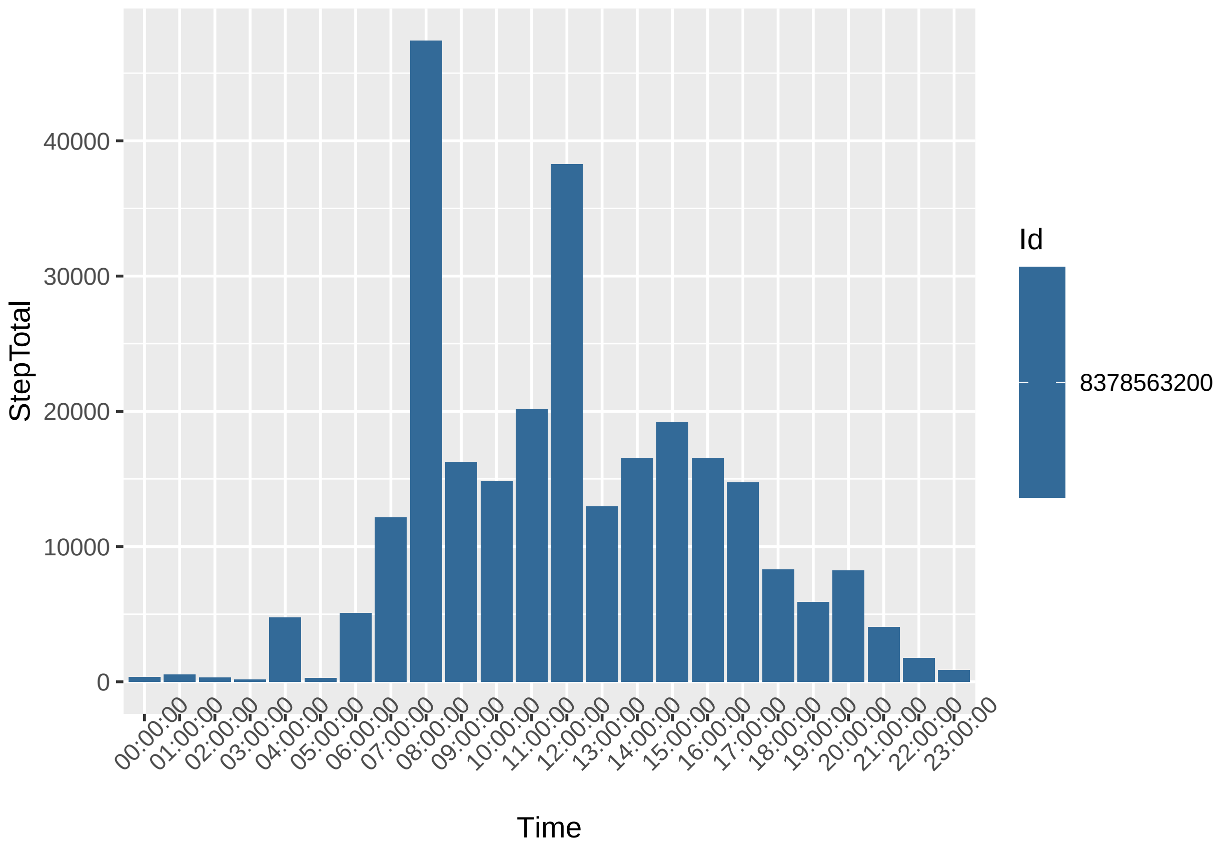
```
## Warning: Tried to calculate with group_by(), but the calculation failed.
## Falling back to ungrouped filter operation...
```



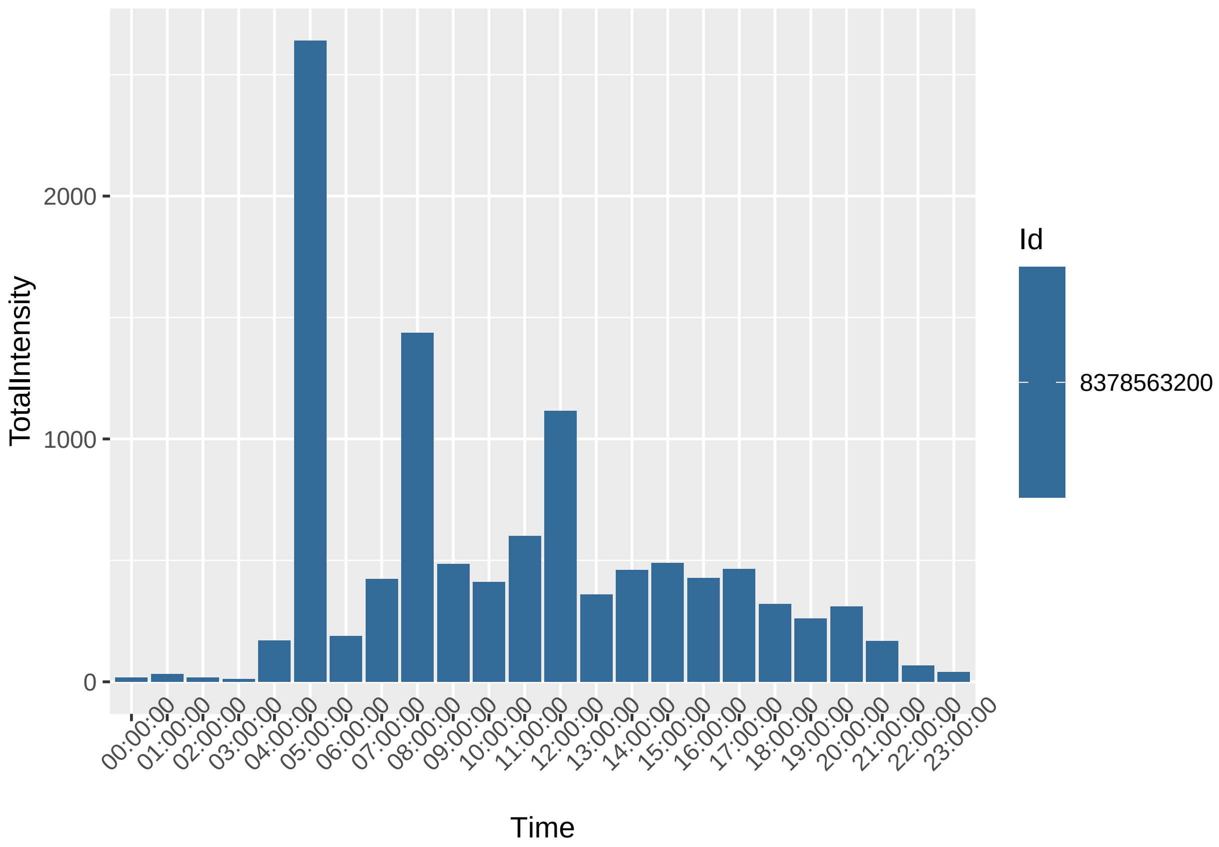
We see that intensity, like step count, elevates during the morning, stays relatively consistent through the day, with a small drop, and then spikes again around 5pm, as individuals get off of work, and either get home and go for a walk or run, or go to the gym on their way home from work.

By graphing for only those users that Log Activities with regularity, we can gain insight into what they may be logging. We can compare the charts for hourlySteps and hourlyIntensities to find differences.

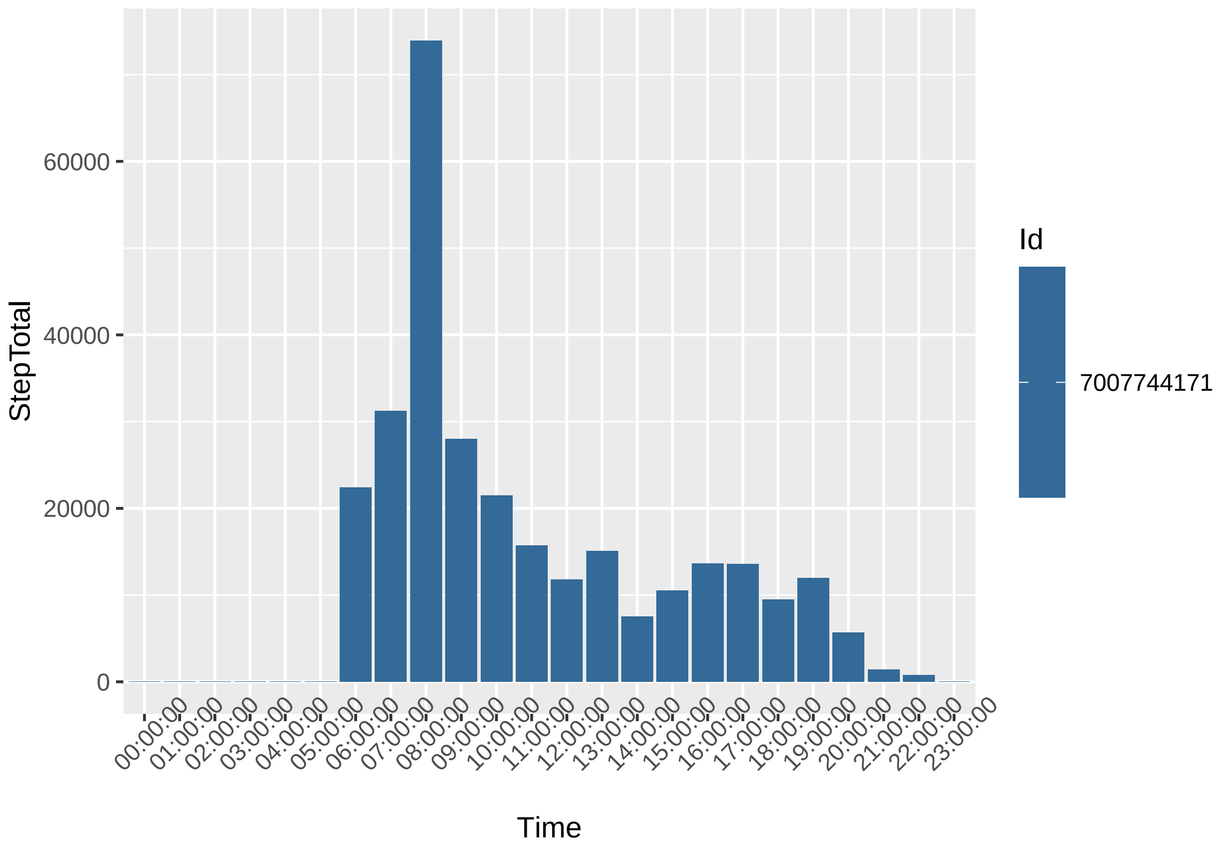
```r
hourlySteps_merged %>%
  filter(Id == 8378563200) %>%
  ggplot(mapping=aes(x=Time, y=StepTotal, fill=Id)) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```



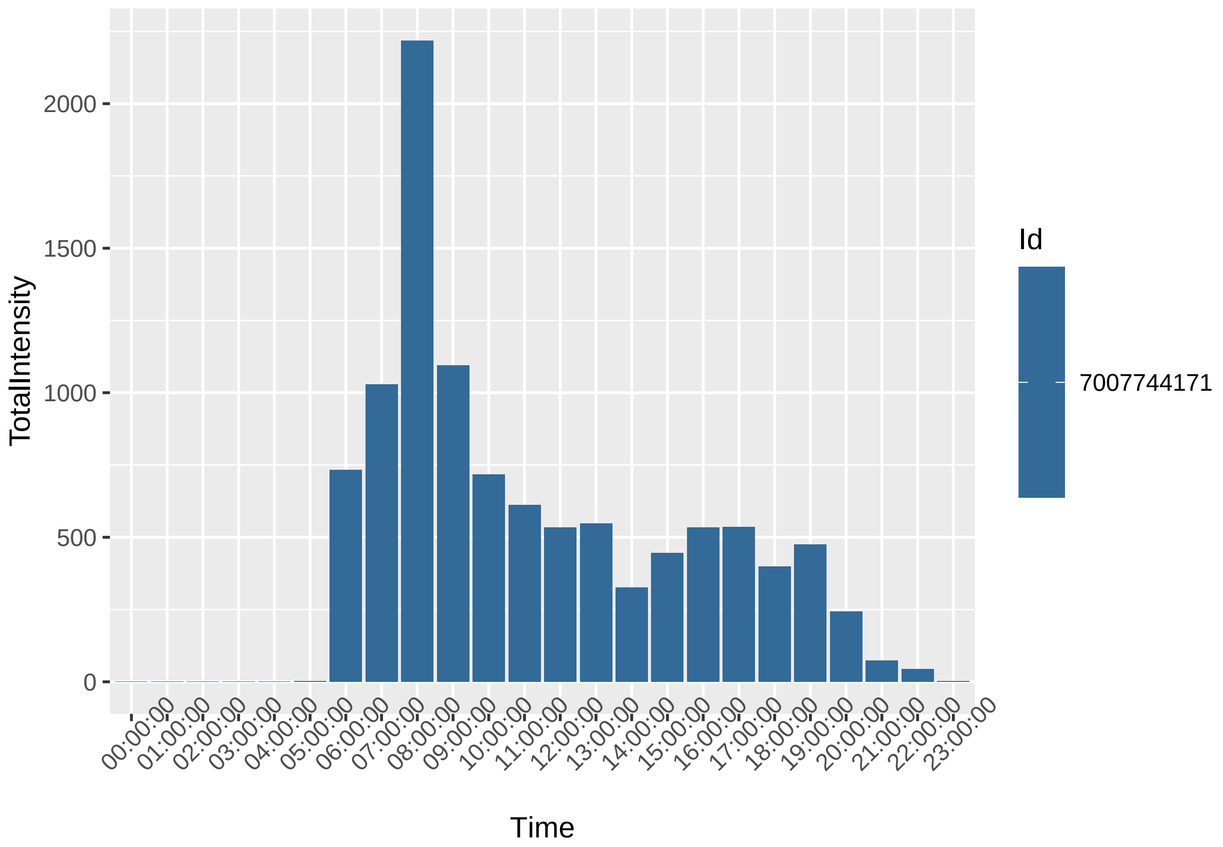
```
hourlyIntensities_merged %>%
  filter(Id == 8378563200) %>%
  ggplot(mapping=aes(x=Time, y=TotalIntensity, fill=Id )) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```



```
hourlySteps_merged %>%
  filter(Id == 7007744171) %>%
  ggplot(mapping=aes(x=Time, y=StepTotal, fill=Id)) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```



```
hourlyIntensities_merged %>%
  filter(Id == 7007744171) %>%
  ggplot(mapping=aes(x=Time, y=TotalIntensity, fill=Id )) + geom_col(size = 2) +
  theme(axis.text.x = element_text(angle=45))
```



From this we can conclude that user 8378563200 is involved in workout at 5am which requires no steps, however is very intense. Perhaps a bike ride before rush hour traffic. Further research may show us that if the company would like to increase the usage of "LoggedActivities" then focus should likely be on activity and exercise that does not show up as "steps" or arm movements.