RecCen2019 EDA

James Cha

2024-02-14

Pulling in Data

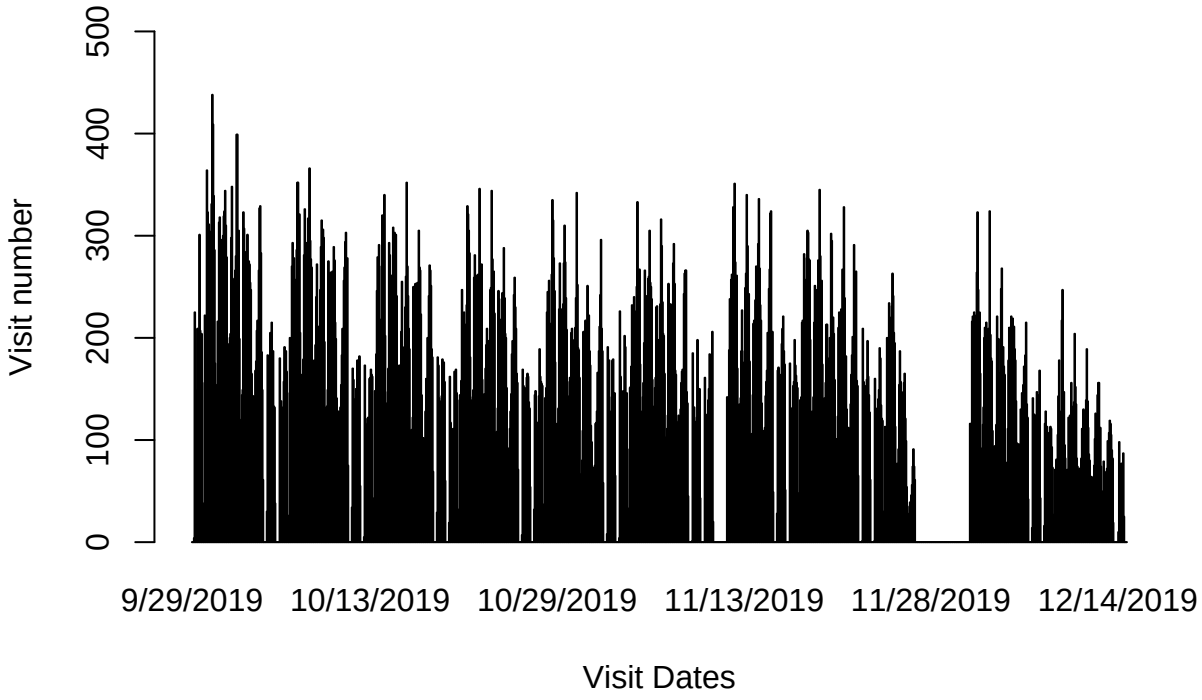
```
# File Locations will differ based on computer

reccen2019 <- read.csv("reccen2019.csv")
head(reccen2019)
```

```
##        Date     Time Visits    Day
## 1 9/29/2019  5:00 AM      0 Sunday
## 2 9/29/2019  6:00 AM      0 Sunday
## 3 9/29/2019  7:00 AM      0 Sunday
## 4 9/29/2019  8:00 AM      4 Sunday
## 5 9/29/2019  9:00 AM    225 Sunday
## 6 9/29/2019 10:00 AM    149 Sunday
```

First, we should look at all data available to guage the discrepancies between the dates and the time. It seems like there is no missing data (it always runs from 5:00 AM to 10:00PM, which is nice. On closed days, it will just display 0 Visits for the day. This may interfere with the analysis and ML training, so we will have to think a bit about this.)

Making sliders of preferred time in our website may be beneficial. Nobody really wants to go at 5:00AM even if the visits are nearly zero constantly.

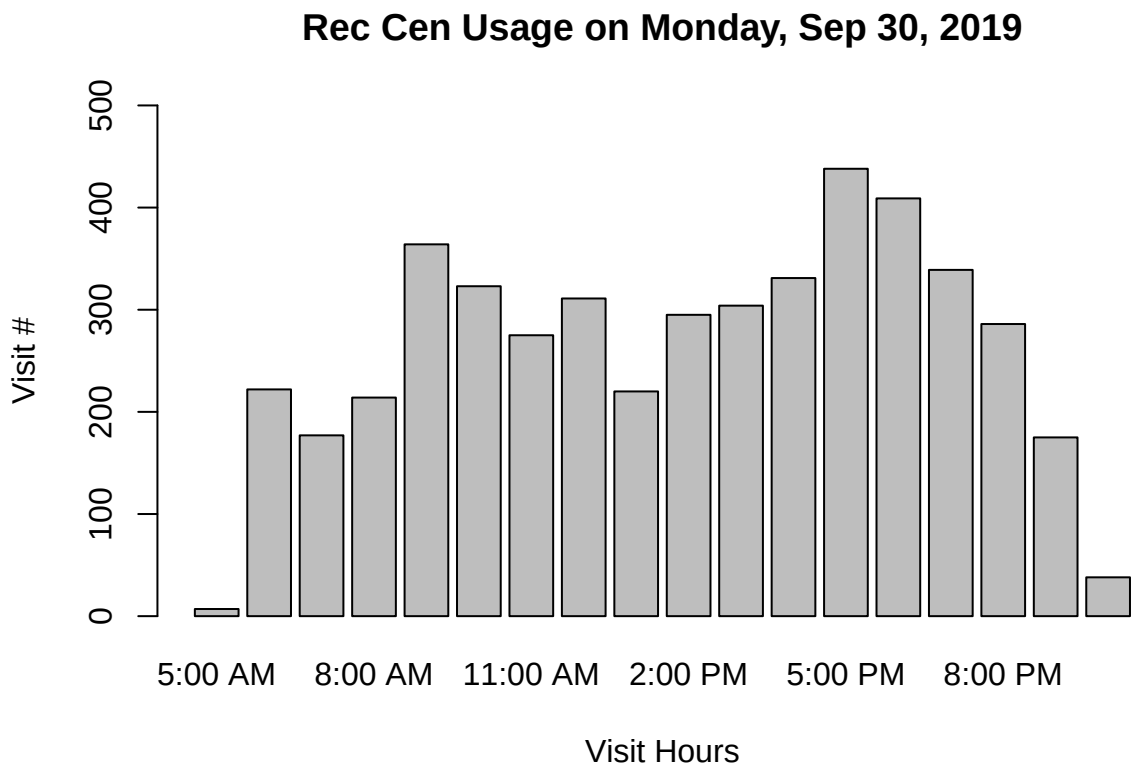
```
barplot(reccen2019$Visits, xlab="Visit Dates", ylab="Visit number",
        names.arg=reccen2019$Date, ylim=c(0,500))
```



This is a bit hard to see, but we can see that there is a missing period near the end of the three month cycle.

Let's take a look at an individual day. Say, 9/30/2019, which seems to have a high volume of people (and hence more data points)

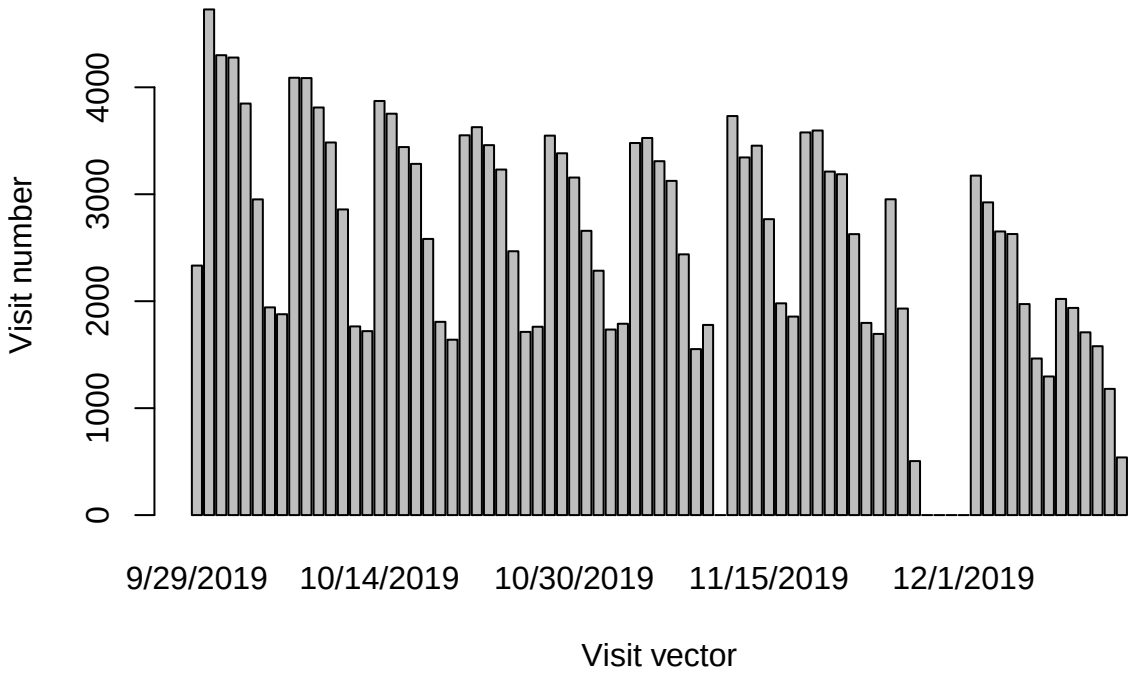
```
sep30 <- reccen2019[which(reccen2019$Date == "9/30/2019"),]

barplot(sep30$Visits, xlab="Visit Hours", ylab="Visit #",
        names.arg=sep30$Time, ylim=c(0,500), main="Rec Cen Usage on Monday, Sep 30, 2019")
```



Visit peaks at 5:00 PM, right before dinner. Usage seems to have peaked due to it being start of the school year, and first day of the week.

Let us now try and reduce the granularity of the data (reducing the level of detail) to see if we can't uncover a more general trend. We will aggregate all of the hourly data into daily data and plot them. This should potentially give us a cyclical, weekly trend.

```
daily_data <- aggregate(reccen2019$Visits, by=list(reccen2019$Date), FUN=sum)
daily_data <- daily_data[order(as.Date(daily_data$Group.1, format="%m/%d/%Y")),]

barplot(daily_data$x, xlab="Visit vector", ylab="Visit number",
        names.arg=daily_data$Group.1)
```

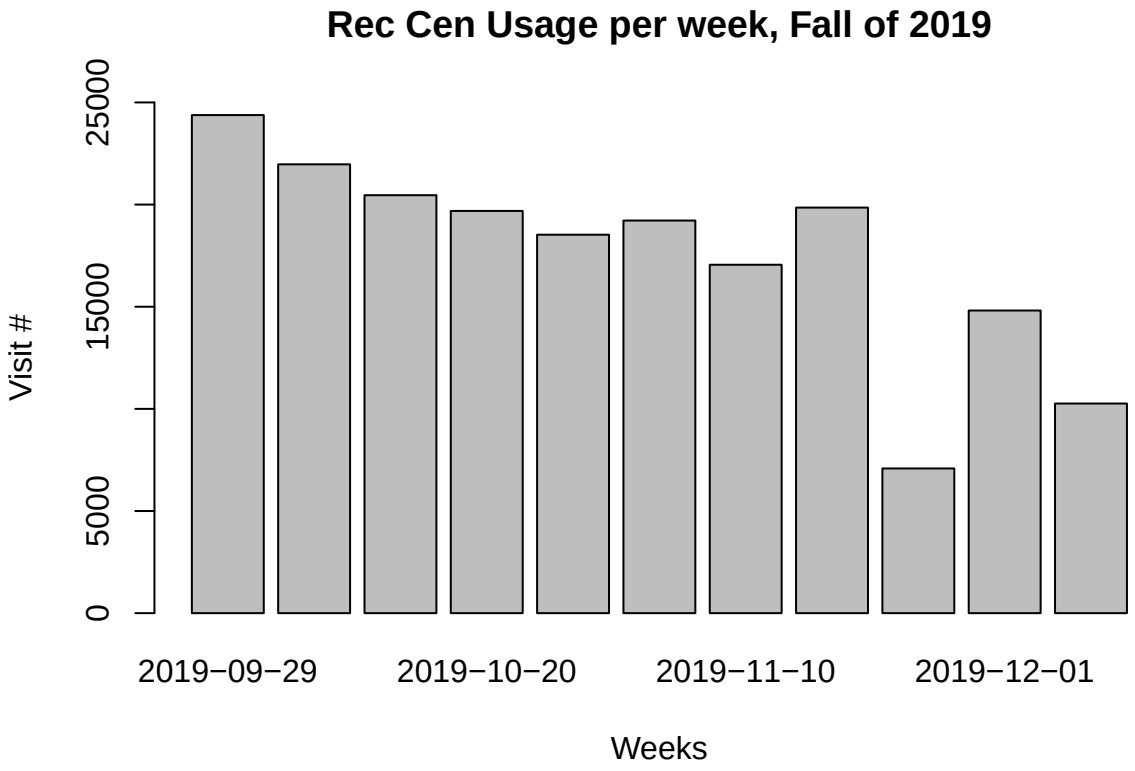


Seasonality defintely exists. It seems like it picks up during the weekdays versus weekends, which I find surprising; I thought many people would be more inclined to go to the gym since you would be less busy. However, there is an explosion of people going to the gym on Mondays, nearly double the amount of weekends. We should probably avoid going to the gym on Mondays, now.

There also seems to be a few missing days before 12/3/2019. Checking the academic calendar of 2019, that is near Thanksgiving. It picks right back up near end of instruction (December 6th).

We can opt to cluster the data into a week as well. We will not do an analysis on clusters of a month, since we only have three months of data to work with. If used on a larger data set, though, clusters of a month may uncover seasonality over the year.

```
# Convert the date column to Date type
daily_data$Group.1 <- as.Date(daily_data$Group.1, format = "%m/%d/%Y")

# Create a week column
daily_data$week <- floor_date(daily_data$Group.1, "week")

# Aggregate data by week
weekly_data <- daily_data %>%
  group_by(week) %>%
  summarise(weekly_sum = sum(x, na.rm = TRUE))

barplot(weekly_data$weekly_sum, xlab="Weeks", ylab="Visit #",
        names.arg=weekly_data$week, main="Rec Cen Usage per week, Fall of 2019",
        ylim=c(0,25000))
```



Not much can be uncovered here, other than an overall decreasing trend (probably due to less and less people visiting as they lose the will to go to the gym)

Otherwise, we have uncovered what seems to be an alright amount of observations.