Introduction

This EDA looks at crime incident reports in the city of Boston from June 2015 to September 2018. I use Folium for plotting an interactive heatmap of Boston, and seaborn for everything else.

The data is originally provided by Boston's open data hub, **Analyze Boston (https://data.boston.gov/dataset/crime-incident-reports-august-2015-to-date-source-new-system)**. This kernel by helped me get started with this dataset, and this other **Kernel (https://www.kaggle.com/daveianhickey/how-to-folium-for-maps-heatmaps-time-analysis)** by Dave Fisher-Hickey helped me get started with Folium.

In [1]:

```
import pandas as pd
import numpy as np
import pandas_profiling
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
import os
from wordcloud import WordCloud ,STOPWORDS, ImageColorGenerator
import seaborn as sns
%matplotlib inline
import folium
from folium.plugins import HeatMap
p = "YlGnBu"
p2 = "YlGn"
p3 = "Greys"
p4="viridis"
p5="coolwarm"
```

In [2]:

```
data=pd.read_csv('crime.csv', encoding='iso-8859-1')
data
```

Out[2]:

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPT
0	I182070945	619	Larceny	LARCENY ALL OTH
1	I182070943	1402	Vandalism	VANDAL
2	I182070941	3410	Towed	TOWED MO VEHI
3	I182070940	3114	Investigate Property	INVESTIG PROPE
4	I182070938	3114	Investigate Property	INVESTIG PROPE
5	I182070936	3820	Motor Vehicle Accident Response	M/V ACCID INVOLVING PEDESTF - INJ
6	I182070933	724	Auto Theft	AUTO TH
7	I182070932	3301	Verbal Disputes	VERBAL DISP
8	I182070931	301	Robbery	ROBBERY - STR
9	I182070929	3301	Verbal Disputes	VERBAL DISP
10	I182070928	3301	Verbal Disputes	VERBAL DISP
11	I182070927	3114	Investigate Property	INVESTIG PROPE
12	I182070923	3108	Fire Related Reports	FIRE REPORT - HOU BUILDING, E
13	I182070922	2647	Other	THREATS TO DO BOI HA
14	I182070921	3201	Property Lost	PROPERTY - L
15	I182070920	3006	Medical Assistance	SICK/INJURED/MEDI - PERS
16	I182070919	3301	Verbal Disputes	VERBAL DISP
17	I182070918	3305	Assembly or Gathering Violations	DEMONSTRATIONS/F
18	I182070917	2647	Other	THREATS TO DO BOI HA
19	I182070915	614	Larceny From Motor Vehicle	LARCENY THEFT FF MV - NON-ACCESS(
20	I182070913	3006	Medical Assistance	SICK/INJURED/MEDI - PERS
21	I182070911	3801	Motor Vehicle Accident Response	M/V ACCIDENT - OT
22	I182070910	3006	Medical Assistance	SICK/INJURED/MEDI - PERS

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPT
23	I182070909	3803	Motor Vehicle Accident Response	M/V ACCIDE PERSONAL INJ
24	I182070908	522	Residential Burglary	BURGLA RESIDENTIAL · FO
25	I182070906	3831	Motor Vehicle Accident Response	M/V - LEAVING SCE PROPERTY DAM/
26	I182070905	3006	Medical Assistance	SICK/INJURED/MEDI - PER
27	I182070904	802	Simple Assault	ASSAULT SIMF BATT
28	I182070904	2007	Restraining Order Violations	VIOL. OF RESTRAIN ORDER W NO ARR
29	I182070903	2900	Other	VAL - VIOLATION AUTO LAW - OT
...	...	...	...	
319043	I110551302-00	3125	Warrant Arrests	WARRANT ARR
319044	I110551302-00	623	Larceny	LARCENY SHOPLIFT $50TC
319045	I110372326-00	403	Aggravated Assault	ASSAULT & BATT D/W - OT
319046	I110372326-00	3125	Warrant Arrests	WARRANT ARR
319047	I110261417-00	3125	Warrant Arrests	WARRANT ARR
319048	I110261417-00	619	Larceny	LARCENY OTHER $2 O
319049	I110177502-00	3125	Warrant Arrests	WARRANT ARR
319050	I110177502-00	802	Simple Assault	ASSAULT & BATT
319051	I110177502-00	3125	Warrant Arrests	WARRANT ARR
319052	I100636670-00	629	Larceny	LARCENY OTH $50TC
319053	I100636670-00	3125	Warrant Arrests	WARRANT ARR
319054	I100340225-00	3125	Warrant Arrests	WARRANT ARR
319055	I100340225-00	339	Robbery	ROBBERY - UNARM STR
319056	I100222105-02	3125	Warrant Arrests	WARRANT ARR
319057	I100033064-00	2907	Violations	VAL - OPERATING AF REV/SI
319058	I100033064-00	2910	Violations	VAL - OPERATING AF REV/SI
319059	I090321958-00	3125	Warrant Arrests	WARRANT ARR

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPT
319060	I090321958-00	3125	Warrant Arrests	WARRANT ARR
319061	I090317057-00	403	Aggravated Assault	ASSAULT & BATT D/W - OT
319062	I090317057-00	3125	Warrant Arrests	WARRANT ARR
319063	I080542626-00	3125	Warrant Arrests	WARRANT ARR
319064	I080542626-00	1848	Drug Violation	DRUGS - POSS CLAS - INTENT TO MFR D
319065	I080542626-00	1849	Drug Violation	DRUGS - POSS CLAS - COCAINE, E
319066	I060168073-00	1864	Drug Violation	DRUGS - POSS CLAS - INTENT MFR DIST D
319067	I060168073-00	3125	Warrant Arrests	WARRANT ARR
319068	I050310906-00	3125	Warrant Arrests	WARRANT ARR
319069	I030217815-08	111	Homicide	MURDER, N NEGLIGI MANSLAUGH
319070	I030217815-08	3125	Warrant Arrests	WARRANT ARR
319071	I010370257-00	3125	Warrant Arrests	WARRANT ARR
319072	142052550	3125	Warrant Arrests	WARRANT ARR

319073 rows × 17 columns

In [3]:

```
d2= pd.read_csv('offense_codes.csv', encoding='iso-8859-1')
```

First, let's clean up and simplify this data set. I am going to focus on the two years with complete data (2016 and 2017). I will also narrow in on UCR Part One offenses, which include only the most serious crimes.

In [4]:

```
# Keep only data from complete years (2016, 2017)
data = data.loc[data['YEAR'].isin([2016,2017])]
data=data[data["UCR_PART"]=='Part One']
data = data.drop(['INCIDENT_NUMBER','OFFENSE_CODE','UCR_PART','Location'], axis=1)
data['OCCURRED_ON_DATE'] = pd.to_datetime(data['OCCURRED_ON_DATE'])
data['SHOOTING'].fillna('N', inplace=True)
# Convert DAY_OF_WEEK to an ordered category
data['DAY_OF_WEEK'] = pd.Categorical(data.DAY_OF_WEEK,
              categories=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday'
,'Sunday'],
              ordered=True)

# Replace -1 values in Lat/Long with Nan
data.Lat.replace(-1, None, inplace=True)
data.Long.replace(-1, None, inplace=True)

# Rename columns to something easier to type (the all-caps are annoying!)
rename = {'OFFENSE_CODE_GROUP':'Group',
          'OFFENSE_DESCRIPTION':'Description',
          'DISTRICT':'District',
          'REPORTING_AREA':'Area',
          'SHOOTING':'Shooting',
          'OCCURRED_ON_DATE':'Date',
          'YEAR':'Year',
          'MONTH':'Month',
          'DAY_OF_WEEK':'Day',
          'HOUR':'Hour',
          'STREET':'Street'}
data.rename(index=str, columns=rename, inplace=True)

# Check
data['Group'].value_counts()
```

Out[4]:

```
Larceny                       15709
Larceny From Motor Vehicle     6707
Aggravated Assault             4769
Residential Burglary           3309
Auto Theft                     2930
Robbery                        2882
Commercial Burglary             863
Other Burglary                  268
Homicide                        101
Name: Group, dtype: int64
```

In [5]:

```
data.dtypes
data.isnull().sum()
data.shape
```

Out[5]:

```
(37538, 13)
```

Types of serious crimes

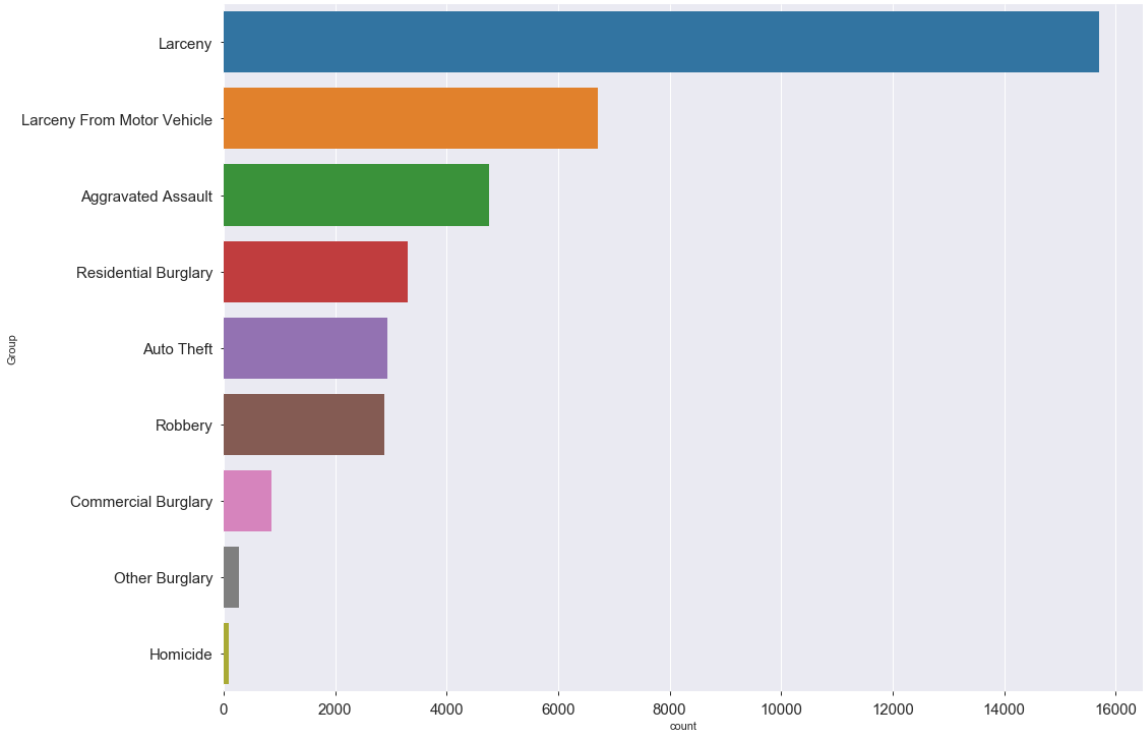
Let's start by checking the frequency of different types of crimes. Since we have subsetted to only 'serious' crimes, there are only 9 different types of offenses - much more manageable than the 67 we started with.

In [6]:

```
sns.catplot(y='Group', kind='count', height=10, aspect=1.5, order=data.Group.value_coun
ts().index,  data=data)
plt.xticks(size=15)
plt.yticks(size=15)
```

Out[6]:

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8]), <a list of 9 Text yticklabel objects
>)
```



Larceny is by far the most common serious crime, and homicides are pretty rare.

When do serious crimes occur?

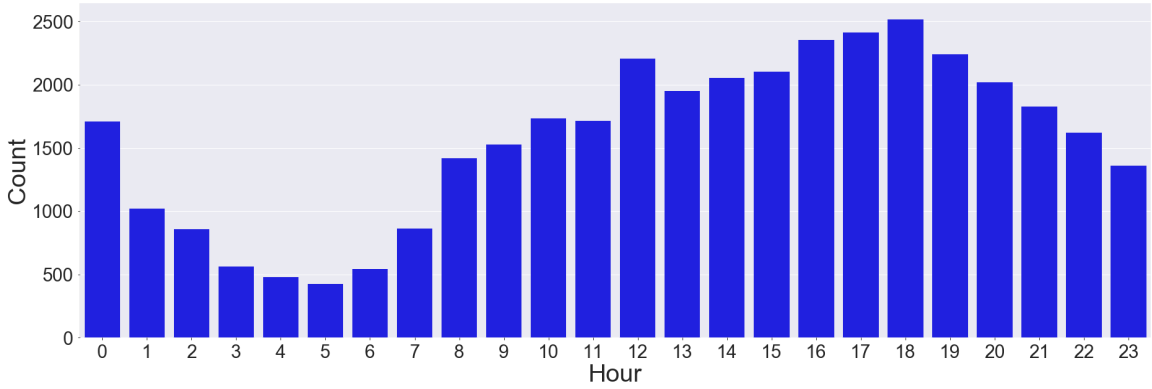
We can consider patterns across several different time scales: hours of the day, days of the week, and months of the year.

In [7]:

```
sns.catplot(x='Hour',
            kind='count',
             height=8.27,
             aspect=3,
             color='blue',
            data=data)
plt.xticks(size=30)
plt.yticks(size=30)
plt.xlabel('Hour', fontsize=40)
plt.ylabel('Count', fontsize=40)
```

Out[7]:

```
Text(-1.4500000000000028, 0.5, 'Count')
```

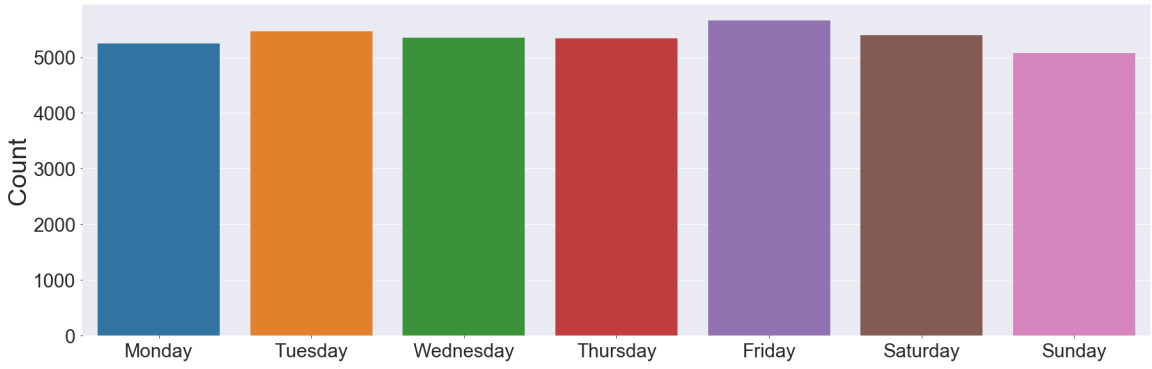


In [8]:

```
sns.catplot(x='Day',
            kind='count',
             height=8,
             aspect=3,
            data=data)
plt.xticks(size=30)
plt.yticks(size=30)
plt.xlabel('')
plt.ylabel('Count', fontsize=40)
```

Out[8]:

```
Text(-1.4499999999999744, 0.5, 'Count')
```



In [9]:

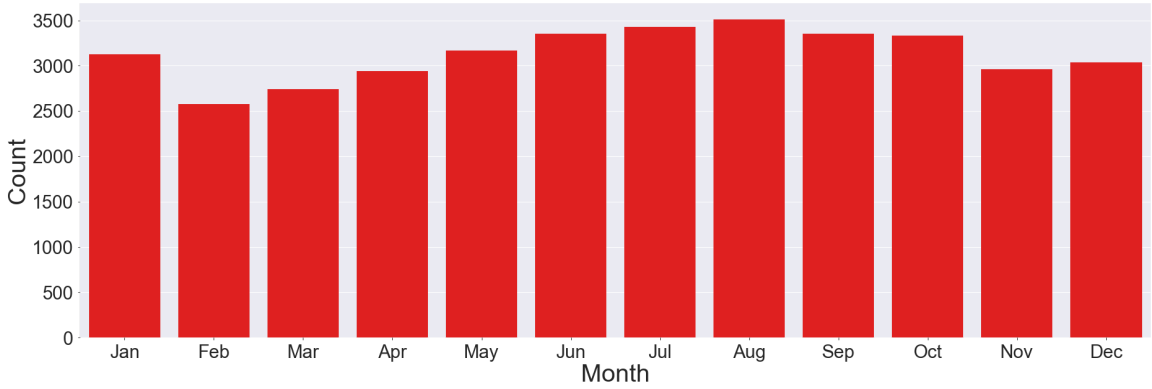
```
x=data.groupby('Group')['Day'].value_counts()
x= pd.DataFrame(x)
```

In [10]:

```
Months = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

sns.catplot(x='Month',
            kind='count',
             height=8.27,
             aspect=3,
             color='red',
            data=data)
plt.xticks(np.arange(12), Months, size=30)

# plt.xticks(size=30)
plt.yticks(size=30)
plt.xlabel('Month', fontsize=40)
plt.ylabel('Count', fontsize=40)
```

Out[10]:

```
Text(-1.4500000000000028, 0.5, 'Count')
```



Crimes rates are low between 1-8 in the morning, and gradually rise throughout the day, peaking around 6 pm. There is some variation across days of the week, with Friday having the highest crime rate and Sunday having the lowest. The month also seems to have some influence, with the winter months of February-April having the lowest crime rates, and the summer/early fall months of June-October having the highest crime rates. There is also a spike in crime rates in the month of January.

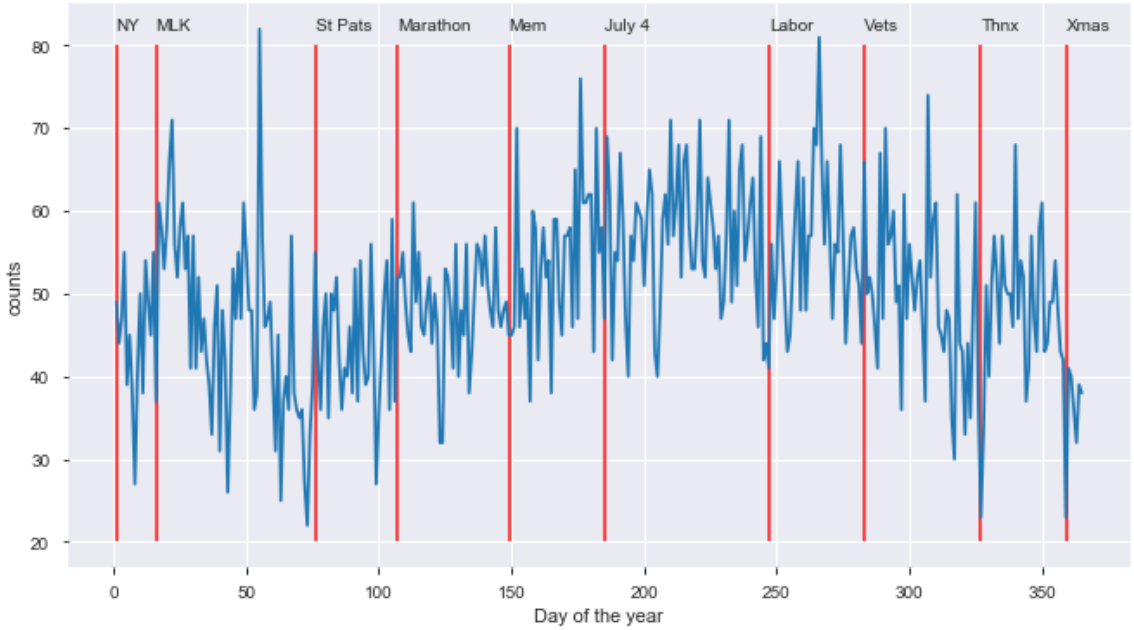
Are any other temporal factors associated with crime? [According to some crime experts (https://www.oxygen.com/homicide-for-the-holidays/blogs/its-the-most-dangerous-time-of-the-year-why-do-crimes-increase)](https://www.oxygen.com/homicide-for-the-holidays/blogs/its-the-most-dangerous-time-of-the-year-why-do-crimes-increase), several types of crime tend to increase around the holidays, particularly larsony and robbery. This can occur for many reasons: crowded shopping centers create more cover for thieves, travelers leave their homes vulnerable to burglary, and increased alcohol and drug use can raise the likelihood of conflict-related crime. Let's see if there is any evidence for this in our data, focusing in on the year 2017. I also added in a couple of days that are known to be especially rowdy in Boston, even though they aren't official holidays: St. Patrick's Day and the Boston Marathon.

In [11]:

```
# Create data for plotting
data['Day_of_year'] = data["Date"].dt.dayofyear
data_holidays = data[data.Year == 2017].groupby(['Day_of_year']).size().reset_index(nam
e='counts')
```

In [12]:

```
# Dates of major U.S. holidays in 2017
holidays = pd.Series(['2017-01-01', # New Years Day
                      '2017-01-16', # MLK Day
                      '2017-03-17', # St. Patrick's Day
                      '2017-04-17', # Boston marathon
                      '2017-05-29', # Memorial Day
                      '2017-07-04', # Independence Day
                      '2017-09-04', # Labor Day
                      '2017-10-10', # Veterans Day
                      '2017-11-23', # Thanksgiving
                      '2017-12-25']) # Christmas
holidays = pd.to_datetime(holidays).dt.dayofyear
holidays_names = ['NY',
                  'MLK',
                  'St Pats',
                  'Marathon',
                  'Mem',
                  'July 4',
                  'Labor',
                  'Vets',
                  'Thnx',
                  'Xmas']

import datetime as dt
# Plot crimes and holidays
fig, ax = plt.subplots(figsize=(11,6))
sns.lineplot(x='Day_of_year',
             y='counts',
             ax=ax,
             data=data_holidays)
plt.xlabel('Day of the year')
plt.vlines(holidays, 20, 80, alpha=0.8, color ='r')
for i in range(len(holidays)):
    plt.text(x=holidays[i], y=82, s=holidays_names[i])
```



Hm, I'm not seeing any clear signals here. In fact, many of these holidays appear to line up with especially low crime rates, particularly Thanksgiving and Christmas. Of course, this is data from just a single year, and detecting an association between a given holiday and crime rates would require a lot more data and a model that accounts for other factors. However, this does cause me to question the general idea that crime increases surrounding holidays - if that *is* true, it isn't super obvious from a birds-eye view of the data. Even the entire "holiday season" (https://www.cpss.net/about/blog/2013/11/stay-safe-crime-rates-increase-during-holiday-season/) from Thanksgiving to Christmas doesn't seem to be especially elevated compared to the summer.

Where do serious crimes occur?

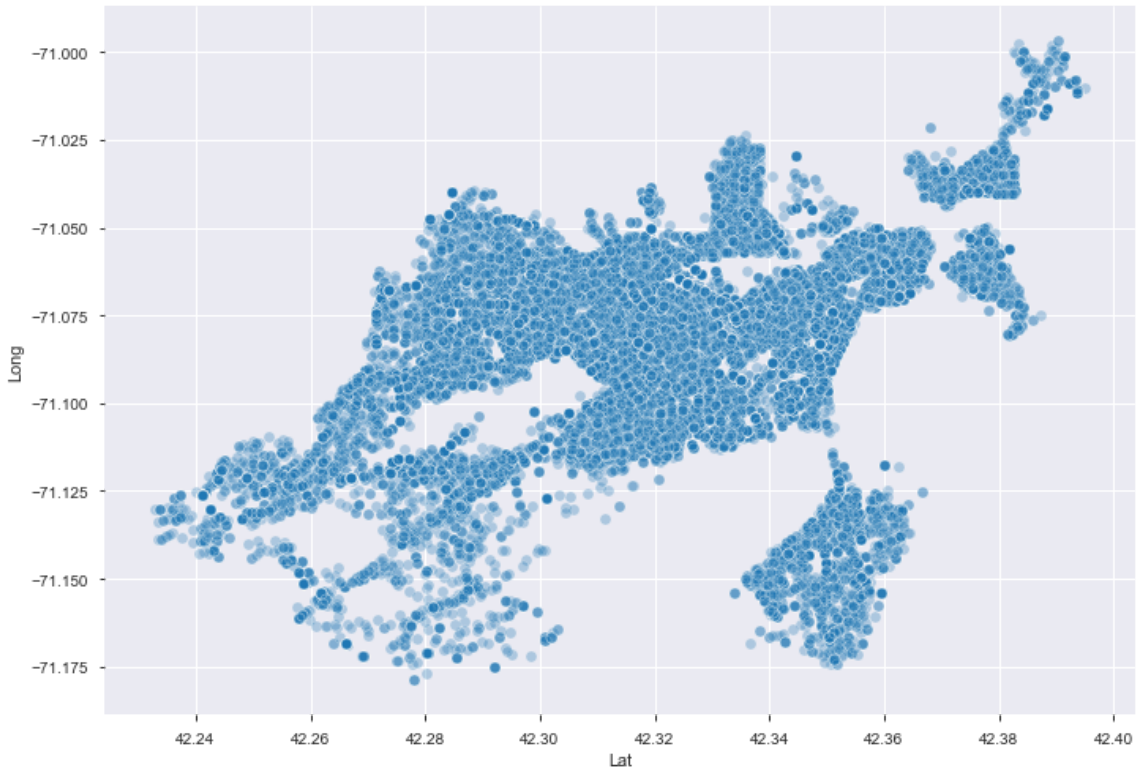
We can use the latitude and longitude columns to plot the location of crimes in Boston. By setting the alpha parameter to a very small value, we can see that there are some crime 'hotspots'.

In [13]:

```
# Simple scatterplot
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
sns.scatterplot(x='Lat',
                y='Long',
                 alpha=0.3,
                data=data, ax=ax)
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2086c672588>
```



That looks like Boston alright. If you are at all familiar with Boston, you will not be too surprised to see that downtown Boston has the darkest points, but there are also some localities outside of the city center that have especially high crime rates.

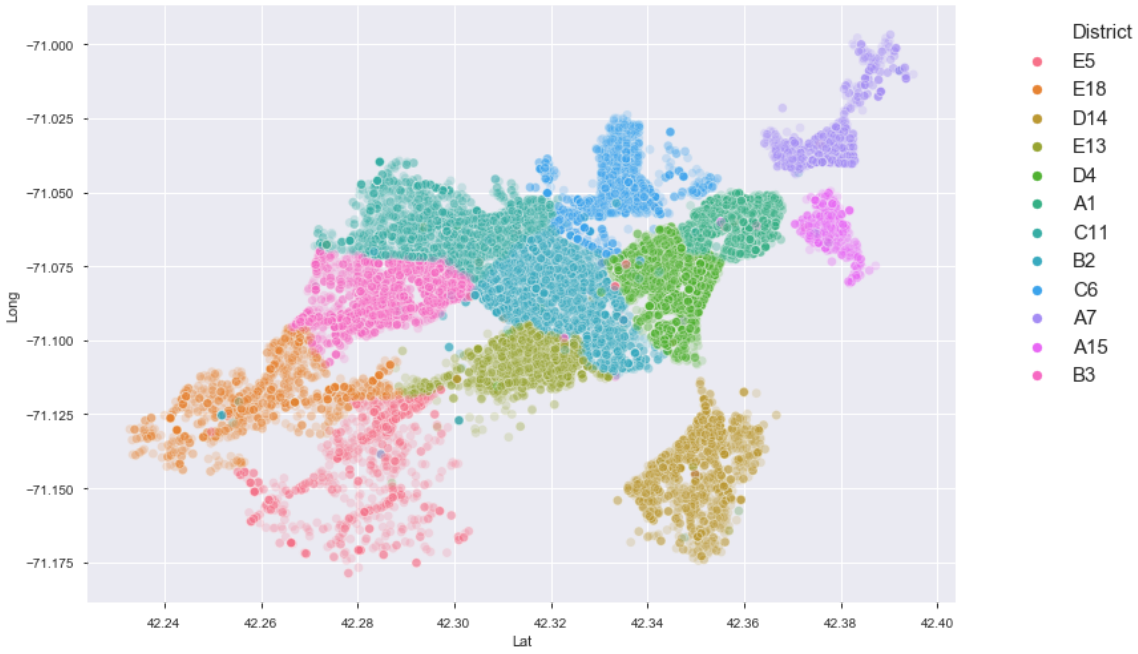
Let's make another scatterplot, but this time we'll color points by district to see which districts have the highest crime rates.

In [14]:

```
# Plot districts
a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
sns.scatterplot(x='Lat',
                y='Long',
                 hue='District',
                 alpha=0.2,
                data=data, ax=ax)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, fontsize=15)
```

Out[14]:

```
<matplotlib.legend.Legend at 0x2086c73d710>
```



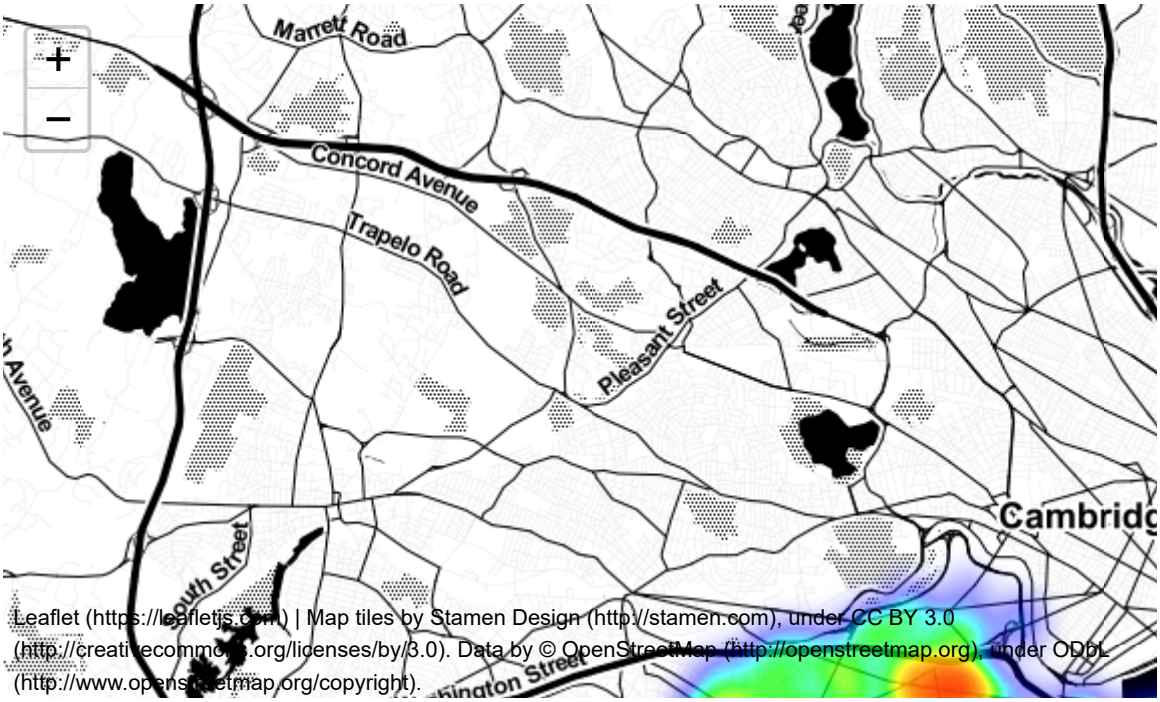
We can now associate high crime rates with particular districts, most noteably A1 and D4, which correspond to the most crowded areas of downtown Boston. There is also a very high crime region visibe in district D14.

Let's make things pretty by using Folium to make an interactive heatmap of Boston crimes. I will use the 2017 data only for this plot.

In [15]:

```
# Create basic Folium crime map
crime_map = folium.Map(location=[42.3125,-71.0875],
                       tiles = "Stamen Toner",
                      zoom_start = 11)

# Add data for heatmp
data_heatmap = data[data.Year == 2017]
data_heatmap = data[['Lat','Long']]
data_heatmap = data.dropna(axis=0, subset=['Lat','Long'])
data_heatmap = [[row['Lat'],row['Long']] for index, row in data_heatmap.iterrows()]
HeatMap(data_heatmap, radius=10).add_to(crime_map)

# Plot!
crime_map
```

Out[15]:



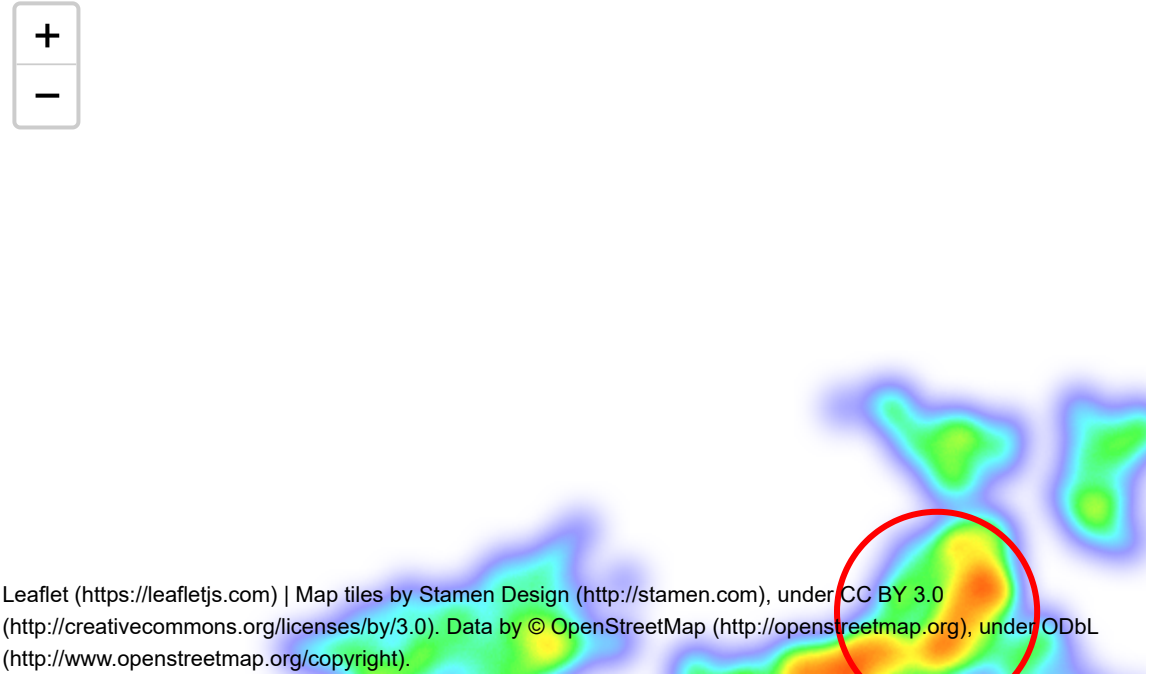
ABOUT LARCENY

Most of them around Newburry Street, Boylston Street, State Street and Downtown Crossing

In [16]:

```
import folium
from folium.plugins import HeatMap

map_hooray = folium.Map(location=[42.361145,-71.057083],
                    zoom_start = 12, min_zoom=12, tiles= "Stamen Toner" ) #Giving the l
ocation just write boston coordinat to google

heat_df = data[(data['Year']==2017 )& (data['Group']=='Larceny')]# I take 2017 cause th
ere is more crime against to other years
# heat_df = data[data['Group']=='Larceny']
heat_df = heat_df[['Lat', 'Long']] #giving only latitude and longitude now in heat_df j
ust latitude and longitude
                                        #from 2017 larceny responde
heat_df=heat_df.dropna()
folium.CircleMarker([42.356145,-71.064083],
                    radius=50,
                    popup='Homicide',
                    color='red',
                    ).add_to(map_hooray) #Adding mark on the map but it's hard to find
 correct place.
                                         #it's take to muhc time


heat_data = [[row['Lat'],row['Long']] for index, row in heat_df.iterrows()]
#We have to give latitude and longitude like this [[lat, lon],[lat, lon],[lat, lon],[la
t, lon],[lat, lon]]

HeatMap(heat_data, radius=10).add_to(map_hooray) #Adding map_hooray to HeatMap
map_hooray #Plotting
```

Out[16]:



ABOUT MOTOR VEHICLE ACCIDENT RESPONSE

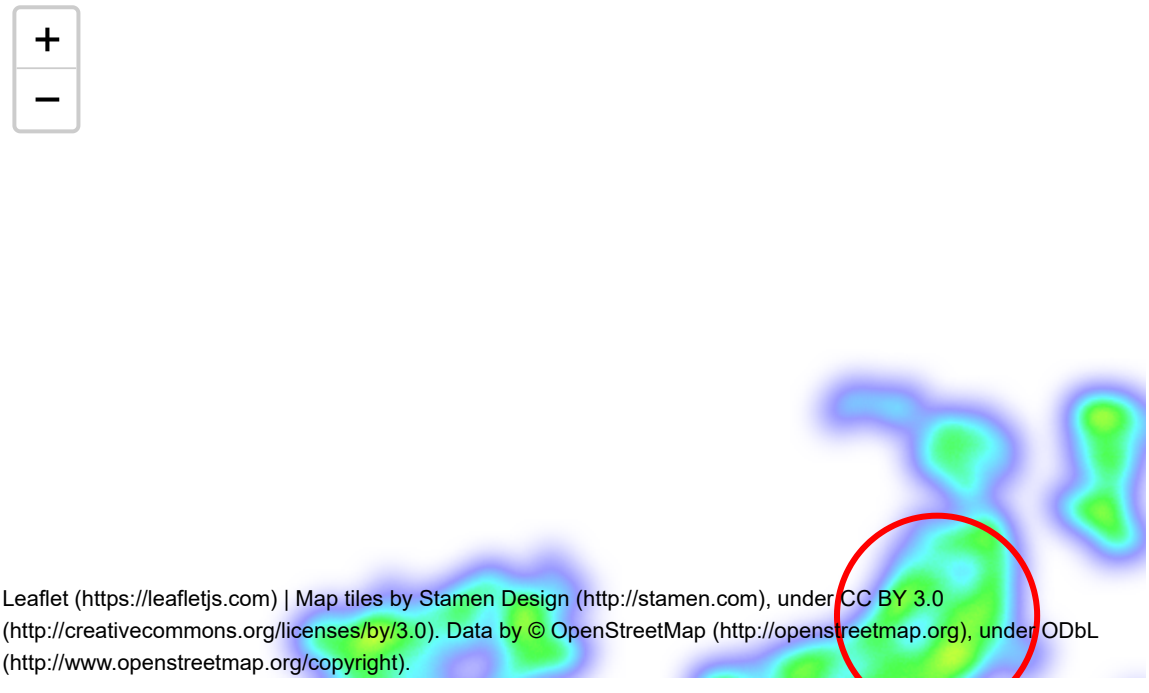
It's look everywhere is almost same accident it's mean thoese accident not cause of city road planning

Probably it's cause of human mistakes

In [17]:

```
map_hooray = folium.Map(location=[42.361145,-71.057083],
                    zoom_start = 12, min_zoom=12, tiles= "Stamen Toner" )

heat_df = data[(data['Year']==2017 )& (data['Group']=='Larceny From Motor Vehicle')]# I
take 2017 cause there is more crime against to other years
heat_df = heat_df[['Lat', 'Long']]
heat_df = heat_df.dropna()

folium.CircleMarker([42.356145,-71.064083],
                    radius=50,
                    popup='Homicide',
                    color='red',
                    ).add_to(map_hooray) #Adding mark on the map but it's hard to find
 correct place.
                                         #it's take to muhc time

heat_data = [[row['Lat'],row['Long']] for index, row in heat_df.iterrows()]
HeatMap(heat_data, radius=10).add_to(map_hooray)
map_hooray
```

Out[17]:

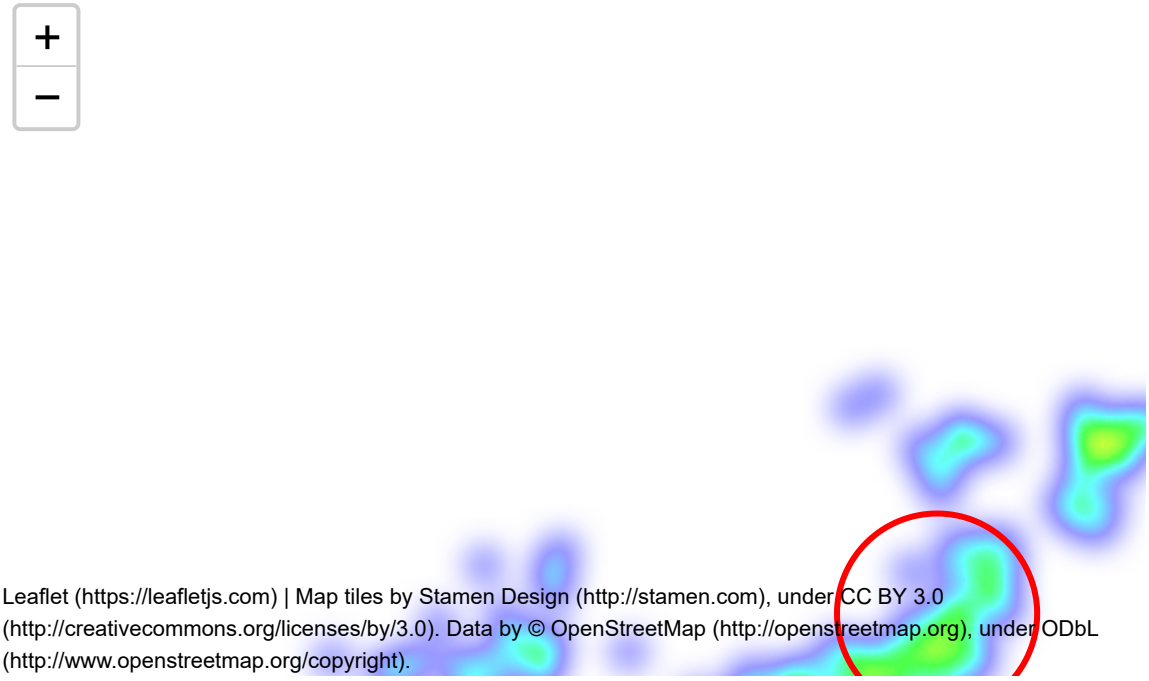


About Robbery

In [18]:

```
map_hooray = folium.Map(location=[42.361145,-71.057083],
                    zoom_start = 12, min_zoom=12, tiles= "Stamen Toner" )

heat_df = data[(data['Year']==2017 )& (data['Group']=='Robbery')]# I take 2017 cause th
ere is more crime against to other years
heat_df = heat_df[['Lat', 'Long']]
heat_df = heat_df.dropna()

folium.CircleMarker([42.356145,-71.064083],
                    radius=50,
                    popup='Homicide',
                    color='red',
                    ).add_to(map_hooray) #Adding mark on the map but it's hard to find
 correct place.
                                         #it's take to muhc time

heat_data = [[row['Lat'],row['Long']] for index, row in heat_df.iterrows()]
HeatMap(heat_data, radius=10).add_to(map_hooray)
map_hooray
```

Out[18]:



Conclusions

In summary, this EDA shows:

- Larceny is by far the most common type of serious crime.
- Serious crimes are most likely to occur in the afternoon and evening.
- Serious crimes are most likely to occur on Friday and least likely to occur on Sunday.
- Serious crimes are most likely to occur in the summer and early fall, and least likely to occur in the winter (with the exeption of January, which has a crime rate more similar to the summer).
- There is no obvious connection between major holidays and crime rates.
- Serious crimes are most common in the city center, especially districts A1 and D4.

This EDA just scratches the surface of the dataset. Further analyses could explore how different types of crimes vary in time and space. I didn't even consider the less serious UCR Part Two and Part Three crimes, which are far more common than Part One crimes, but include interesting categories such as drug crimes. Another interesting direction would be to combine this with other data about Boston, such as demography or even the weather (http://www.chicagotribune.com/news/data/ct-crime-heat-analysis-htmlstory.html), to investigate what factors predict crime rates across time and space.