# Project

April 6, 2019

## 0.1 Background, Introduction, and Description of Data

### 0.1.1 Brief Background

After moving to Boston as freshmen entering Northeastern, we wanted to explore the city for more than just the touristy elements of it. After finding two disjoint datasets from AirBnB and the Government of Massachusetts, we found that by cross-examining crime and real estate, we can see if the number of crimes in an area has any effect on prices, availability, and ratings of AirBnB listings.

### 0.1.2 Data Composition

The AirBnB data came from here and contains thousands of rows of every AirBnB listing in the greater Boston area. It also has a separate CSV file containing reviews for each AirBnB, but those were not used in the scope of this project. The listings file included data like average reviews, availability, coordinates, address, etc. for each AirBnB.

The crime data came from here and contains hundreds of thousands of rows for every crime in the greater Boston area. This had a lot of data to analyze effectively, but luckily Pandas was able to read it very quickly in comparison to Excel. This data also had coordinates of each crime, which was useful to cross reference with the AirBnB data to examine crimes v. listings in a given cluster. In addition, it had the date/time of the crime, whether or not a shooting was involved, the offense code, and a short description of the offense.

### 0.1.3 Importance

Analyzing this data is important for property owners as well as AirBnB customers. Owners of real estate to appropriately gauge demand for their properties and see what pays off the best - do cheap properties in shady areas do better than expensive properties in nice areas? What about nice places in shady areas vs. meh places in nice areas? What's an ideal property to have?

For an AirBnB consumer, we can analyze how risky an AirBnB is. No host will ever say that their place is in a crime-heavy section of the city, so it's important for travelers to know that they will be living in a safe place when they visit Boston.

```
In [1]: # System imports
        import os
        from tqdm import tqdm

        # Data manipulation and analysis imports
```

```
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np
from scipy.spatial import Voronoi, voronoi_plot_2d

# Data visualization imports
from matplotlib import pyplot as plt
import mplleaflet
from plot_voronoi import voronoi_finite_polygons_2d
```

## 0.2 Methods and Analyses

First, let's start off by reading all the files in the data/ directory. This includes the crime.csv file, the reviews.csv file, and the listings.csv file. The reviews and listings belong to the AirBnB dataset, while the crime file obviously belongs to the crime dataset.

Also, we set the number of desired clusters here to use for analysis. Changing this number up here keeps it consistent with the rest of the file, so if we want to use more or less, we can change it here. We decided on 30 since we felt like that would be an appropriate number to help visualize.

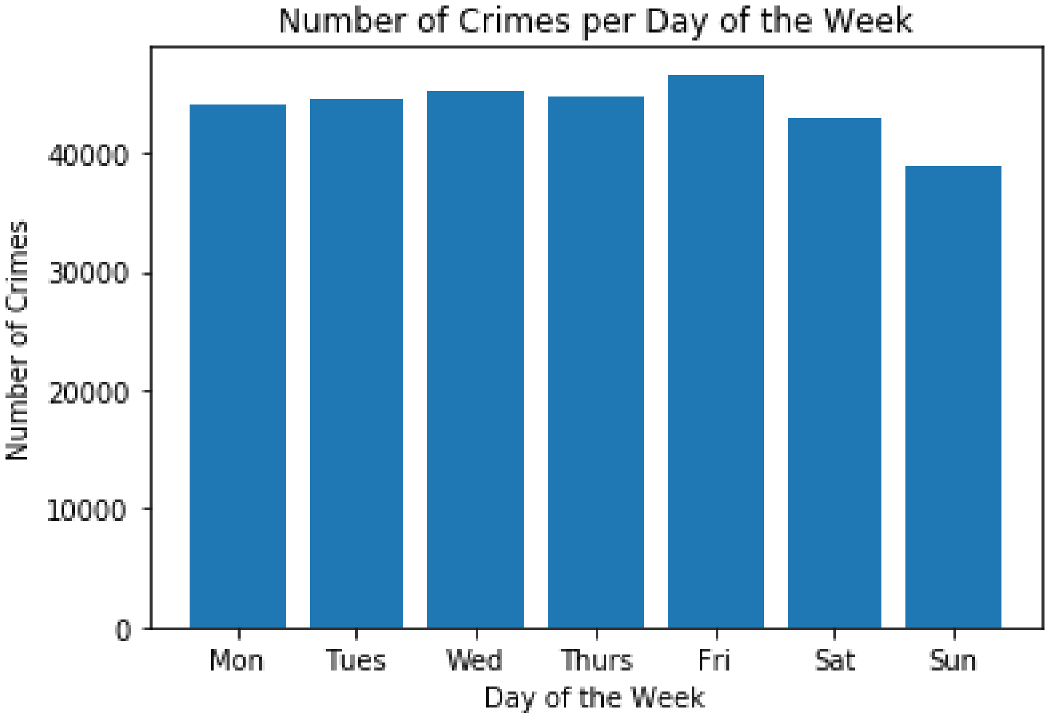
```
In [2]: dfs = {}
        for csv in os.listdir('data'):
            name = csv.split('.')[0]
            dfs[name] = pd.read_csv('data/'+csv)

        crime, listings, reviews = [dfs[a] for a in sorted(dfs.keys())[1:]]

        n_clusters = 30
```

### 0.2.1 Day of the week analysis

Can we say when's the safest time to visit Boston based on day of the week? Do crimes tend to happen on certain days more than other days?

```
In [3]: # There's probably a more elegant way of doing this in Pandas, but since we couldn't f
        # dictionary of the number of crimes committed on each day, then made a bar graph usin
        # dictionary

        days = crime.groupby(['DAY_OF_WEEK']).count()\
            .reset_index()[['DAY_OF_WEEK', 'INCIDENT_NUMBER']].set_index('DAY_OF_WEEK').to_dic

        plt.bar(['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun'],
                [days['Monday']['INCIDENT_NUMBER'],
                 days['Tuesday']['INCIDENT_NUMBER'],
                 days['Wednesday']['INCIDENT_NUMBER'],
                 days['Thursday']['INCIDENT_NUMBER'],
                 days['Friday']['INCIDENT_NUMBER'],
                 days['Saturday']['INCIDENT_NUMBER'],
                 days['Sunday']['INCIDENT_NUMBER']])
        plt.title('Number of Crimes per Day of the Week')
        plt.ylabel('Number of Crimes')
        plt.xlabel('Day of the Week')
```

```
Out[3]: Text(0.5, 0, 'Day of the Week')
```



### 0.2.2 Geographical Clustering

By breaking down the greater Boston area into a bunch of clusters, we can analyze segments of the city in greater detail. Clusters are based on local density, so smaller clusters typically have higher crime rates, whereas bigger clusters will have less crime rates. With 30 clusters, each cluster is relatively the same size.

```
In [4]: coors = np.array(list(zip(listings['latitude'], listings['longitude'])))
        kmeans = KMeans(n_clusters = n_clusters)
        coors = listings[['latitude', 'longitude']].dropna()
        fig = plt.figure()

        points = np.array(list(zip(coors['longitude'], coors['latitude'])))

        kmeans.fit(points)

        centers = kmeans.cluster_centers_
```

```
# A lot of this code was taken from this Github Gist:
# https://gist.github.com/pv/8036995

# This helped plot and colorize the cluster areas using the Voronoi technique.

vor = Voronoi(centers)
boundaries = voronoi_plot_2d(vor)

# make up data points
np.random.seed(1234)

# plot
regions, vertices = voronoi_finite_polygons_2d(vor)

# colorize
for region in regions:
    polygon = vertices[region]
    plt.fill(*zip(*polygon), alpha=0.4)

#plt.plot(points[:,0], points[:,1], 'ko')
plt.xlim(vor.min_bound[0] - 0.1, vor.max_bound[0] + 0.1)
plt.ylim(vor.min_bound[1] - 0.1, vor.max_bound[1] + 0.1)
plt.plot([-42, -43], [70, 71], alpha = 0)
mplleaflet.display(tiles='cartodb_positron')
```

```
/home/andy/anaconda3/lib/python3.7/site-packages/IPython/core/display.py:689: UserWarning: Cons
  warnings.warn("Consider using IPython.display.IFrame instead")
```

```
Out[4]: <IPython.core.display.HTML object>
```

```
<Figure size 432x288 with 0 Axes>
```

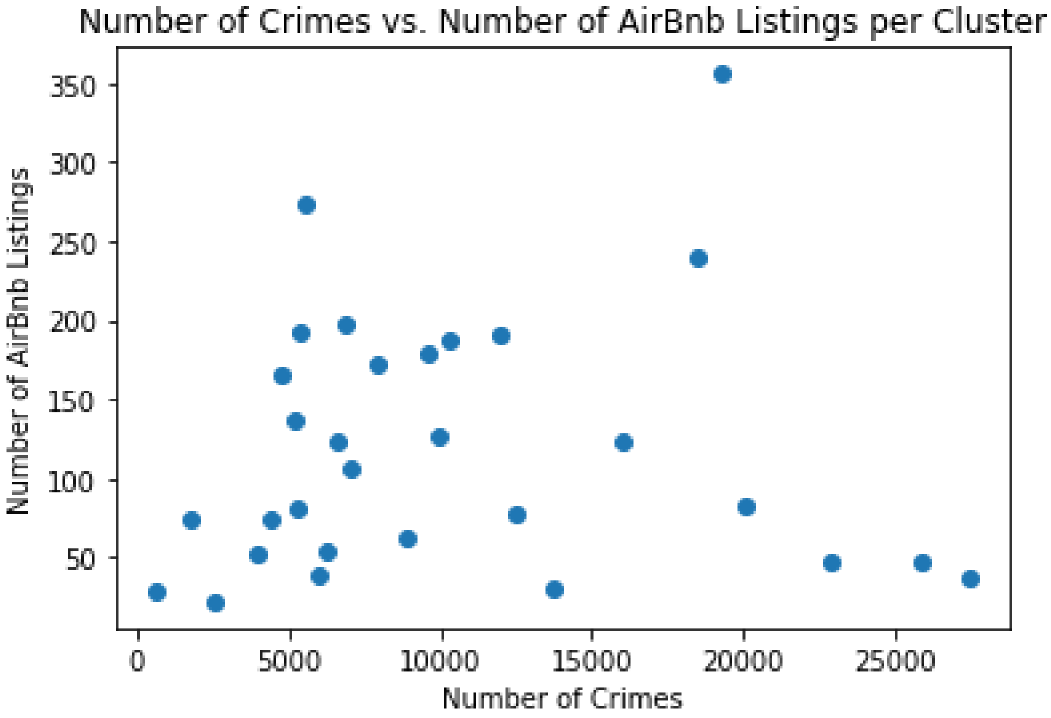
```
In [5]: # We can drop crimes with null lat/long coordinates in order to cluster each crime and
        # a cluster column to each dataframe

        crime = crime.dropna(subset=['Lat', 'Long'])

        crime['cluster'] = kmeans.predict(np.array(list(zip(crime['Long'], crime['Lat']))))
        #crime.to_csv('data/crime.csv')

        listings['cluster'] = kmeans.predict(np.array(list(zip(listings['longitude'], listings
        #listings.to_csv('data/listings.csv')
```

### 0.2.3 Cross-Referencing

Now that we have geographical data for AirBnB listings as well as crimes, we can group them by clusters and see if crimes have any effects on prices, availabilities, and reviews of AirBnB listings

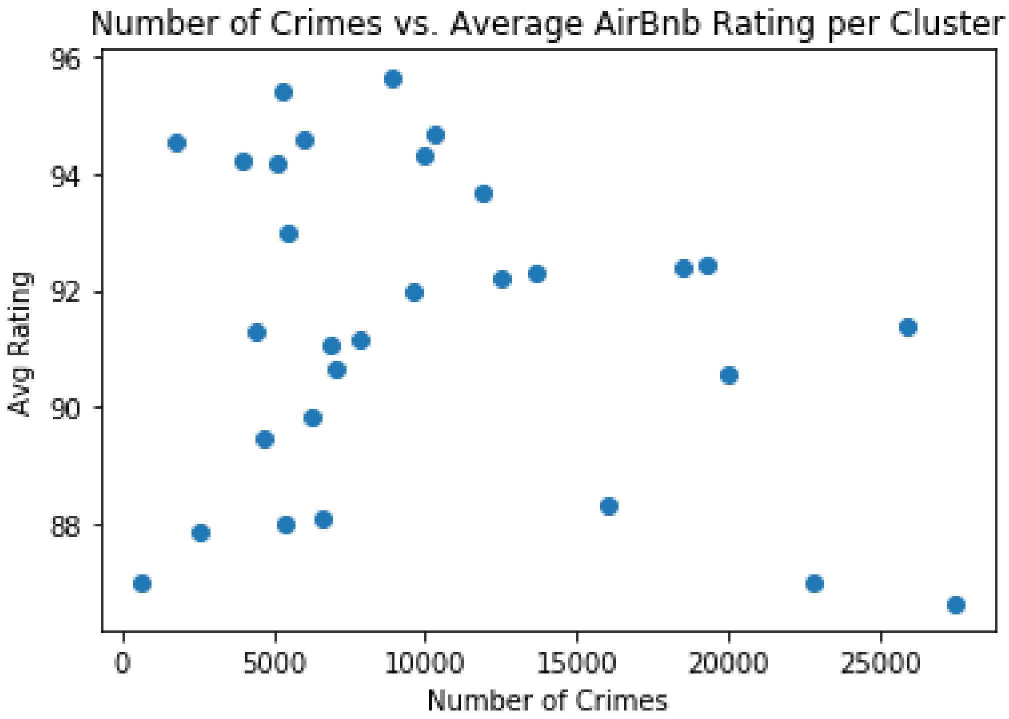
```
In [6]: count_crimes = crime.groupby(['cluster']).size().reset_index().set_index('cluster')
        count_listings = listings.groupby(['cluster']).size().reset_index().set_index('cluster

        crimes_listings = count_crimes.join(count_listings, lsuffix='_crimes', rsuffix='_listin
            .rename(index=str, columns={"0_listings": "listings", "0_crimes": "crimes"})

        plt.scatter(crimes_listings['crimes'], crimes_listings['listings'])
        plt.xlabel('Number of Crimes')
        plt.ylabel('Number of AirBnb Listings')
        plt.title('Number of Crimes vs. Number of AirBnb Listings per Cluster')
```

```
Out[6]: Text(0.5, 1.0, 'Number of Crimes vs. Number of AirBnb Listings per Cluster')
```



```
In [20]: grouped_listings = listings[listings['number_of_reviews'] > 0].groupby(['cluster']).me
             .reset_index().set_index('cluster')

         crimes_listings = count_crimes.join(grouped_listings, lsuffix='_crimes', rsuffix='_lis
             .rename(index=str, columns={0: "crimes", 'review_scores_rating': 'avg_rating'})

         plt.scatter(crimes_listings['crimes'], crimes_listings['avg_rating'])
         plt.xlabel('Number of Crimes')
         plt.ylabel('Avg Rating')
         plt.title('Number of Crimes vs. Average AirBnb Rating per Cluster')
```

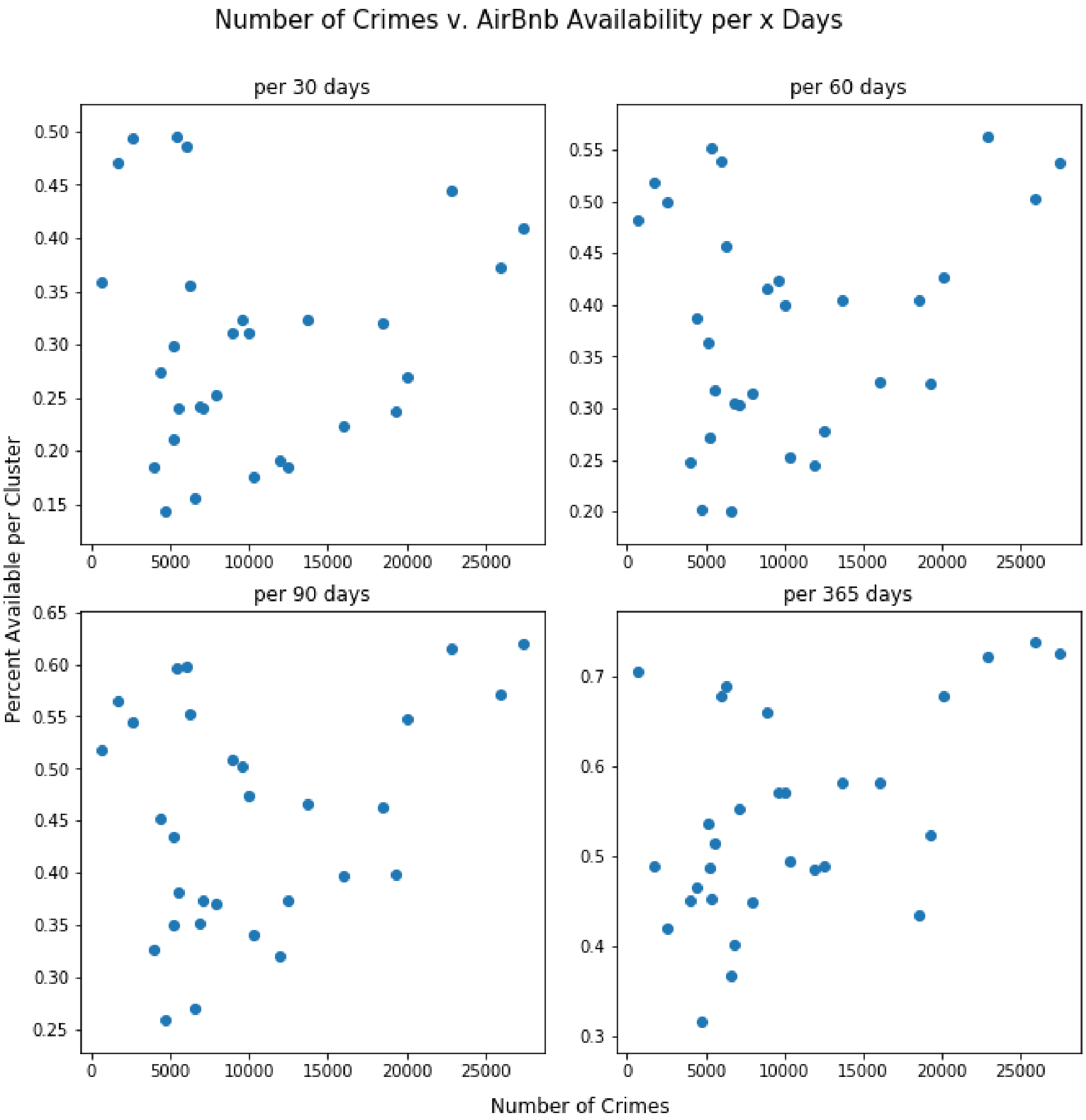
```
Out[20]: Text(0.5, 1.0, 'Number of Crimes vs. Average AirBnb Rating per Cluster')
```



```
In [22]: #60, 90, 365

         fig = plt.figure(figsize=(10,10))
         ax = fig.add_subplot(111)

         ax.spines['top'].set_color('none')
         ax.spines['bottom'].set_color('none')
         ax.spines['left'].set_color('none')
         ax.spines['right'].set_color('none')
         ax.tick_params(labelcolor='w', top='off', bottom='off', left='off', right='off')
         ax.set_xlabel('Number of Crimes', fontsize = 12, labelpad=10)
         ax.set_ylabel('Percent Available per Cluster', fontsize = 12, labelpad=10)

         for i, days in enumerate([30, 60, 90, 365]):
             temp_ax = fig.add_subplot(2, 2, i+1)
             temp_ax.scatter(crimes_listings['crimes'], crimes_listings[f'availability_{days}']
             #temp_ax.set_xlabel('Number of Crimes')
             #temp_ax.set_ylabel('Avg Rating')
             temp_ax.set_title(f'per {days} days')

         fig.suptitle('Number of Crimes v. AirBnb Availability per x Days', fontsize=15)
```

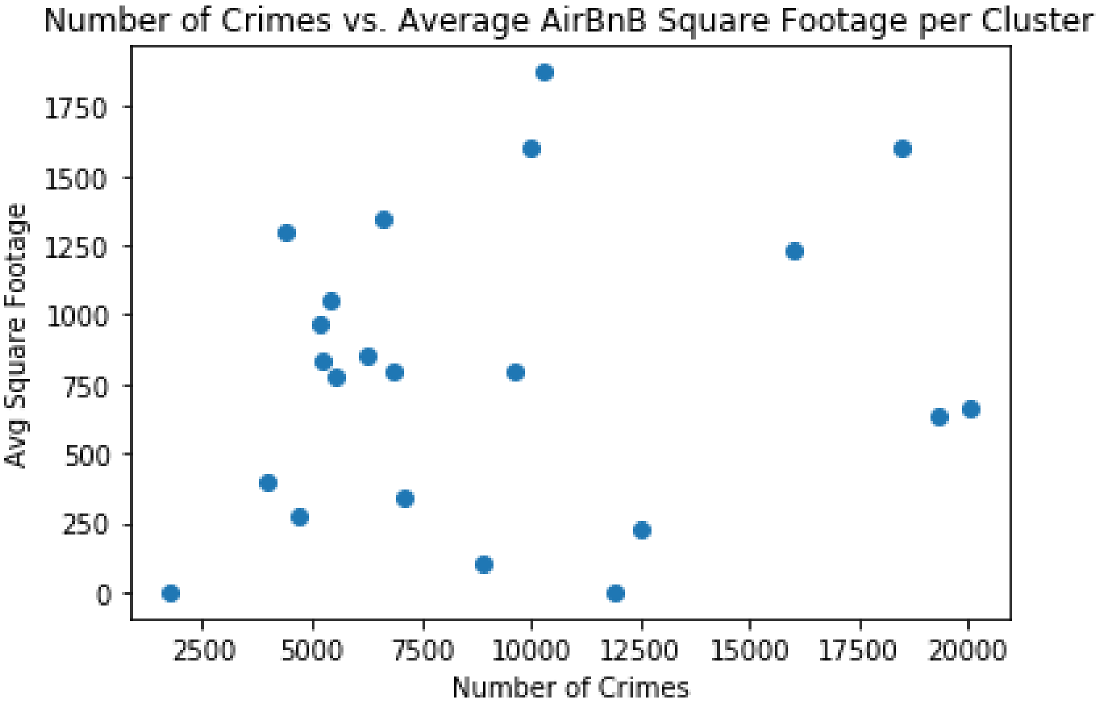
```
        plt.tight_layout()
        plt.subplots_adjust(top=.9)
```

```
/home/andy/anaconda3/lib/python3.7/site-packages/matplotlib/cbook/__init__.py:424: MatplotlibDe
Passing one of 'on', 'true', 'off', 'false' as a boolean is deprecated; use an actual boolean
  warn_deprecated("2.2", "Passing one of 'on', 'true', 'off', 'false' as a "
```



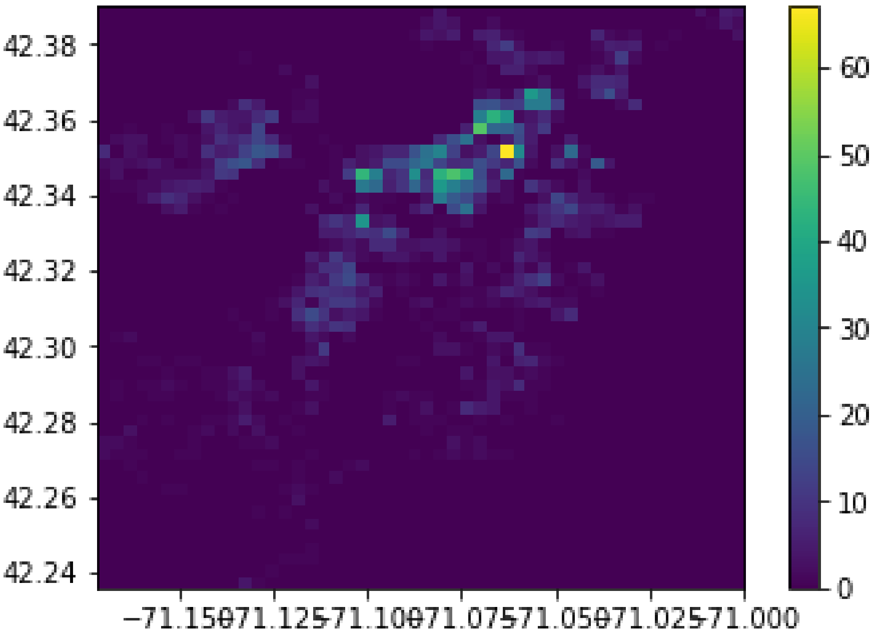
```
In [23]: plt.scatter(crimes_listings['crimes'], crimes_listings['square_feet'])
         plt.xlabel('Number of Crimes')
         plt.ylabel('Avg Square Footage')
         plt.title('Number of Crimes vs. Average AirBnB Square Footage per Cluster')
```

```
Out[23]: Text(0.5, 1.0, 'Number of Crimes vs. Average AirBnB Square Footage per Cluster')
```

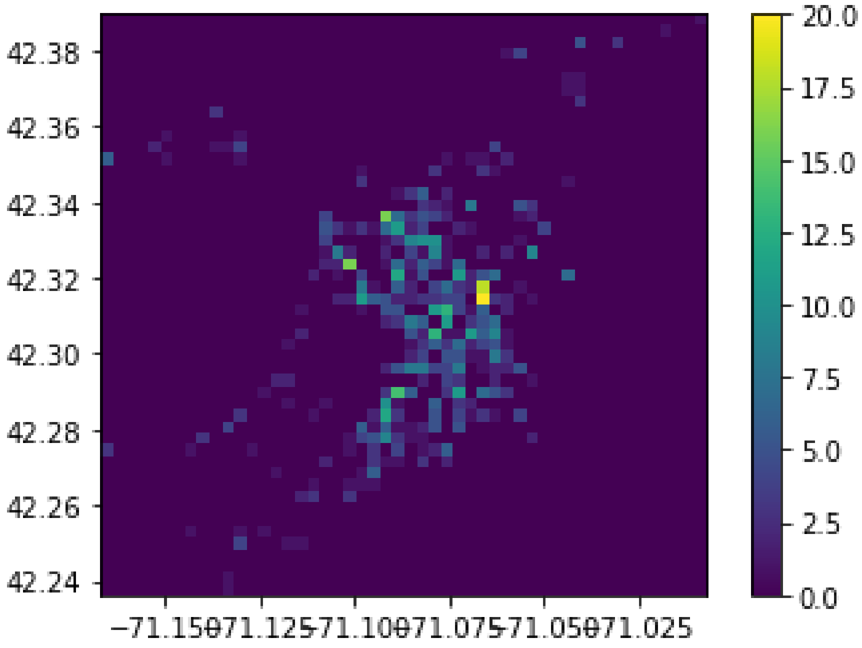


```
In [24]: heatmap, xedges, yedges = np.histogram2d(*list(zip(*points)), bins=50)
         extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

         plt.clf()
         boundaries = plt.imshow(heatmap.T, extent=extent, origin='lower')
         plt.colorbar()
         #mplleaflet.display(boundaries.figure, tiles='cartodb_positron')
```

```
Out[24]: <matplotlib.colorbar.Colorbar at 0x7fb765b48ba8>
```



```
In [31]: # TODO: Add more colors

         shooting = crime.dropna(subset=['SHOOTING'])
         c = shooting.dropna()[['Lat', 'Long', 'cluster']]
         points = np.array(list(zip(c['Long'], c['Lat'])))

         heatmap, xedges, yedges = np.histogram2d(*list(zip(*points)), bins=50)
         extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

         plt.clf()
         boundaries = plt.imshow(heatmap.T, extent=extent, origin='lower')
         plt.colorbar()

Out[31]: <matplotlib.colorbar.Colorbar at 0x7fb750540b00>
```



```
In [32]: # TODO: Add more colors

         shooting = crime.dropna(subset=['SHOOTING'])
         coors = shooting.dropna()[['Lat', 'Long', 'cluster']]

         for i in range(n_clusters):
             c = coors.loc[coors['cluster'] == i]
             points = np.array(list(zip(c['Long'], c['Lat'])))
             try:
                 plt.scatter(*zip(*points))
             except:
                 1+1

         mplleaflet.display(tiles='cartodb_positron')

Out[32]: <IPython.core.display.HTML object>
```

## 0.3 Results and Conclusions

Instead of doing an analysis after every cell, it would be easier to read if everything was just down here. Some graphs showed great correlations, while others had little to no real meaning.

### 0.3.1 Analysis 1: Days of the Week

There was honestly very little to be gained from this graph. The most crimes happened on Fridays, whereas the least crimes happened on Sundays, presumably because all the criminals were too busy in church to be out on the streets committing crimes. In all seriousness, the distribution among days of the week was fairly even, and nothing really stood out as a common day of the week for crimes to occur.

### 0.3.2 Analysis 2: Crimes v. AirBnB Metrics

A lot of these were pretty hit or miss - while many displayed a downward trend as crimes went up, there were quite a few outliers. For example, the crimes v. number of AirBnB listings graph had a solid downward trend, but there was one very strong outlier in the middle of the graph, presumably because that area had a lot of listings at a below-average price due to the number of crimes in the area. This probably means that while the area is generally safe, the rather high crime rate makes for an attractive price, thus benefitting both the property owner as well as the tenant.

The crimes v. ratings graph showed no real correlations at first, but as crimes went up, the ratings went down. This makes sense because there's obviously a lot more that goes into a user's experience in an AirBnB besides the crime rate in the area. However, as the crimes become ridiculously high, chances are they had an effect on the tenant's experience.

The availability graph had the best correlations, where as the crimes grew, so did the availability. This really showed that people look at the area before renting out an AirBnB, so the more in advance you book an AirBnB, the high-crime areas are the ones that are the most available. These tend to go out pretty soon though, so as time goes by, people might get stuck with the higher-crime AirBnBs, which is what may have affected the other graphs.

### 0.3.3 Analysis 3: Density maps - shooting vs. non-shooting

The first heatmap shown includes all crimes in the greater Boston area. Unfortunately, the map library used in this project, mplleaflet, does not support heatmaps, so one's left to assume where things are in the basic matplotlib representation. What's interesting, however, is that when shooting is the only metric to plot by, the crimes go from central Boston/NW Boston to more Eastern and Southern Boston.

### 0.3.4 Conclusion

While crimes may not have an effect on a traveler's experience in Boston, they definitely have an effect on how a traveler plans his/her trip to Boston. The availablity plots show this best - there's a high correlation between number of crimes in an area and the AirBnB availability; as crimes go up, less people want to book AirBnBs in that area. However, measuring things like square footage is completely irrelevant because square footage of an AirBnB is usually a nonfactor when deciding where to book an AirBnB unless the place is heinously small or overpriced. This means that price is typically a better thing to analyze, along with the ratings of the AirBnB to see what a traveler's experience will be like, however these price and ratings are typically codependent on the experiences of both the tenant and the property owner. Thus, all we can say is that the more crimes in an area, the less people will want to live there during their time in Boston. Obvious? Maybe. Proven before? Maybe not so much.

```
In [ ]:
```