We can find tons of free data available on the internet. Most of these data are clean, properly structured, and ready for analysis. But, I am interested in taking care of the complete raw data. When I search data for my project, I find the most challenging part is finding the correct raw data. In this project, I plan to build the data myself. I learned about web scraping and pulled the listings of the housing rental price on Craigslist. Craigslist is an American advertisement website where people can post for sections such as jobs, housing, personals for sale, items wanted, services, community, resumes, and discussion forums. Among its many benefits, it is famous for housing advertisements. It is on the hotlist for people searching for a house.

Among the two different ways of web scraping 1) Selenium 2) Beautiful Soup, I used the code from the library BeautifulSoup. It can help to obtain the wholesome data from the web page at once. I need to figure out the structure of the html code to obtain the relevant information from the wholesome data we pull.

At first I tried to make a list of the links of the listings from the first page of the Cragilist housing search. I find more than 2000 links which are individually used in a loop to open and pull the different features of the listings.

Features like:

No\_of\_bathrooms', 'No\_of\_bedrooms', 'latitude', 'longitude', 'pet', 'laundry', 'parking', 'floor', 'Apartment\_type', 'smoking', Neighbourhood,

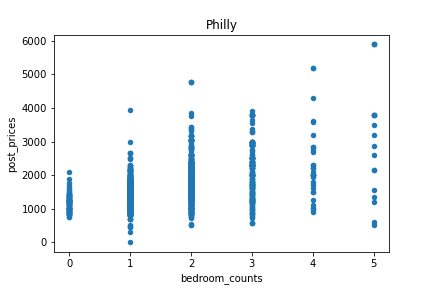
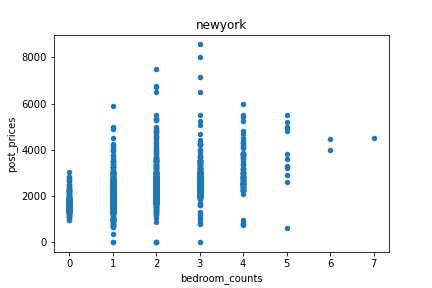
We were careful in distinguishing the listings as the Multi, Single and Duplex housing types. The singles category consists of cottage/cabin, house and townhouse. Duplexes have double, in law and the rest are in Multi. We did similar categorizing for laundry and parking to group similar categories into broad one. This image visualizes clearly the different features we were able to extract.



Since this data is completely raw, I find it an opportunity to learn data cleaning and data wrangling. I have to deal with missing values for different features. In most cases I find a better option to replace the missing values than to drop. Like for area sq fts, I used the mean obtained from the grouped bedroom and the apartment type. In latitude and longitude I try to replace it with similar neighbourhoods.

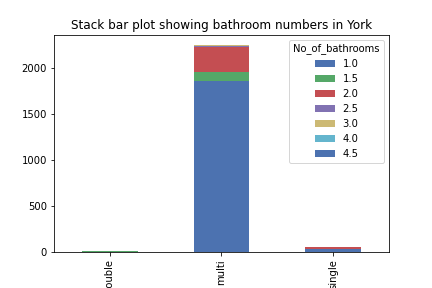
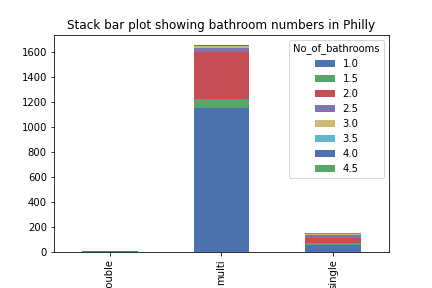
For bedroom counts, I try to replace it with the values in post title text.

I also take care of the outliers in the post price and area(sq fts).



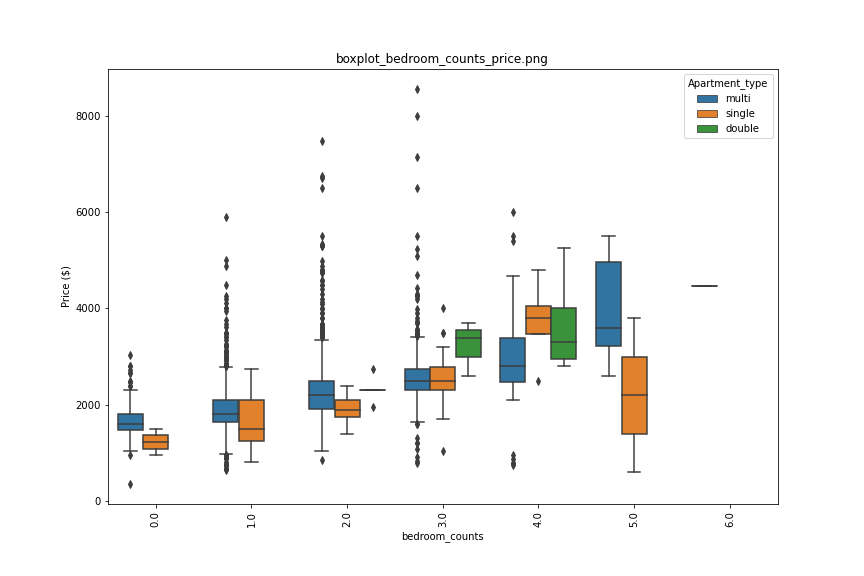
In the picture we can clearly see some housing prices are less than 5$ even after removing the highest values. So we found the price in the post title text for these missings so I manually replaced them since they were only a few. Finally I also dropped the columns with missing values greater than 50%.

I need to take care of the number of bathrooms. The links and the title tells few shareBa, these are the listings for renting the room but not any apartment or housing. Since we are dealing with an apartment or house, I drop these room rent listings.



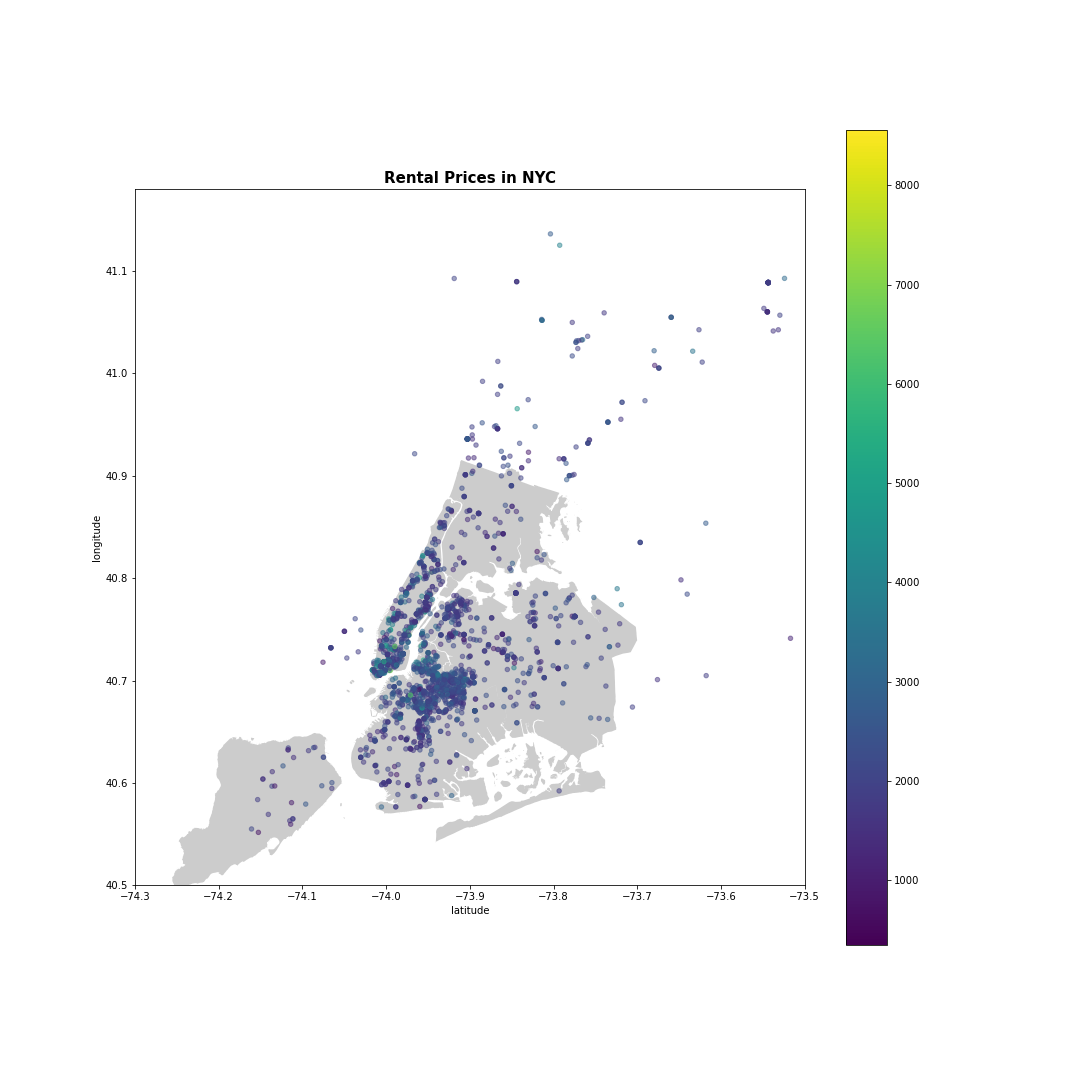
I try to see the number of bathrooms for different apartment types. The multi and single type housing listings have shown variation in the bathroom numbers. Must multi has 1 bathroom and it is followed by 2.

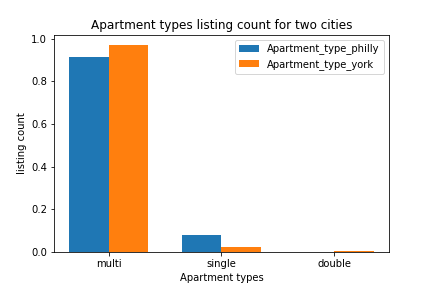
I also replaced the studio room with 0 bedroom, many posts I found as 1 bedroom.



Here is the boxplot showing the price variation for different bedroom counts for different apartment types.We see the huge variation on the 3 and 2 bedroom for the multi type, and least on the 0 bedroom the studio type. This also helps to compare the median of the price and the inter quartile range.

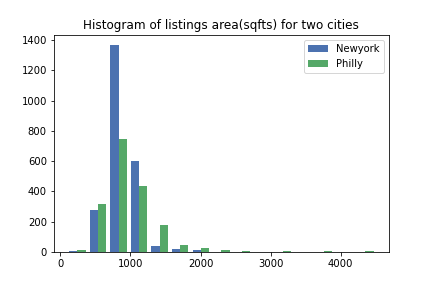
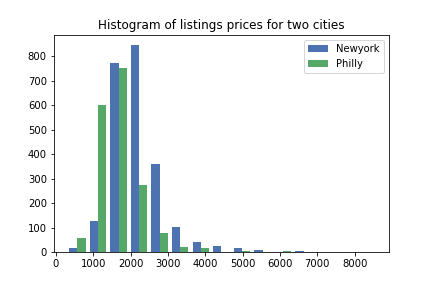
The plot below shows the distribution of listings and their variation in prices in Newyork city map. We find the listings of prices more than 7000 concentrated in expensive place like Manhattan in Newyork.



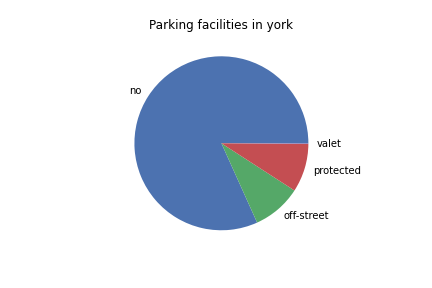
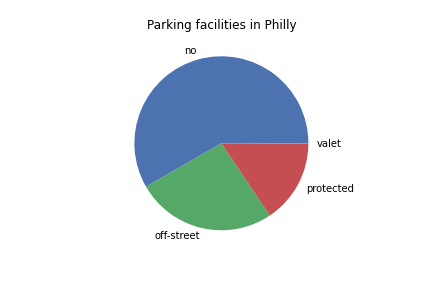


Here I explore the listings count for different types of housing. There is bias in our data, most of them are in the multi type more than 80% while less percentage in single and double has the smallest percentage less than 1.

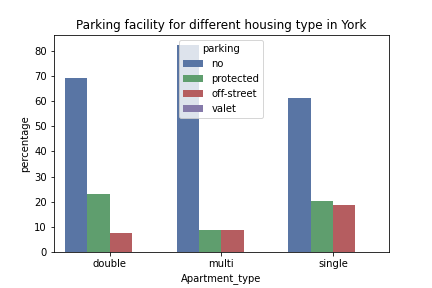
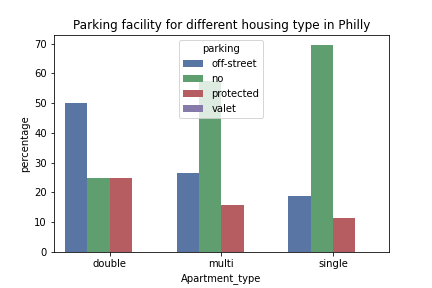
I am interested to see the histogram for the listings for two different cities based on the listings prices. It is difficult to come to any conclusion for two different cities. But comparatively the Newyork price is higher than Philadelphia. I also try to build the histogram of the area in sq ft. This is just opposite to post prices. The sq fts in Newyork listings is smaller than the Philadelphia.



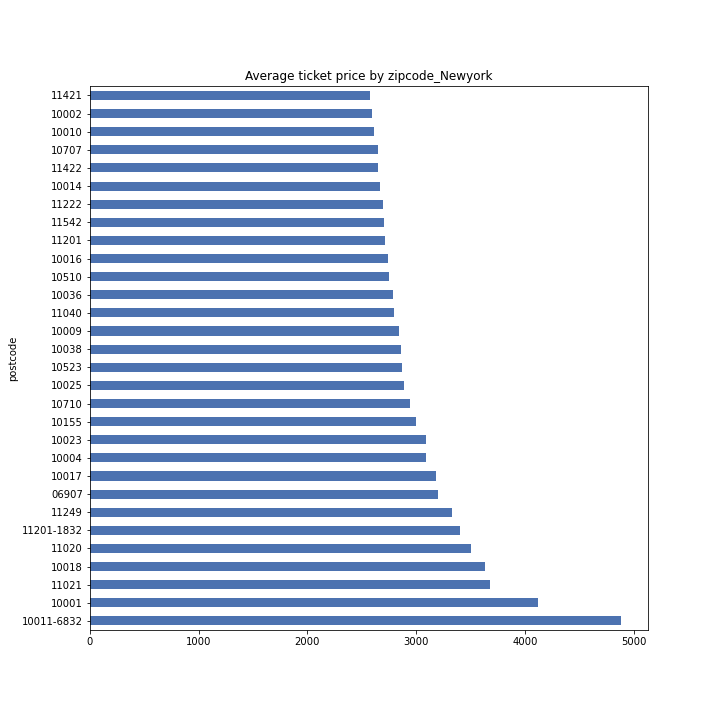
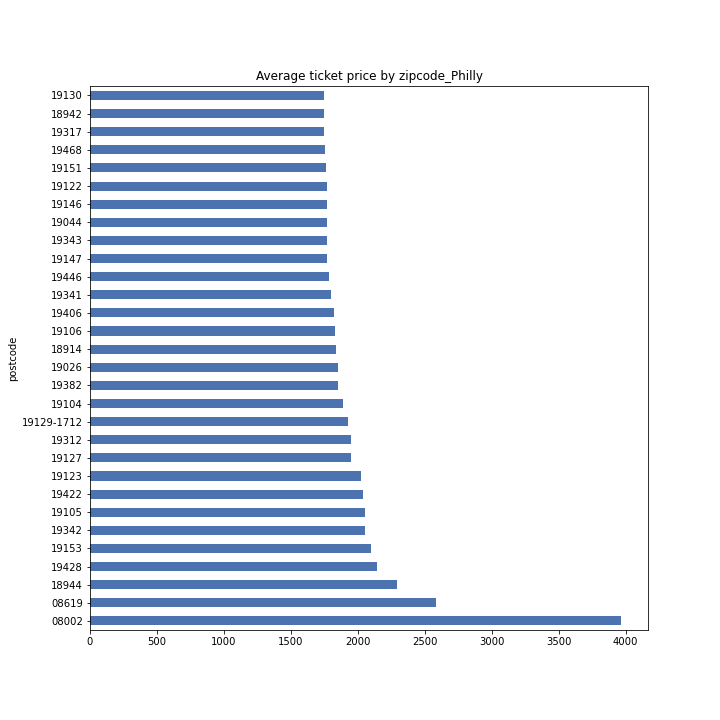
In the pie plot most of the listings don't have parking facilities in both the cities. The second percentage of listings has off-street parking and only a few have protected parking facilities.

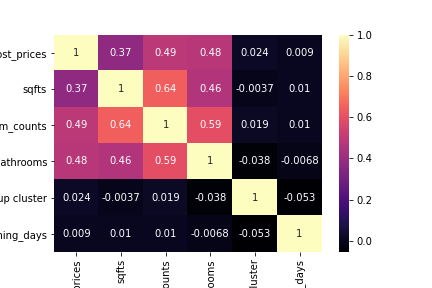
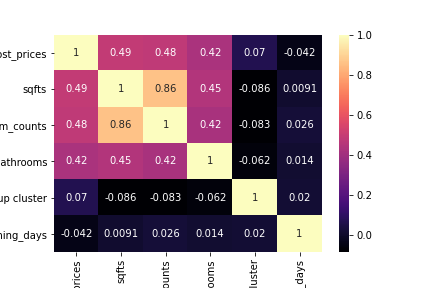


I want to see the parking facility for different apartment types. Single and Multi types are among the biggest percentage without parking. Both cities have similar variations of the parking facilities for different housing types.



I want to visualize the variation in the post prices as per the zip code. We don’t see the big difference in Phildelphia and Newyork except the few exceptions .





The plot above shows the heat map of the correlation coefficient among the different features. The post prices have a positive medium correlation with sqfts, bedrooms and bathrooms but it didn’t show with the cluster we created. Same is the case with the sqfts.

I used these features 'sqfts', 'bedroom\_counts', 'No\_of\_bathrooms', 'pickup cluster', 'post\_timing\_days','pet', 'laundry', 'parking', 'Apartment\_type', 'postcode' to predict the post prices.

I tried different models in search of the highest accuracy. After the label encoding categorical data ,I tried the simplest LinearRegression model. Later I used Bayesian and XGBoostRegressor and calculated their respective accuracy and the mean squared error.

I tuned the hyper parameter using Grid Search and used the best parameters to build the model. RandomForestRegression gave the best result.