SOUTHERN ILLINOIS UNIVERSITY EDWARDSVILLE

CMIS 567 Business Analytics Capstone

Business Intelligence Capstone Project Report

Understanding Airbnb Data

Author:
Luis Barrios
MS CMIS Student

Professor:
Dr. Joseph Vithayathil

May 5, 2021

Table of content

Executive Summary

The purpose of this report is to show the processes used to mine a dataset with listings in Boston, Massachusetts, from the online company Airbnb. It starts with background information explaining how the company operates and makes money from providing a free listing space on their website to a host in any part of the world, and how the company is changing the tourism industry become a new trend for travelers. During this analysis, the data was collected from the internet on a public domain website that scratches the last twelve months of listings.

The analysis is divided into two main sections visual data analysis and prediction analysis. Tableau was the primary tool to show the relationship between price and location, concluding with higher prices on listings located in high-density areas surrounding downtown and Boston Common City Park. Also, this part of the analysis has shown that hosts tend to publish the same property based on location multiple times, leading to corruption in the data because of duplicates. The last part of the analysis concludes using Jupiter Notebook to run Python scripts to clean the non-corrected dataset with more attributes to get a high-quality model. However, the data cleanse process revealed that the data is not high in quality, and it has too many missing values and outliers that limit the prediction model. Therefore, to obtain an accurate prediction model, more consistent data must be obtained from the listings in the city.

Airbnb: The New Way to Travel

Airbnb is a public-hold online company that started in 2008 when a couple of friends were trying to earn some extra money by renting a room of their apartment per night in a similar way to a hotel. Since then, the company has been growing exponentially, adding tons of new listings every year and becoming a top choice for travelers around the world. The company works on a straightforward premise where everybody could be a host. People with some extra space in their house or people with extra properties could list their space to anyone around the world who wants to stay for a night or even stay for a couple of months. The website offers a wide selection of properties that go from a shared room to entire houses and even castles. The hosts do not have to pay for their listing, and they set how much they want to charge per night, per week, or month if it is available. Therefore, the website made its revenue through a service fee from the bookings.

On the other hand, the new traveling format has been growing because it offers more benefits to travelers than hotels. Guests found more value in the price paid for an Airbnb room with better accommodations than spending the same price for a smaller space at a hotel. Also, they communicate directly with the host (property owner), which provides a faster and more efficient customer service interaction. There is a larger supply with a listing that could be located at a more convenient location for the guest with more flexibility during check-ins and check-outs. In general, the benefits of Airbnb over hotels are countless, and it is driving the tourism market in their way.

Inside Airbnb

The purpose of this analysis is to get a deeper view of the data behind all the listings on the Airbnb website. Understanding the listing demand and supply could lead to helpful ideas in managing a listing from a host's point of view or where to book from a guest's point of view. In this case, the analysis focuses on Airbnb data from Boston, Massachusetts. Data was obtained from the website insideairbnb.com that contains public records from the last twelve months of the listings in several cities around the world. The data is divided into two tables, "Listings" and "Listings_Summary."

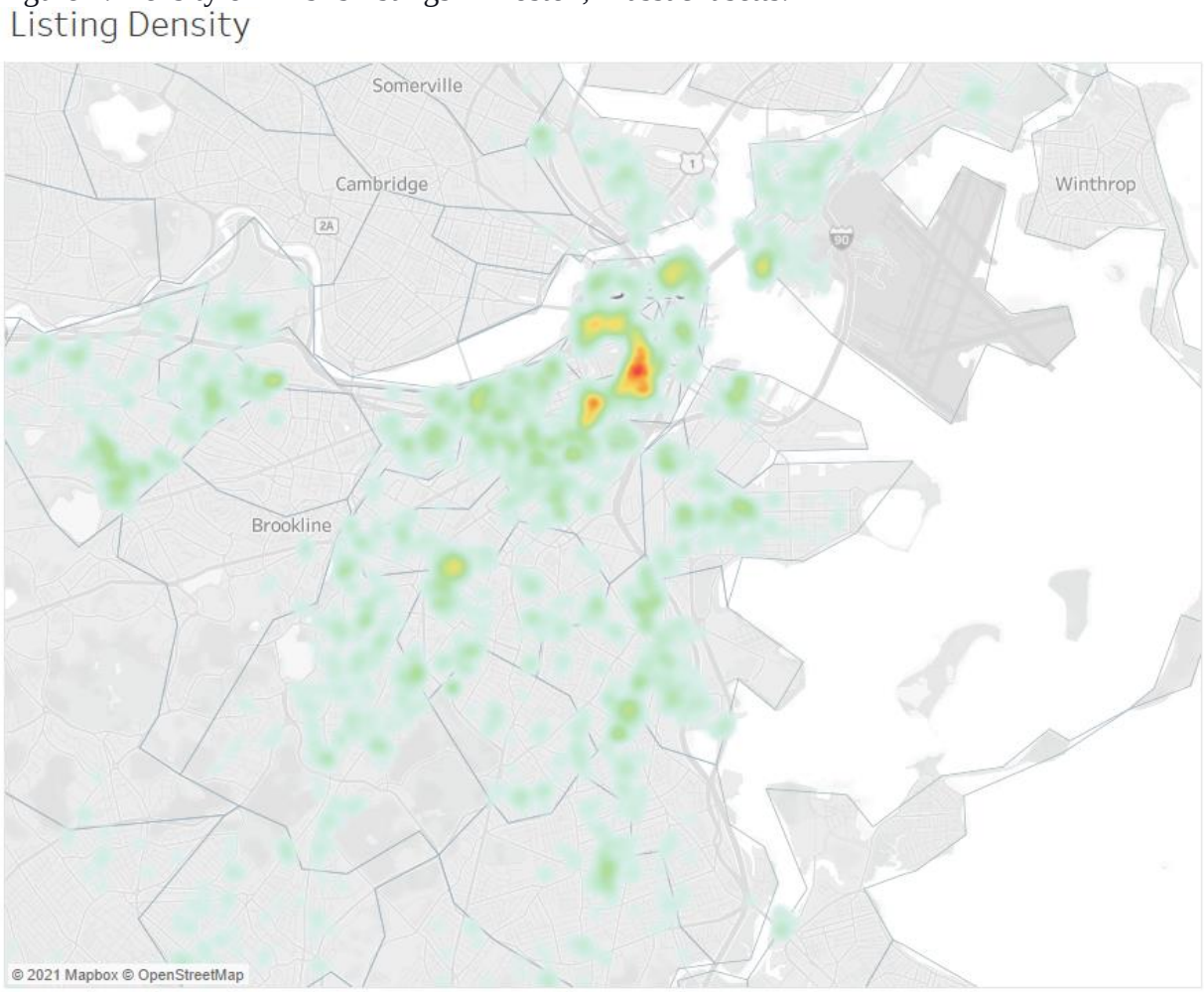
Visual Data Analysis

This section consists of visualizing the data to gain preliminary insights about the current listing status in the city using the table "Listing_summary." The dataset does not require a previous cleanse, and it has approximately 3100 listings from 2021 spread around the city in different neighborhoods with different prices. The next couple of visualizations are going to show the current behavior of hosts and their listings. Some of the visualizations would provide answers to some of the business questions.

What is the most popular listing place in the city?

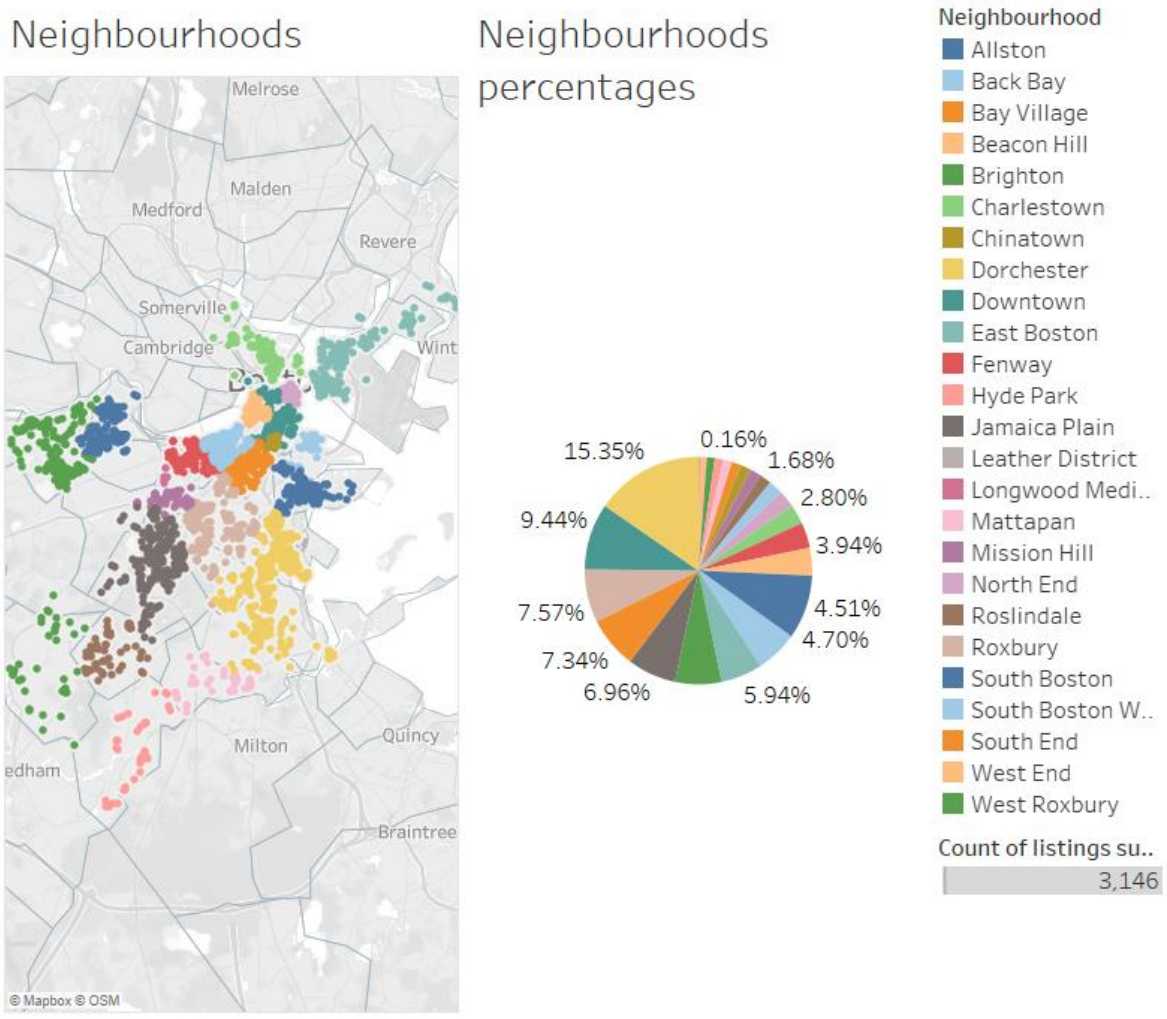
Tableau allows for pinning each listing on the map using the geographic latitude and longitude of the listings. It also provides the option to change the individual point to density points with color. Thus, downtown has the highest density of listings closed to each other. There is a clear pattern of heat that surrounds the park Boston Common, one of the most iconic tourist places in the city. There is a high concentration of listings in Downtown, Chinatown, Bay Village, Beacon Hill, West End, and North End (See figure 1.)

Figure 1: Density of Airbnb listings in Boston, Massachusetts.



However, the density of the area does not mean that those areas control the market share. The highest quantity of listings are located at Dorchester on the southeast side of the city, with 15.35 percent of the total listings. Then, it is followed by Downtown with 9.44 percent and Roxbury with 7.57 percent (See figure 2.)

Figure 2: Neighborhood distribution and total percentage



What are the most expensive areas?

The average price per neighborhood is aligned with the listing density. Tableau can create a calculated measure containing the average price based on neighborhood. Most of the areas close to Downtown have a higher price due to the high demand in the zone. See Figure 3. At the same time, the price varies according to the listing type; in general, entire apartments would be more expensive than single bedrooms avoiding some exceptions. See Figure 4.

Figure 3: Heatmap Average Prices by Top 20 Neighborhoods

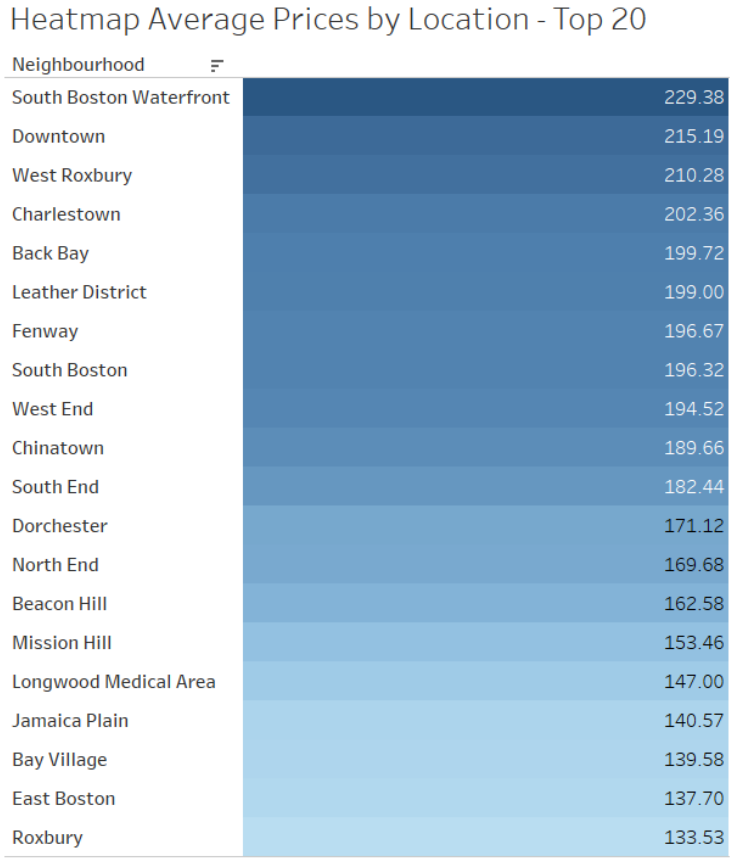
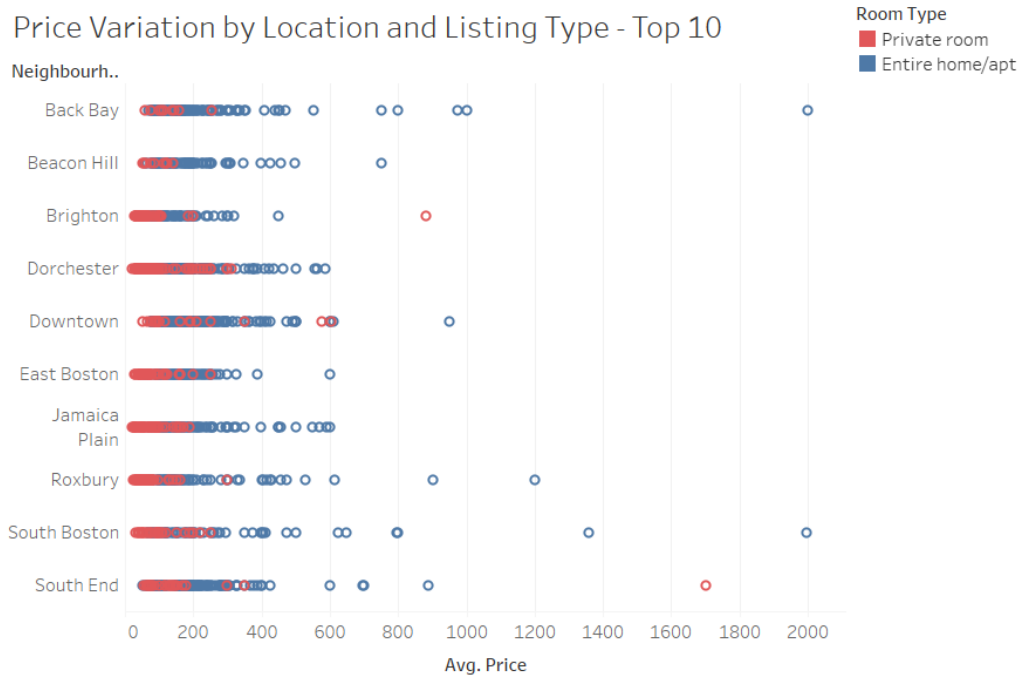


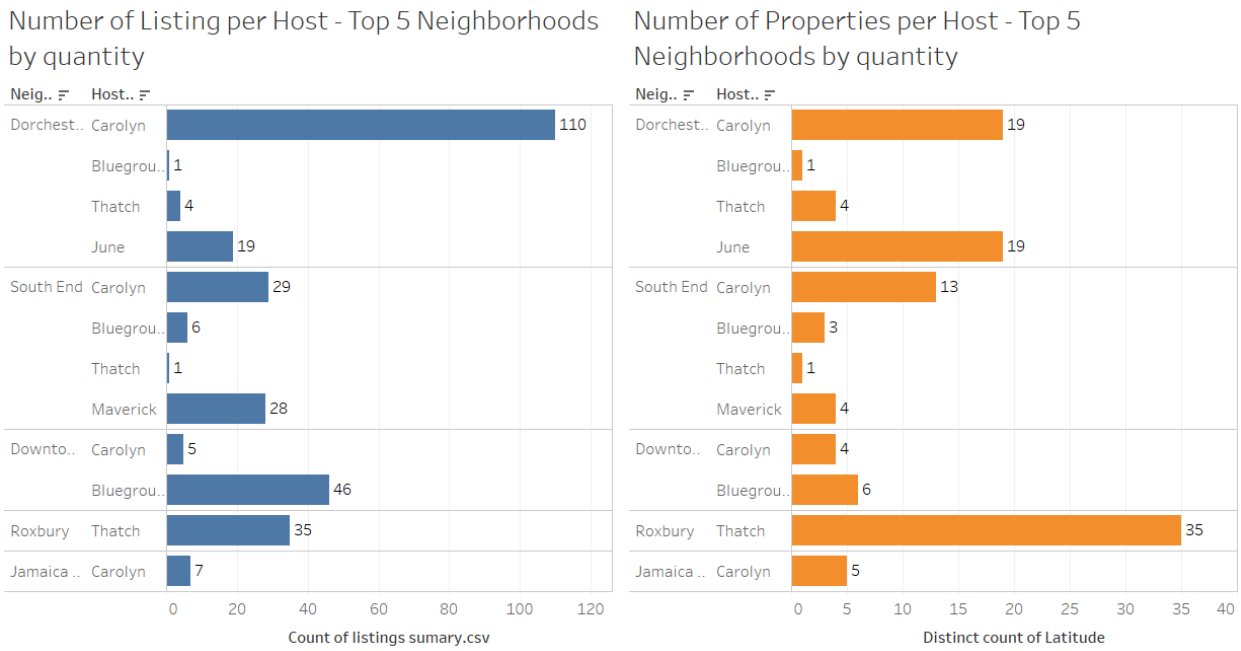
Figure 4: Price Variation by Location and Listing Type – Top 10



How is the behavior of hosts related to the listings?

The quality of the data could be affected due to the hosts' effort to get more reservations. Hosts are not charged for their listing, and some of them may be abusing this benefit. Hosts may have realized that statistically speaking, more listings lead to more reservations. Some of the hosts have been listing multiple times the same space with some little changes on the specifications in order to increase their presence on the website. Figure 6 shows how many listings a host has on the top 5 neighborhoods compared with the actual number properties based on repeated geographical coordinates for the listing. In other word, figure 6 shows how the host Carolyn has 110 listings from only 19 units available.

Figure 6: Number of listings compared with actual number of properties.



Data Prediction Analysis

This section consists of cleaning and mining the table "Listing" because it contains a wide range of attributes that could help predict future listing prices. At the same time, this section is divided into two subsections based on process used during the analysis. Jupiter Notebook is the primary tool used to run Python scripts to mine the dataset and several Python libraries.
Libraries used:

- Pandas for data manipulation.
- Matplotlib to create static visualizations.
- Numpy for data manipulation.
- Statsmodels to create linear regression modelT.

Data Preparation

The dataset has a total of 73 attributes, but some of them are missing too many values, and they could affect the outcome. Any attribute with a count of 20 percent of null values is dropped from the dataset in order to maintain data quality.

- The attribute "Room_Type" contains four rooms: Entire home/apt, Private room, Hotel room, Shared room. Hotel and shared room are dropped from the analysis due to their low volume.
- The attribute "Price" is shown as an object data type. For further manipulation, any values that were not a number are dropped. The remaining were converted to strings by using the .astype(int) function. Because the remains have two additional zeros, the column is divided by 100 to get the exact price.
- The bedroom attribute is shown as float data type. It is converted to a string by dividing the value count by the total count. Then, any value that equals zero is dropped from the data set. Null values are set to one to maintain the attribute in the analysis.
- Columns with a price above 2000 are removed to avoid outliers.

Linear Regression

Statsmodel is imported as a library to the Python script and then calls the OLS methods to create a linear regression with the already clean attributes. The analysis will be using the number of accommodates, the number of bedrooms and the number of reviews compared to their relationship with price as the dependable variable. The X values are assigned to the attributes accommodates, bedrooms, and the number of reviews; The Y values are given to the price attribute. Then, the code is run, and it provides the following output: (See Figure 7)

Figure 7: Linear Regression Matrix Results

OLS Regression Results

Dep. Variable:	price	R-squared (uncentered):	0.682
Model:	OLS	Adj. R-squared (uncentered):	0.682
Method:	Least Squares	F-statistic:	2218.
Date:	Sun, 02 May 2021	Prob (F-statistic):	0.00
Time:	17:36:27	Log-Likelihood:	-19131.
No. Observations:	3104	AIC:	3.827e+04
Df Residuals:	3101	BIC:	3.829e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
accommodates	23.7803	1.651	14.400	0.000	20.542	27.018
bedrooms	52.9073	3.859	13.710	0.000	45.341	60.474
number_of_reviews	-0.0709	0.028	-2.519	0.012	-0.126	-0.016

Omnibus:	3498.595	Durbin-Watson:	1.745
Prob(Omnibus):	0.000	Jarque-Bera (JB):	566129.800
Skew:	5.528	Prob(JB):	0.00
Kurtosis:	68.231	Cond. No.	161.

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The R-squared represents the fit of the model from 0 to 1. A high number indicates less error and a good fit for the model. In this case, 68.2 percent is a bad number from a conservative point of view. Adding the fact, R-squared is uncentered only explains the non-varying part of Y, which means the model's accuracy equals 0. The coefficients at least propose a glue in how the attributes affect the price. A high coefficient (even negative) means a high relationship with the dependent variable. However, the standard error provides a level of accuracy to the coefficients; low standard errors mean a high coefficient accuracy. Therefore, in this model coefficient of the number of reviews is the most accurate attribute in relationship with price, but still meaningless due to the low value. The model needs more quality information in order to work. The dataset used for the analysis has further too many missing values and several outliers, leading to inaccurate predictions.

References

Barrios, L. (n.d.). Airbnb Boston Visualizaciones. Retrieved from https://public.tableau.com/profile/luis.barrios6890#!/vizhome/AirbnbBoston_16197432101360/Densit

Cox, M. (n.d.). *"Listings Summary" Boston, Massachusetts, United States*. Retrieved from http://data.insideairbnb.com/united-states/ma/boston/2021-04-20/visualisations/listings.csv

Cox, M. (n.d.). *"Listings" Boston, Massachusetts, United States*. Retrieved from http://data.insideairbnb.com/united-states/ma/boston/2021-04-20/data/listings.csv.gz

Appendix

Code use for the analysis part I

In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

airbnb = pd.read_csv('listings.csv')

airbnb.info()

airbnb.room_type.value_counts()

#Dropping Shared rooms because of their low volume
airbnb = airbnb[airbnb.room_type != 'Shared room']

#Dropping Shared rooms because of their low volume
airbnb = airbnb[airbnb.room_type != 'Hotel room']

# Shared Room and Hotel room is dropped
airbnb.room_type.value_counts()

#convert price from object to integer
print (airbnb['price'])
airbnb['price'] = airbnb['price'].replace('[\$\,\.]','',regex=True).astype(int)
airbnb['price'] = airbnb['price']//100

airbnb.bedrooms.value_counts()/airbnb.bedrooms.count()

# Dropping values where bedrooms are equal to zero
airbnb = airbnb[airbnb.bedrooms != 0]

#Values without zero bedrooms
airbnb.bedrooms.value_counts()

plt.hist(airbnb.price,bins=20,range=(0,1000))

plt.hist(airbnb.accommodates,bins=16)

plt.scatter(airbnb.price, airbnb.number_of_reviews)

plt.scatter(airbnb.accommodates,airbnb.price)

#Plotting neighborhood count
airbnb.neighbourhood_cleansed.value_counts().plot(kind='bar',figsize=(20,5))
```

Code use for the analysis part II

In [ ]:

```
# Removing rows which are above Price range of 2000
airbnb[airbnb.price > 2000].count()
airbnb = airbnb[airbnb.price <2000]

airbnb['bedrooms'].fillna(value=1,inplace=True)

X = airbnb.iloc[:,[33,36,55]]

X.head()

Y = airbnb.price

display(X.shape,Y.shape)

# Creating linear regresssion model
import statsmodels.api as sm

model = sm.OLS(Y, X).fit()

predictions = model.predict(X)

print_model = model.summary()

print(print_model)
```