AIRBNB, NYC

25-20-21

METHODOLOGY

A document attached in the appendix of both the presentations that showcase the methodology that was undertaken for the analysis along with all the steps/codes that were performed in detail.

Python Codes:

1) Removing Price outliers

- Max price in the dataset was $10000, 99th percentile was $799, by looking at the dataset it can be said that in New York $799 for an AirBnb room is possible bur above $1000 can be considered as an outlier because there only 7 properties with price more than $1000.

- df.price.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])

```
In [8]: df.price.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])
```

```
Out[8]: count    48895.000000
        mean       152.720687
        std        240.154170
        min          0.000000
        1%          30.000000
        10%         49.000000
        20%         60.000000
        30%         75.000000
        40%         90.000000
        50%        106.000000
        60%        130.000000
        70%        155.000000
        80%        200.000000
        90%        269.000000
        93%        300.000000
        95%        355.000000
        97%        450.000000
        99%        799.000000
        max      10000.000000
        Name: price, dtype: float64
```

- df = df[df.price < 800]

```
In [9]: #Removing the outlier
        df = df[df.price < 800]
```

```
In [11]: df.price.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])
```

```
Out[11]: count    48421.000000
         mean       137.543917
         std        103.789003
         min          0.000000
         1%          30.000000
         10%         49.000000
         20%         60.000000
         30%         75.000000
         40%         90.000000
         50%        105.000000
         60%        129.000000
         70%        150.000000
         80%        199.000000
         90%        250.000000
         93%        300.000000
         95%        345.000000
         97%        400.000000
         99%        550.000000
         max        799.000000
         Name: price, dtype: float64
```

2) Removing Minimum Nights outlier

- df.minimum_nights.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])

```
In [12]: 1 df.minimum_nights.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])

Out[12]: count    48421.000000
         mean         6.979596
         std         20.291590
         min          1.000000
         1%           1.000000
         10%          1.000000
         20%          1.000000
         30%          2.000000
         40%          2.000000
         50%          3.000000
         60%          3.000000
         70%          4.000000
         80%          6.000000
         90%         28.000000
         93%         30.000000
         95%         30.000000
         97%         30.000000
         99%         40.000000
         max       1250.000000
         Name: minimum_nights, dtype: float64
```

- df = df[df.minimum_nights < 61]

```
         5 df = df[df.minimum_nights < 61]

4]:      1 df.minimum_nights.describe([.01,.1,.2,.3,.4,.5,.6,.7,.8,.9,.93,.95,.97,.99])

ıt[14]: count    48107.000000
        mean         5.883447
        std          8.877846
        min          1.000000
        1%           1.000000
        10%          1.000000
        20%          1.000000
        30%          2.000000
        40%          2.000000
        50%          2.000000
        60%          3.000000
        70%          4.000000
        80%          6.000000
        90%         21.000000
        93%         30.000000
        95%         30.000000
        97%         30.000000
        99%         30.000000
        max         60.000000
        Name: minimum_nights, dtype: float64
```

- In the end committing the new data to a csv file .

```
: 1 df.to_csv(r'C:\Users\shiva\Downloads\AB_NYC_2019.csv')
```

- The given data was explored using pandas library in Python and all different numerical columns were explored to find any outliers <insert code block from python.

- The outliers were treated and 1.6% of the dataset was removed to avoid any bias when further exploring the dataset

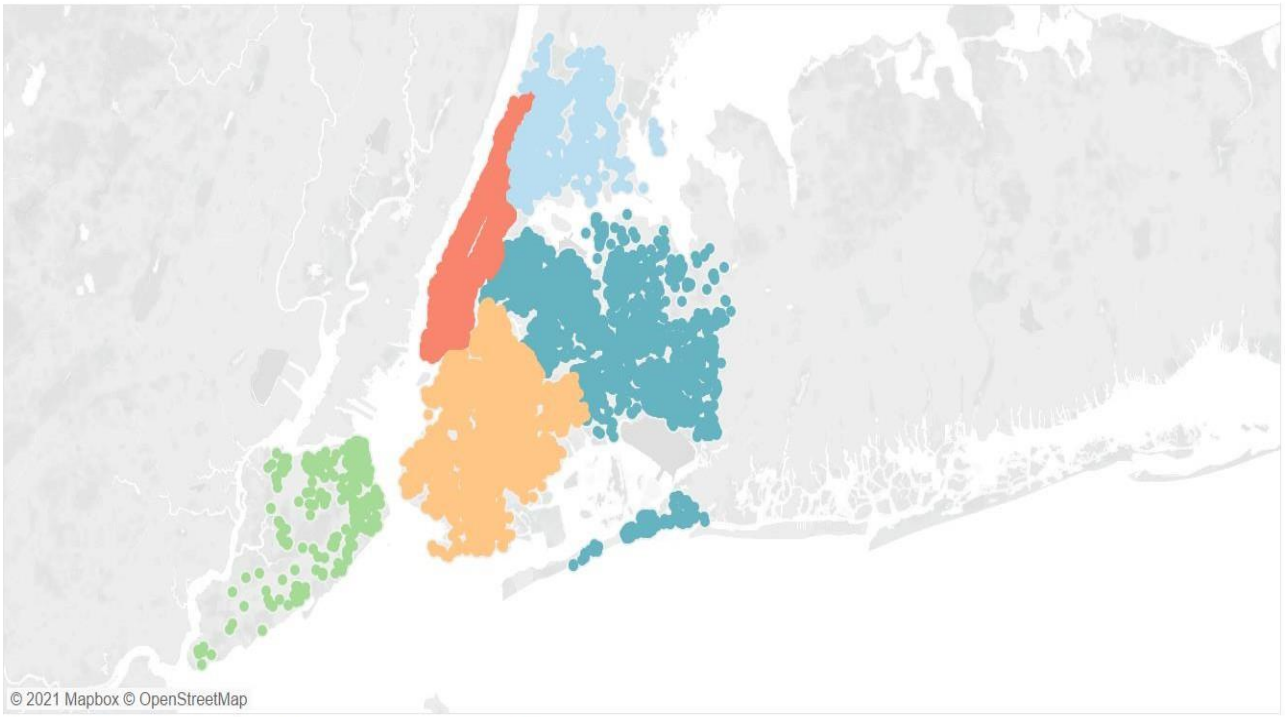
- This new dataset was then fed into Tableau to perform EDA and derive meaningful insights

A few assumptions were made in this case study:

- There is no column that shows the number of bookings, hence it is assumed that no. of reviews against a listing is the no. of bookings made.
- It was also assumed that outliers in the dataset was an anomaly and that such prices or minimum number of nights is impossible and hence those data were removed in the data cleaning process
- No. of listings has been created based on whether the listing has a name or not and those which did not have a name were not available
    - It is assumed that the listing is not available any more.
- It is assumed that the company does not want to expand beyond the 5 territories it already has listings

Graphs that were not part of the Presentations

Neighborhood



Map based on Longitude and Latitude. Color shows details about Neighbourhood Group. The view is filtered on Neighbourhood Group, which keeps Bronx, Brooklyn, Manhattan, Queens and Staten Island.

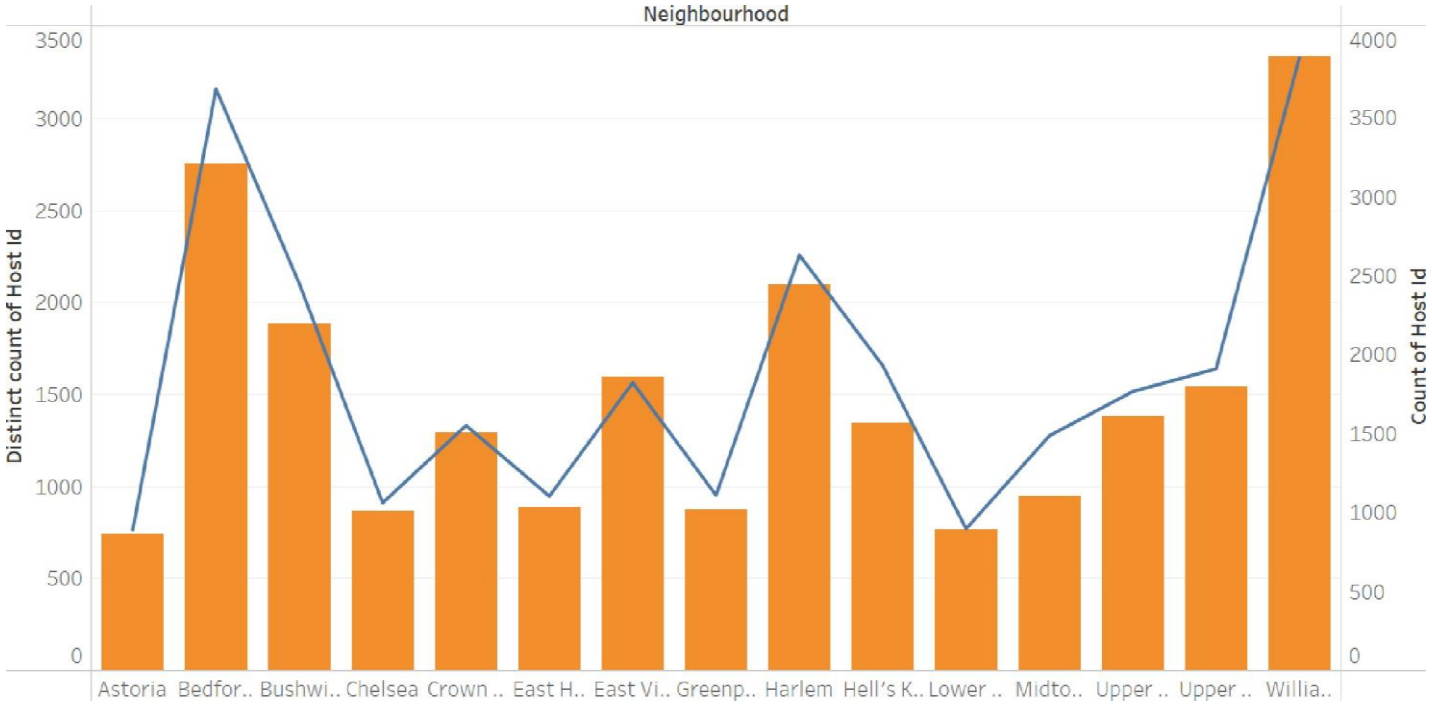
.

Avg Price per Room

	Neighbourhood Group				
Room Type	Bronx	Broo..	Manh..	Quee..	State..
Entire home/apt	126.4	165.5	217.3	140.9	131.3
Private room	63.1	72.1	106.6	67.8	62.3
Shared room	47.6	50.4	84.7	49.4	57.4

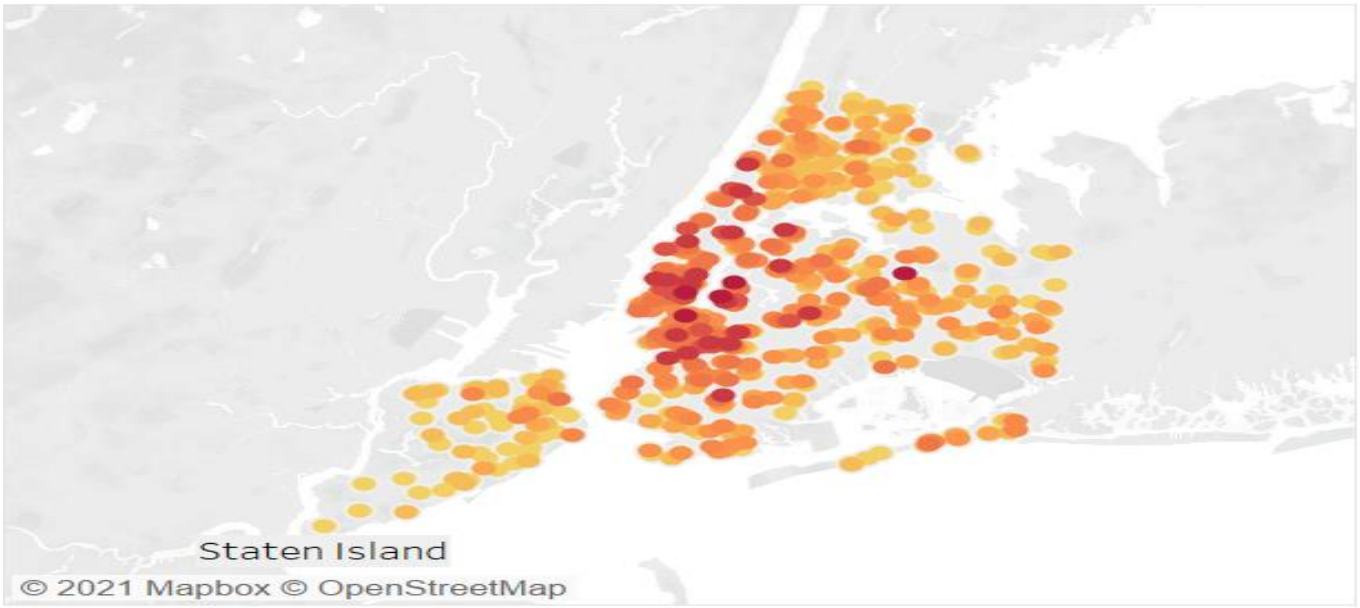
Avg. Availabilit..
64.8 226.4

Average of Price broken down by Neighbourhood Group vs. Room Type. Color shows average of Availability 365. The marks are labeled by average of Price.



No. of properties vs host

Last review



Map based on average of Longitude and average of Latitude. Color shows details about Last Review Year. The marks are labeled by Neighbourhood Group. Details are shown for Neighbourhood. The view is filtered on Last Review Year, which excludes Null.

Year of Last Review

- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018
- 2019