Storytelling Case Study: Airbnb, NYC

Methodology Document

- **Problem Background:**

  Suppose that you are working as a data analyst at Airbnb. For the past few months, Airbnb has seen a major decline in revenue. Now that the restrictions have started lifting and people have started to travel more, Airbnb wants to make sure that it is fully prepared for this change.The different leaders at Airbnb want to understand some important insights based on various attributes in the dataset so as to increase the revenue.

- **Objective:**

  To prepare for the next best steps that Airbnb needs to take as a business, you have been asked to analyze a dataset consisting of various Airbnb listings in New York.

- **Introduction to Airbnb_Dataset:**

  Columns Description:

Column	Description
id	listing ID
name	name of the listing
host_id	host ID
host_name	name of the host
neighbourhood_group	location
neighbourhood	area
latitude	latitude coordinates
longitude	longitude coordinates
room_type	listing space type
price	
minimum_nights	amount of nights minimum
number_of_reviews	number of reviews
last_review	latest review
reviews_per_month	number of reviews per month
calculated_host_listings_count	amount of listing per host
availability_365	number of days when listing is available for booking

Dataset Description

❖ Step 1: Loading Libraries ,Reading dataset & Data Cleaning in Python

We will start our initial analysis process with loading important libraries, reading the dataset,checking statistical information & performing data cleansing activities (handling missing values, outliers etc) in python.

Loading libraries:

Loading Libraries

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

        import warnings
        warnings.filterwarnings('ignore')
```

```
In [2]: pd.set_option('display.max_rows', 500)
        pd.set_option('display.max_columns', 500)
        pd.set_option('display.width', 1000)
```

Reading Dataset and checking statistical information:

Reading csv

```
In [3]: airbnb_data = pd.read_csv(r'/Users/shikhargoyal/Documents/Shikhar/Ug/Airbnb/AB_NYC_2019.csv')

        print(airbnb_data.shape)
```

```
(48895, 16)
```

```
In [4]: # checking the data types

        airbnb_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              48895 non-null  int64
 1   name                            48879 non-null  object
 2   host_id                         48895 non-null  int64
 3   host_name                       48874 non-null  object
 4   neighbourhood_group             48895 non-null  object
 5   neighbourhood                   48895 non-null  object
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object
 9   price                           48895 non-null  int64
 10  minimum_nights                  48895 non-null  int64
 11  number_of_reviews               48895 non-null  int64
 12  last_review                     38843 non-null  object
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64
 15  availability_365                48895 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

```
In [5]: # checking the statistical info

        airbnb_data.describe()
```

Out[5]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listing
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327

Data Cleaning & Handling missing values:

Handling Missing values

In [6]:
```
# Checking the null values

(100*(airbnb_data.isnull().sum())/len(airbnb_data.index))
```

Out[6]:
```
id                                0.000000
name                              0.032723
host_id                           0.000000
host_name                         0.042949
neighbourhood_group               0.000000
neighbourhood                     0.000000
latitude                          0.000000
longitude                         0.000000
room_type                         0.000000
price                             0.000000
minimum_nights                    0.000000
number_of_reviews                 0.000000
last_review                      20.558339
reviews_per_month                20.558339
calculated_host_listings_count    0.000000
availability_365                  0.000000
dtype: float64
```

name, host_name, last_review , reviews_per_month columns have missing values which we need to impute

In [7]:
```
# filtering column names having missing values
L1 = []
for i in airbnb_data.columns:
    if (100*(airbnb_data[i].isnull().sum())/len(airbnb_data.index)):
        L1.append(i)

L1
```

Out[7]: `['name', 'host_name', 'last_review', 'reviews_per_month']`

In [8]:
```
#checking the null values in for name column

df1 = airbnb_data[airbnb_data['name'].isnull()]
df1
```

Out[8]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_review
2854	1615764	NaN	6676776	Peter	Manhattan	Battery Park City	40.71239	-74.01620	Entire home/apt	400	1000	
3703	2232600	NaN	11395220	Anna	Manhattan	East Village	40.73215	-73.98821	Entire home/apt	200	1	2

Imputing the missing values

In [9]:
```
# As we don't have the name of the listing we will be filling the null values with "Not available"

airbnb_data['name'].fillna('Not available', inplace=True)
```

In [10]:
```
# checking if the null values have been imputed
# No missing values found

airbnb_data['name'].isnull().sum()
```

Out[10]: `0`

In [11]:
```
# As we don't have the name of the host we will be filling the null values with "Not available"

airbnb_data['host_name'].fillna('Not available', inplace=True)
```

In [12]:
```
# checking if the null values have been imputed
# No missing values found

airbnb_data['host_name'].isnull().sum()
```

Out[12]: `0`

In [13]:
```
can't impute the missing values for last_review column as the customer might not have given the rating so we can impu

_data['last_review'].fillna('', inplace=True)
```

In [14]:
```
# checking if the null values have been imputed
# No missing values found

airbnb_data['last_review'].isnull().sum()
```

Out[14]: `0`

In [15]:
```
# As we can see that if number of reviews are 0 then review per month will also be 0 so imputing 'reviews_per_month'

airbnb_data['reviews_per_month'].fillna(0, inplace=True)
```

```
In [17]: # checking the null values in complete data after imputing the values


         (100*(airbnb_data.isnull().sum())/len(airbnb_data.index))
```

```
Out[17]: id                                0.0
         name                              0.0
         host_id                           0.0
         host_name                         0.0
         neighbourhood_group               0.0
         neighbourhood                     0.0
         latitude                          0.0
         longitude                         0.0
         room_type                         0.0
         price                             0.0
         minimum_nights                    0.0
         number_of_reviews                 0.0
         last_review                       0.0
         reviews_per_month                 0.0
         calculated_host_listings_count    0.0
         availability_365                  0.0
         dtype: float64
```

Imputed all the missing values

❖ Step 2 : Univariate and Bivariate Analysis:

We will perform univariate and bivariate analysis to get the insights from data
Univariate Analysis done on columns such as host_id, neighbourhood_group, room_type, neighbourhood, price, number_of _reviewes, availability_365

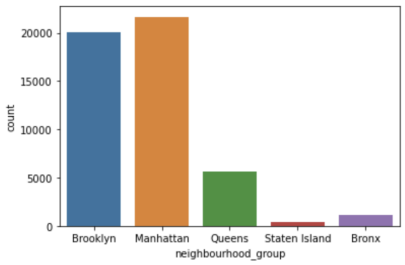
```
In [18]: airbnb_data.host_id.value_counts().iloc[:5]
```

```
Out[18]: 219517861    327
         107434423    232
         30283594     121
         137358866    103
         16098958      96
         Name: host_id, dtype: int64
```

host_id 219517861 has maximum number of bookings

```
In [25]: # checking which neighbourhood more bookings

         sns.countplot(x=airbnb_data['neighbourhood_group'])
         plt.show()
```

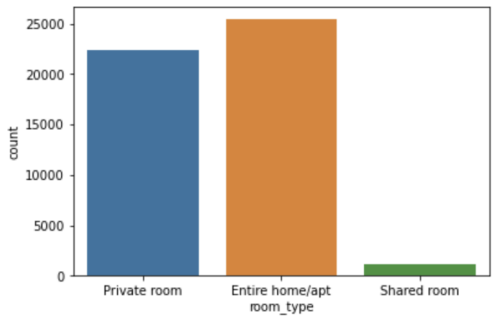


Manhattan city has more no. of transactions whereas Staten Island and Bronx have very less transactions

In [29]:
```
# checking which room type

sns.countplot(x=airbnb_data['room_type'])
plt.show()
```

Out[29]:
```
<AxesSubplot:xlabel='room_type', ylabel='count'>
```



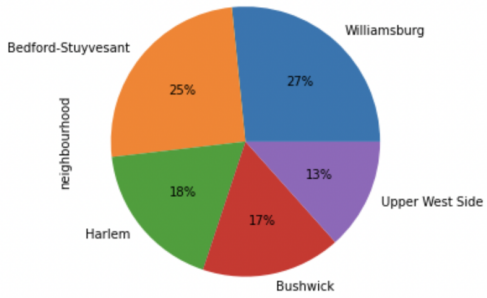
Room type 'Entire home/apt' has more number of transactions as compared to private room and shared room

Shared room has least number of transactions/bookings

In [36]:
```
# checking most preferred neighbourhood
df3= airbnb_data['neighbourhood'].value_counts().iloc[:5]
df3
```

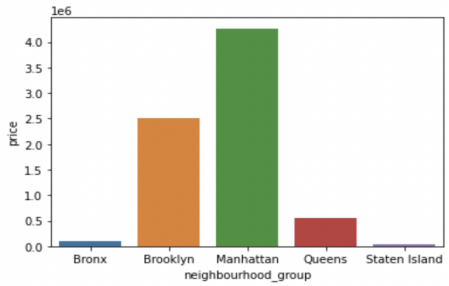
Out[36]:
```
Williamsburg          3920
Bedford-Stuyvesant    3714
Harlem                2658
Bushwick              2465
Upper West Side       1971
Name: neighbourhood, dtype: int64
```

In [48]:
```
df3.plot.pie(autopct='%1.0f%%',figsize=(10, 5))
plt.show()
```



Williamsburg neighbourhood is the most preferred neighbourhood which has maximum number of bookings

```
df4 = airbnb_data.groupby(['neighbourhood_group']).agg('sum')
df4.reset_index(inplace=True)
df4[['neighbourhood_group','price']].sort_values(by='price',ascending = False)
sns.barplot(x=df4['neighbourhood_group'],y=df4['price'])
plt.show()
```




```
df4[['neighbourhood_group','price']]
```

	neighbourhood_group	price
0	Bronx	95459
1	Brooklyn	2500600
2	Manhattan	4264527
3	Queens	563867
4	Staten Island	42825

Manhattan city area has the high prices

Bronx and Staten Island has low prices

Fetching names of Airbnb which have highest price (3 Airbnbs have maximum price of 10k$)

```
df6 = airbnb_data[airbnb_data['price'] == airbnb_data['price'].max()]
df6
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of
9151	7003697	Furnished room in Astoria apartment	20582832	Kathrine	Queens	Astoria	40.76810	-73.91651	Private room	10000	100	
17692	13894339	Luxury 1 bedroom apt. - stunning Manhattan views	5143901	Erin	Brooklyn	Greenpoint	40.73260	-73.95739	Entire home/apt	10000	5	
29238	22436899	1-BR Lincoln Center	72390391	Jelena	Manhattan	Upper West Side	40.77213	-73.98665	Entire home/apt	10000	30	

Room_type "Entire home/apt" has the highest price

```
# checking the number of days when listing is available for booking
airbnb_data['availability_365'].value_counts()
```

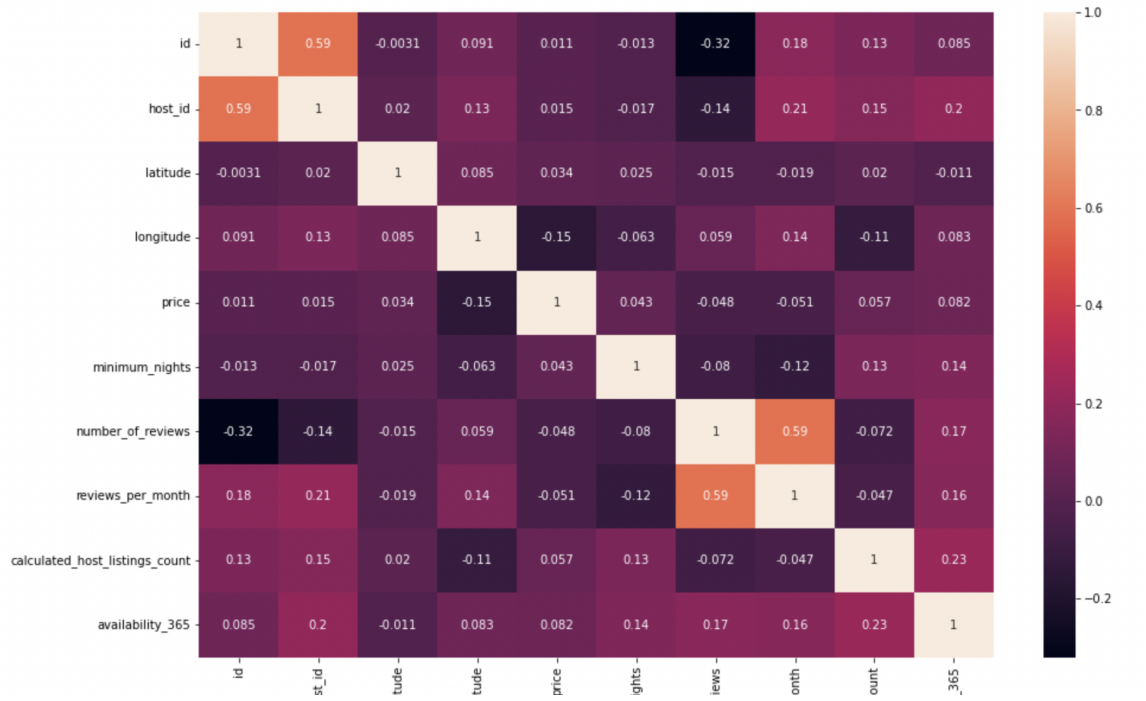
```
0      17533
365     1295
364      491
1        408
89       361
5        340
3        306
179      301
90       290
2        270
6        245
363      239
8        233
4        233
342      230
188      225
7        219
88       200
311      199
```

Around 1295 Airbnbs have min availability of 365 days

Bivariate Analysis:

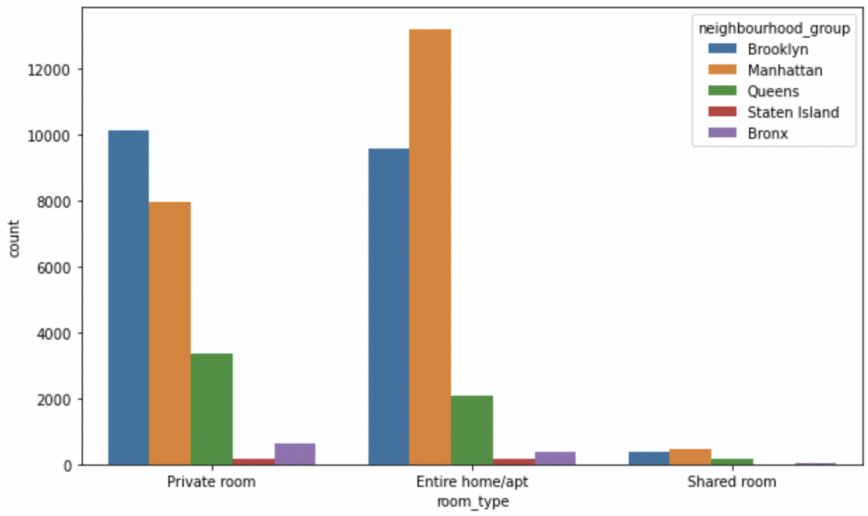
```
# plotting the heat map

plt.figure(figsize=(15,10))
sns.heatmap(airbnb_data.corr(), annot=True)
plt.show()
```




```
plt.figure(figsize=(10,5))
sns.countplot(data = airbnb_data, x = 'room_type', hue = 'neighbourhood_group')
plt.show()
```

```
<AxesSubplot:xlabel='room_type', ylabel='count'>
```



In Manhattan most number of bookings are of Entire home/apt room type and in Broolyn Private rooms more preferred

❖ Step 3 Outlier Analysis:

```
outlier_Ananlysis=airbnb_data[['price','minimum_nights','number_of_reviews','reviews_per_month','calculated_host_lis
```

```
outlier_Ananlysis
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
0.25	69.0	1.0	1.0	0.04	1.0	0.0
0.50	106.0	3.0	5.0	0.37	1.0	45.0
0.75	175.0	5.0	24.0	1.58	2.0	227.0
0.95	355.0	30.0	114.0	4.31	15.0	359.0
0.99	799.0	45.0	214.0	6.80	232.0	365.0

We can take 99percentile as cutt off and remove the Outliers.

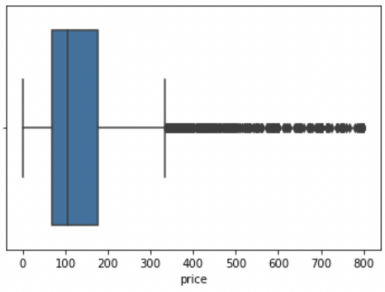
```
for i in outlier_Ananlysis.columns:
    print(outlier_Ananlysis[i][0.99])
    airbnb_data=airbnb_data[airbnb_data[i]<=outlier_Ananlysis[i][0.99]]
```

```
799.0
45.0
214.0
6.8
232.0
365.0
```

```
airbnb_data[['price','minimum_nights','number_of_reviews','reviews_per_month','calculated_host_listings_count','avai
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	46751.000000	46751.000000	46751.000000	46751.000000	46751.000000	46751.000000
mean	137.272144	5.763192	20.689269	0.997857	5.033176	109.046630
std	103.772163	8.436480	35.689302	1.347097	20.100660	130.371889
min	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
25%	69.000000	1.000000	1.000000	0.040000	1.000000	0.000000
50%	105.000000	3.000000	5.000000	0.360000	1.000000	38.000000
75%	175.000000	5.000000	23.000000	1.490000	2.000000	215.000000
max	799.000000	45.000000	214.000000	6.800000	232.000000	365.000000

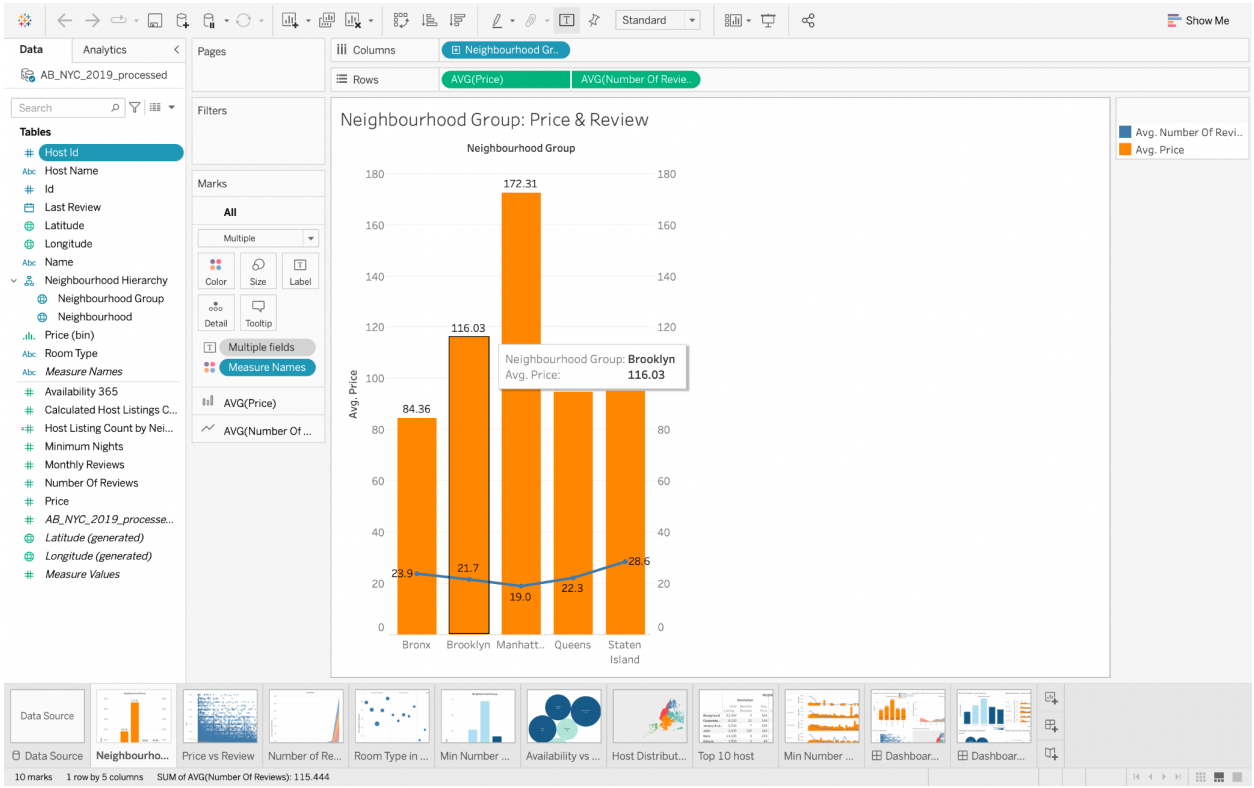
```
#Detecting Outliers through box plots for the ones which has IQR more than zero
for i in outlier_Ananlysis.columns:
    sns.boxplot(airbnb_data[i])
    plt.show()
```



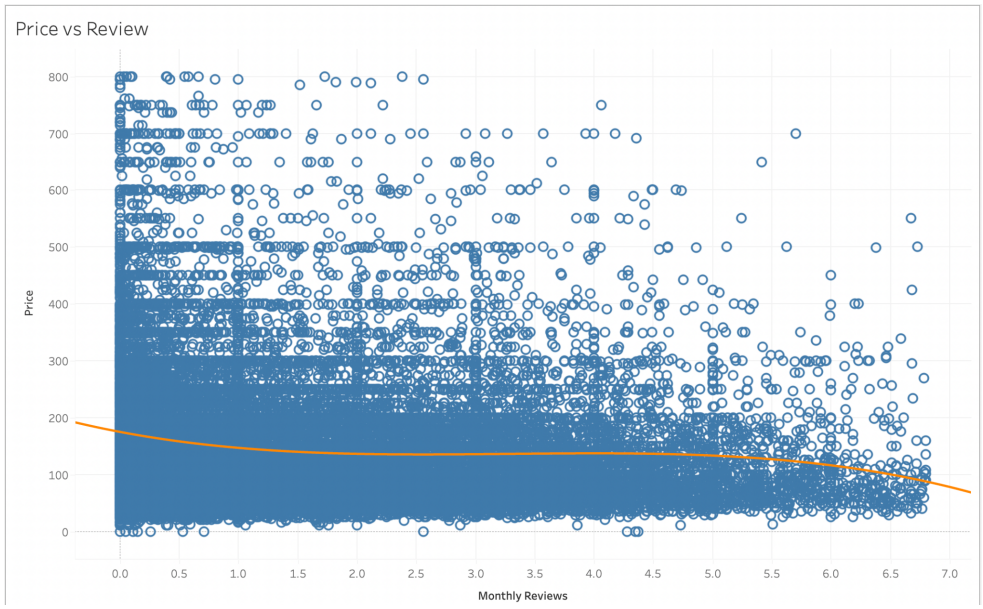
❖ Step 4 Downloading the cleaned dataset and using it for Tableau Visualizations:

```
airbnb_data.to_csv(r'C:\Swarna\Upgrad DS\Tableau\Case Study\AB_NYC_2019_processed.csv',index=False)
```

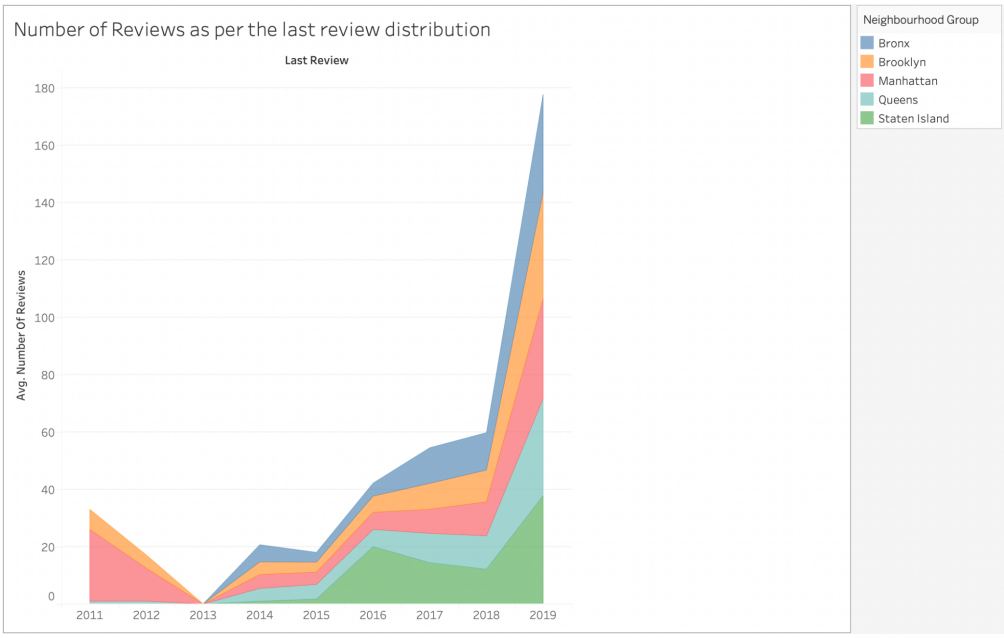
❖ Step 5 Tableau Visualizations:



- Manhattan has the maximum average price as 172.31$ and Bronx has lowest average price as 84.36$.
- Staten Island has maximum no. of reviews

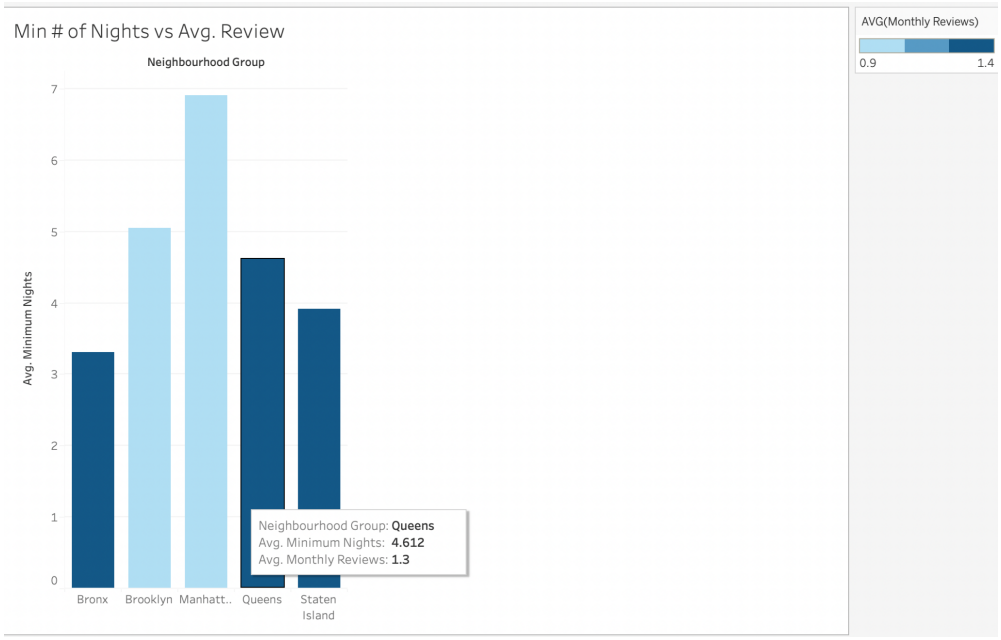


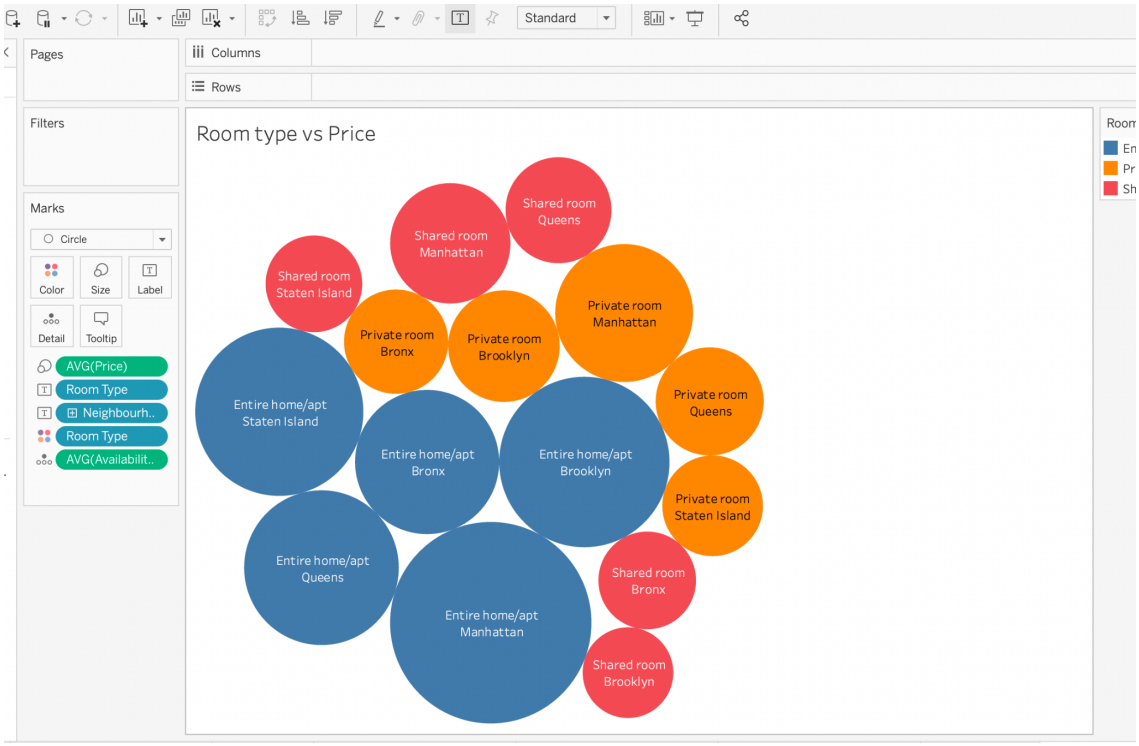
- Airbnbs having price range upto 250$ have more reviews per month



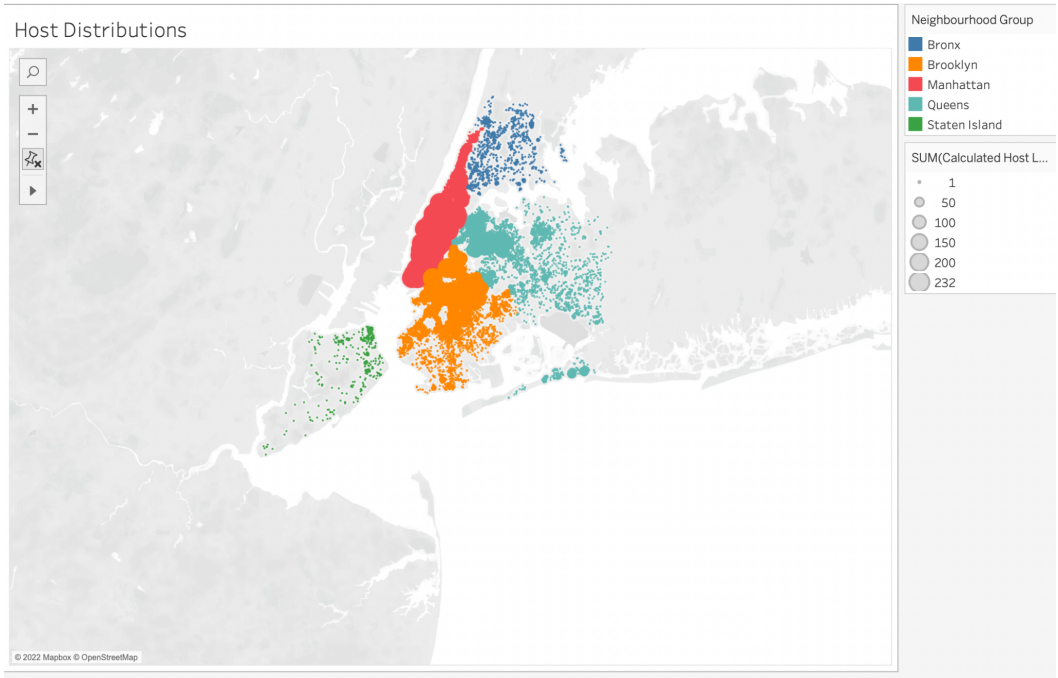
- Customers were interested in booking Airbnbs in Manhattan and Brooklyn till 2013
- Staten Island is having the last review in 2019 with highest average number of reviews 37.74 followed by Brooklyn with average number of reviews 36.85.

- For all the room_types manhattan is having the highest average price as we can see from the size of the bubble.

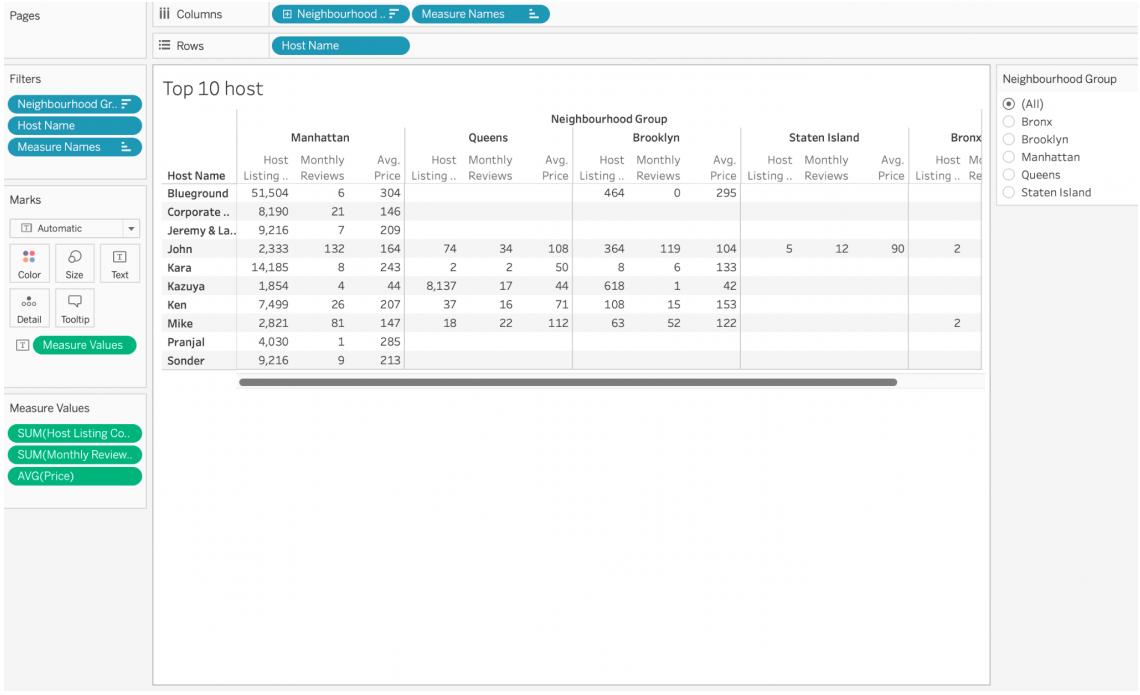




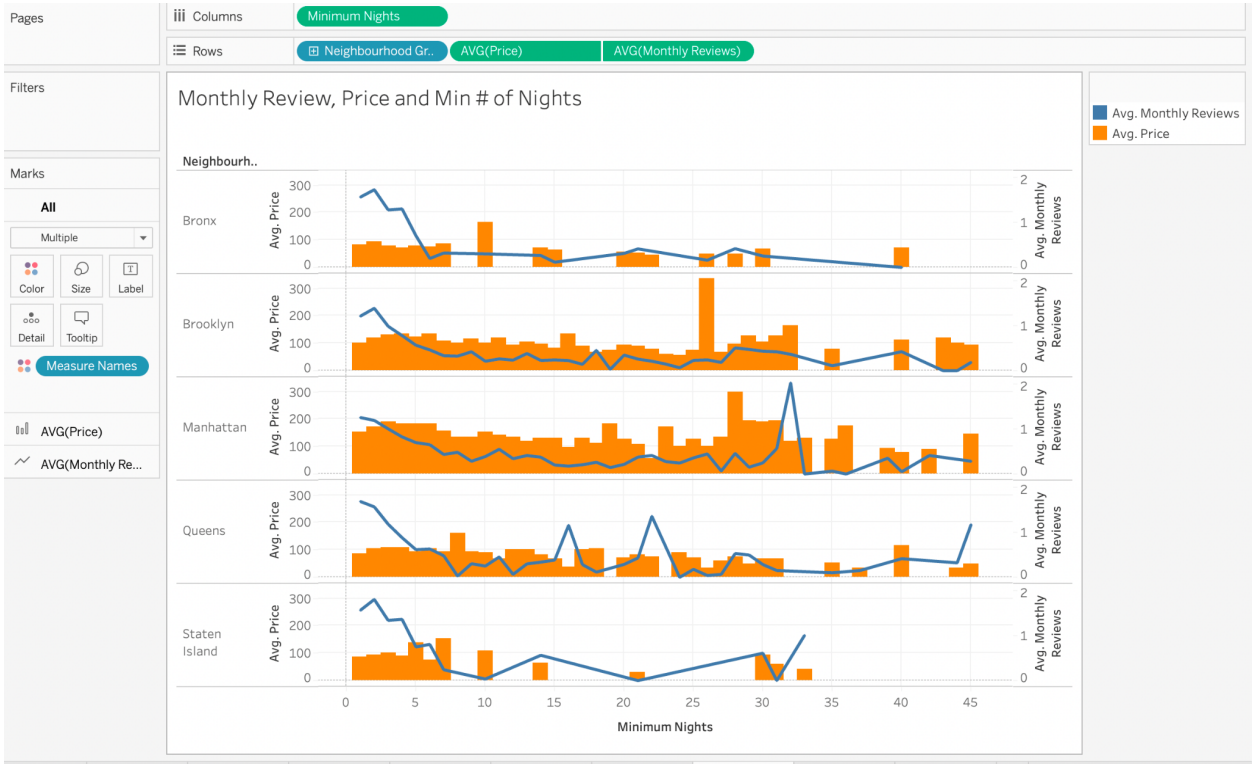
- We can see the different room types and their price distribution
- Manhattan is having highest price distribution



- In Manhattan the host concentration is more as compared to other places

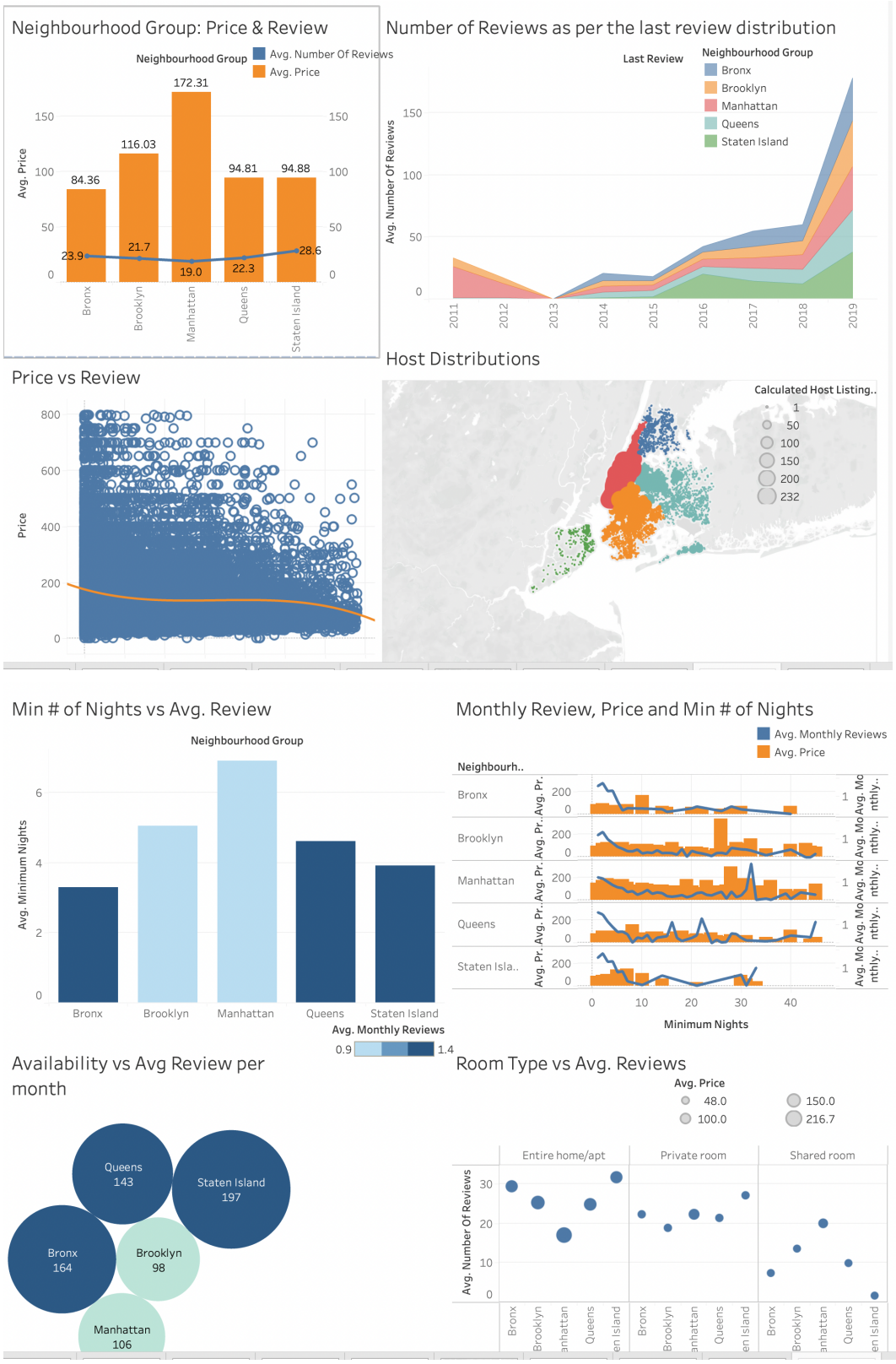


- We can see the top 10 host listing count, monthly reviews received, average price in each neighbourhood group
- Blueground host is having highest listing count and average price in Manhattan



- Manhattan and Brooklyn are the costliest place
- Customers prefer lesser number of minimum nights as they are more economical and the number of reviews are also more with lesser amount of nights

❖ Step 6 Dashboard:



❖ Summary Insights:

- Host Sonder (NYC)host_id 219517861 has maximum number of bookings
- Average availability for Airbnbs in Newyork is ~113 days and average price for an Airbnb is ~153$
- Customers are preferring Airbnbs having price range upto 250$ as they have more reviews per month
- Manhattan and Brooklyn were only 2 major places where customers were more interested till 2013 but from 2013 - 2019 Bronx, Queens and StateIsland places are being added to customers preferences. Hence company can focus on these new places as they are having low price Airbnbs also.
- Company should try to improve/focus more on customer experience in Airbnbs in Manhattan city as Manhattan city has more no. of transactions whereas Staten Island and Bronx have very less transactions
- Room type 'Entire home/apt' has more number of transactions and more preferable by customers as compared to private room and shared room
- Less customers preferring Shared rooms
- Top 5 neighbourhood from which there are more no. of transactions are Williamsburg, Bedford-Stuyvesant, Harlem, Bushwick, Upper West Side.
- Williamsburg neighbourhood is the most preferred neighbourhood which has maximum number of bookings
- Manhattan & Brooklyn has Airbnbs with high prices and Staten Island, Bronx has Airbnbs of lower price
- Airbnbs “Furnished room in Astoria apartment” in Queens, “Luxury 1 bedroom apt. -stunning Manhattan views” in Brooklyn, “1-BR Lincoln Center” in Manhattan are having the highest price as 10k$ with Brooklyn Airbnb having highest number of reviews
- Airbnb “Room near JFK Queen Bed” in Queens is having the highest number of reviews with price as 47$
- Around 1295 Airbnbs have min availability of 365 days
- In Manhattan most number of bookings are of Entire home/apt room type and in Brooklyn Private rooms more preferred
- Blueground host is having highest listing count and average price in Manhattan
- Customers prefer lesser number of minimum nights as they are more economical and the number of reviews are also more with lesser amount of nights
- In Manhattan there are more number of reviews from customers who are opting for minimum nights between 30-33 as the average cost is low

❖ Assumptions:

- Considered name in the data sheet as property name
- Imputed the missing values for “name”, “host_name” columns with “Not available” as we don’t have the name of Airbnb and host in the dataset.
- Imputed the missing values for column “last_review” with minimum date in the data set as we need to keep the data type in same format and for “reviews_per_month” we assume no reviews and updated them as 0.
- Considered up to the 99th percentile for the numerical columns such as “price”, “minimum_nights” etc and then capping the maximum value for these columns with the 99th percentile to remove the outliers.