INT-353

EDA PROJECT

G. Soma Sai Teja

Reg.no: 12017678

Roll.no: RK20CHA33

1.WHY THIS PROJECT?

EDA (Exploratory Data Analysis): It is an approach to analyze dataset using visual techniques. EDA is classified in two ways, first one is either graphical or non-graphical and second one is either univariate or multivariate. It is used to discover any patterns or to check assumptions with statistical summary and graphical representations.

Airbnb Dataset

"Airbnb's mission is to create a world where anyone can belong anywhere"

It is an American company that operates an online marketplace for lodging, primarily homestays for vacation rentals, and tourism activities. Since 2008 a lot of hosts and guests have used Airbnb for stay purpose to travel conveniently and lead a professional view of life, now it became one of the world's largest services that is recognized by everyone

I chose this dataset because, since Airbnb has become large and it needs some changes based on the customers reviews, by analysing the data Airbnb can take business decisions, guiding marketing initiatives, implementation of additional services and much more.

2.Domain Idea

The dataset I have chosen comes under online marketplace and hospitality service. This dataset was taken from Kaggle and is easy to use with no restrictions whereas some organisations doesn't provide the data, data analysis on millions of listings provided through Airbnb is a crucial factor for the company. The main idea is to improve the Airbnb System from customer reviews from different places, improving their business from visualizing the data and to take it the advance level by improving all of its services from its dataset.

3.Dataset

This dataset contains information about customer id details, hotel bookings and its reviews, locations of hotels and etc, it has 16 columns and more than 40,000 rows.

Below are the columns:

- **Id (Listing Id)**
- **Name (Name of the Listing)**
- **Host ID**
- **Host Name (Name of the Host)**
- **Neighbourhood Group (location)**
- **Neighbourhood (area)**
- **Latitude (Co-ordinates)**
- **Longitude (Co-ordinates)**
- **Room Type (Listing Space Type)**

- **Price (Price in Dollars)**
- **Minimum Nights (Number of Nights)**
- **Number Of Reviews**
- **Last Review**
- **Reviews Per Month**
- **Calculated Host Listings (Amount of Listing for Host)**
- **Availability 365 (No of Days When Listing is available for Booking)**

This Airbnb ('AB_NYC_2019') dataset for the 2019 year is a very rich dataset with a variety of columns that allowed us to do deep data exploration on each significant column presented. Overall, we will discover a very good number of interesting relationships between features and explained each step of the process. This data analytics is very much mimicked on a higher level on Airbnb Data/Machine Learning team for better business decisions, control over the platform, marketing initiatives, implementation of new features and much more. Therefore, I hope this kernel helps everyone!

Libraries imported

pandas as pd

NumPy as np

matplotlib. pyplot as plt

seaborn as sns

warnings

%Matplotlib inline

CA – 2 (Data cleaning, Univariate, Bivariate and Statistical Hypothesis)

Data Cleaning

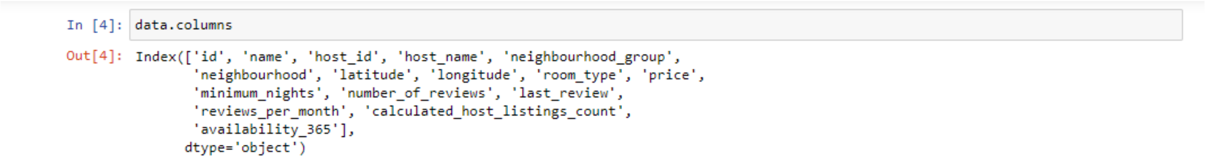
Having clean data will ultimately increase overall productivity and allow for the highest quality information in your decision-making. Benefits include: Removal of errors when multiple sources of data are at play.

When working with multiple data sources, there are many chances for data to be incorrect, duplicated, or mislabeled. If data is wrong, outcomes and algorithms are unreliable, even though they may look correct. *Data cleaning* is the process of changing or eliminating garbage, incorrect, duplicate, corrupted, or incomplete data in a dataset.

- Remove duplicates.
- Remove irrelevant data.
- Renaming the columns
- Convert data type.
- Clear formatting.
- Fix errors.

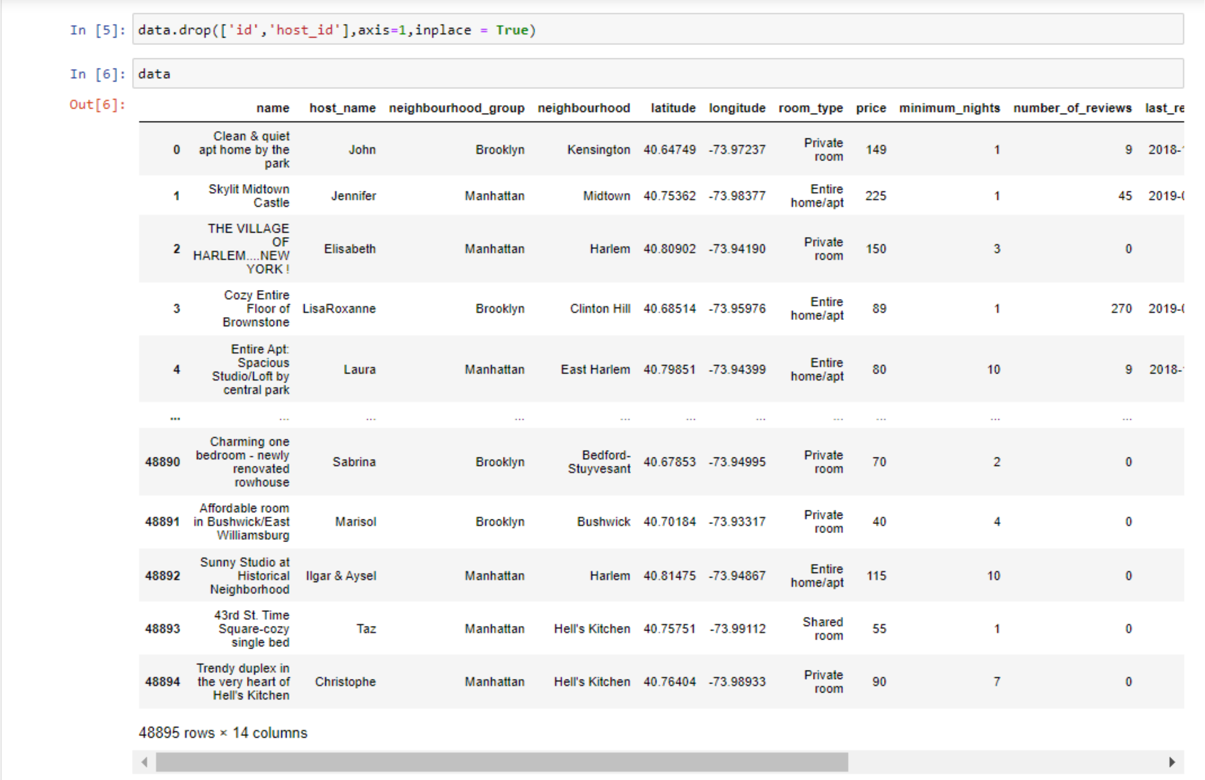
- Handling Outliers
- Handle missing values.
- Skipping unnecessary rows

Checking for Unnecessary columns.



```
In [4]: data.columns

Out[4]: Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
               'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
               'minimum_nights', 'number_of_reviews', 'last_review',
               'reviews_per_month', 'calculated_host_listings_count',
               'availability_365'],
              dtype='object')
```

Dropping the id and host id as they are not useful for my analysis. And rechecked the data by calling it again

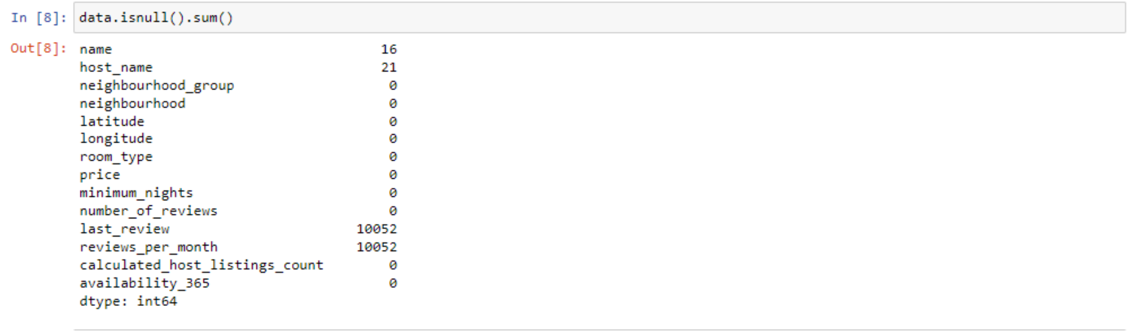


```
In [5]: data.drop(['id','host_id'],axis=1,inplace = True)

In [6]: data

Out[6]:
```

	name	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_re
0	Clean & quiet apt home by the park	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	2018-
1	Skylit Midtown Castle	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	2019-
2	THE VILLAGE OF HARLEM....NEW YORK !	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	
3	Cozy Entire Floor of Brownstone	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	2019-
4	Entire Apt: Spacious Studio/Loft by central park	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	2018-
...	...	...	...	...	...	...	...	...	...	...	
48890	Charming one bedroom - newly renovated rowhouse	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	0	
48891	Affordable room in Bushwick/East Williamsburg	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	0	
48892	Sunny Studio at Historical Neighborhood	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	0	
48893	43rd St. Time Square-cozy single bed	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	0	
48894	Trendy duplex in the very heart of Hell's Kitchen	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90	7	0	

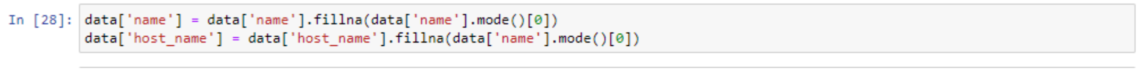
48895 rows × 14 columns

Finding null values



```
In [8]: data.isnull().sum()

Out[8]: name                                 16
        host_name                            21
        neighbourhood_group                   0
        neighbourhood                         0
        latitude                              0
        longitude                             0
        room_type                             0
        price                                 0
        minimum_nights                        0
        number_of_reviews                     0
        last_review                       10052
        reviews_per_month                 10052
        calculated_host_listings_count        0
        availability_365                      0
        dtype: int64
```

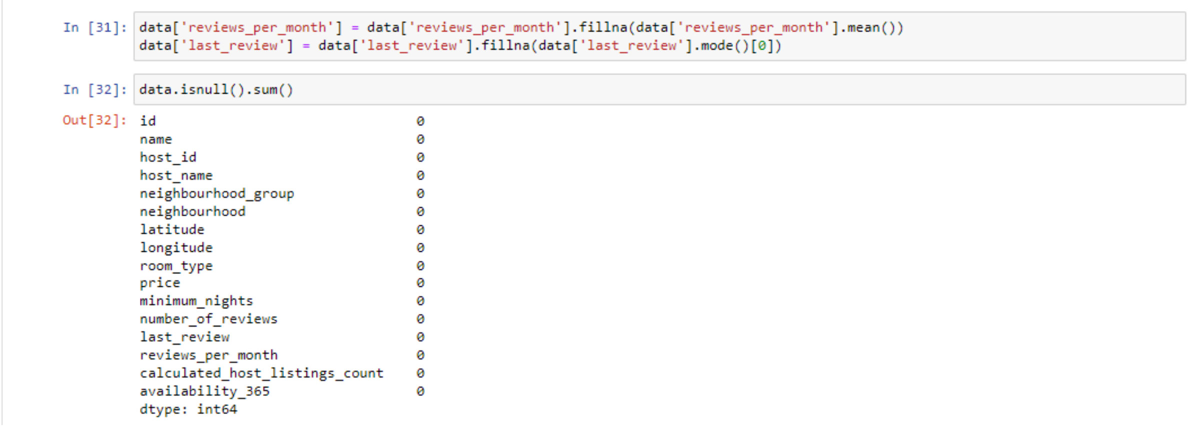
We can see there are null values in my data. Lets fill these null values with some values.



```
In [28]: data['name'] = data['name'].fillna(data['name'].mode()[0])
         data['host_name'] = data['host_name'].fillna(data['name'].mode()[0])
```

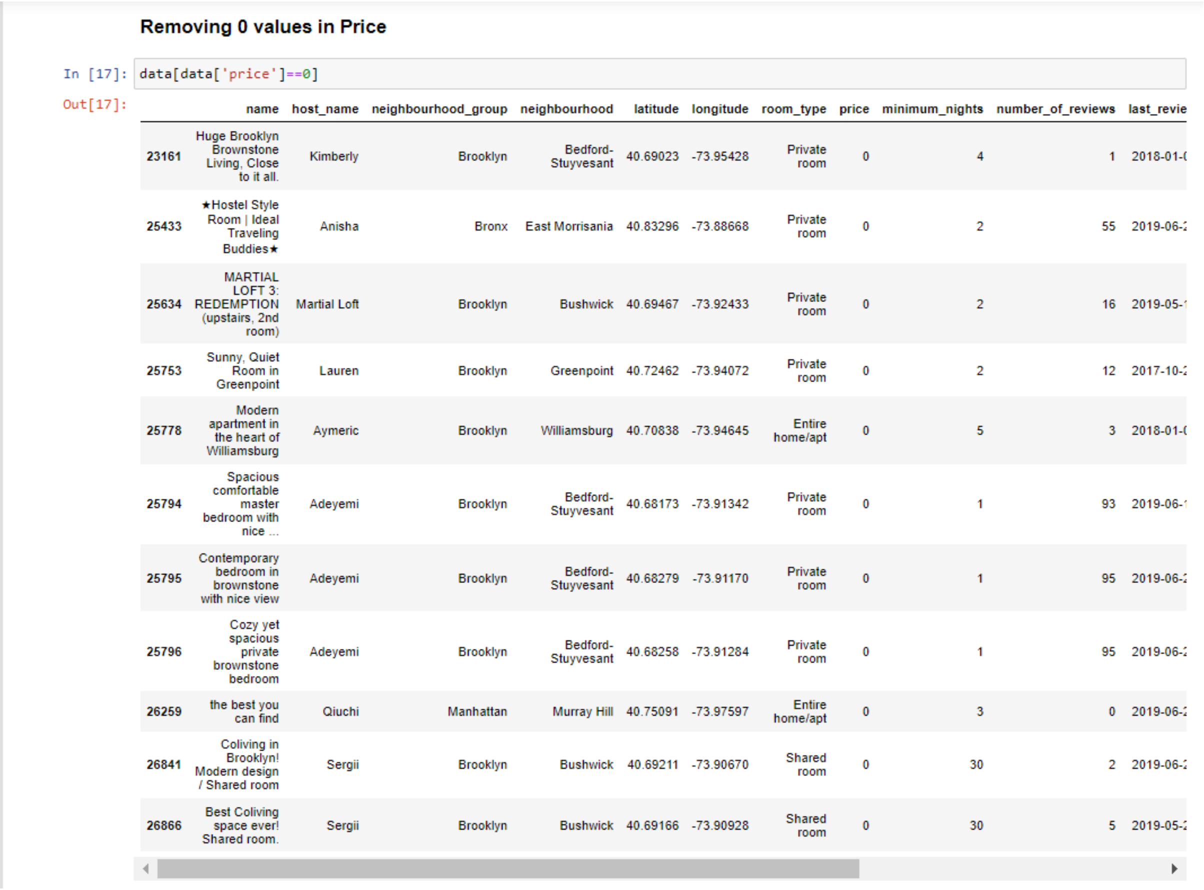
Here I filled the null values in name and host name with mode, which will fill it with most repeated value.

Let’s fill another null value columns



```
In [31]: data['reviews_per_month'] = data['reviews_per_month'].fillna(data['reviews_per_month'].mean())
         data['last_review'] = data['last_review'].fillna(data['last_review'].mode()[0])

In [32]: data.isnull().sum()

Out[32]: id                                0
         name                              0
         host_id                           0
         host_name                         0
         neighbourhood_group               0
         neighbourhood                     0
         latitude                          0
         longitude                         0
         room_type                         0
         price                             0
         minimum_nights                    0
         number_of_reviews                 0
         last_review                       0
         reviews_per_month                 0
         calculated_host_listings_count    0
         availability_365                  0
         dtype: int64
```

And I filled the remaining null values with proper functions like mean and mode and all the null values are filled now.



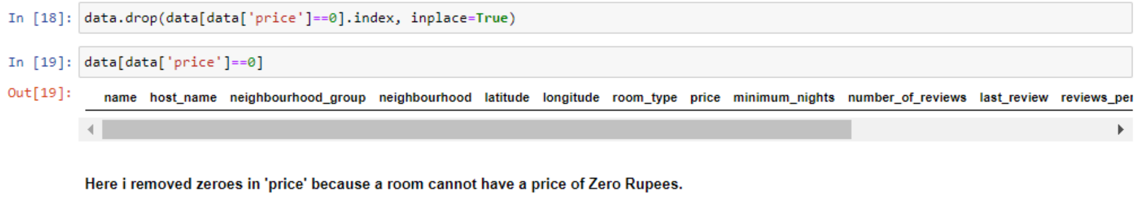
Removing 0 values in Price

```
In [17]: data[data['price']==0]
```

Out[17]:

	name	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_revie
23161	Huge Brooklyn Brownstone Living, Close to it all.	Kimberly	Brooklyn	Bedford-Stuyvesant	40.69023	-73.95428	Private room	0	4	1	2018-01-0
25433	★Hostel Style Room | Ideal Traveling Buddies★	Anisha	Bronx	East Morrisania	40.83296	-73.88668	Private room	0	2	55	2019-06-2
25634	MARTIAL LOFT 3: REDEMPTION (upstairs, 2nd room)	Martial Loft	Brooklyn	Bushwick	40.69467	-73.92433	Private room	0	2	16	2019-05-1
25753	Sunny, Quiet Room in Greenpoint	Lauren	Brooklyn	Greenpoint	40.72462	-73.94072	Private room	0	2	12	2017-10-2
25778	Modern apartment in the heart of Williamsburg	Aymeric	Brooklyn	Williamsburg	40.70838	-73.94645	Entire home/apt	0	5	3	2018-01-0
25794	Spacious comfortable master bedroom with nice ...	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68173	-73.91342	Private room	0	1	93	2019-06-1
25795	Contemporary bedroom in brownstone with nice view	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68279	-73.91170	Private room	0	1	95	2019-06-2
25796	Cozy yet spacious private brownstone bedroom	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68258	-73.91284	Private room	0	1	95	2019-06-2
26259	the best you can find	Qiuchi	Manhattan	Murray Hill	40.75091	-73.97597	Entire home/apt	0	3	0	2019-06-2
26841	Coliving in Brooklyn! Modern design / Shared room	Sergii	Brooklyn	Bushwick	40.69211	-73.90670	Shared room	0	30	2	2019-06-2
26866	Best Coliving space ever! Shared room.	Sergii	Brooklyn	Bushwick	40.69166	-73.90928	Shared room	0	30	5	2019-05-2

There are some 0 values in my price column, well there cant be zeroes in price, lets drop them.



```
In [18]: data.drop(data[data['price']==0].index, inplace=True)

In [19]: data[data['price']==0]
```

Out[19]:

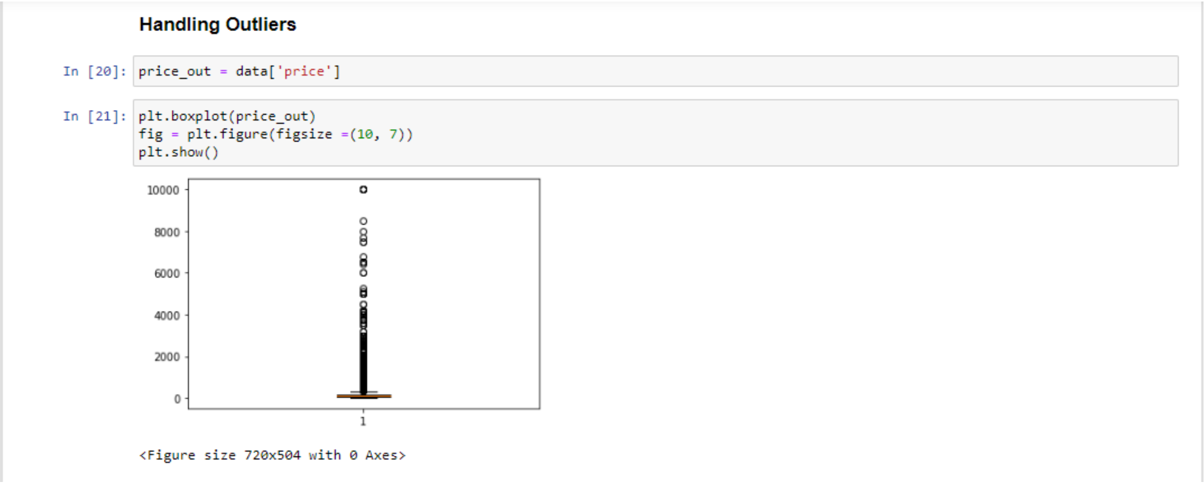
name	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_pei

Here i removed zeroes in 'price' because a room cannot have a price of Zero Rupees.

And zero values in price are dropped.

<u>Let’s check for Outliers</u>

Price has outliers



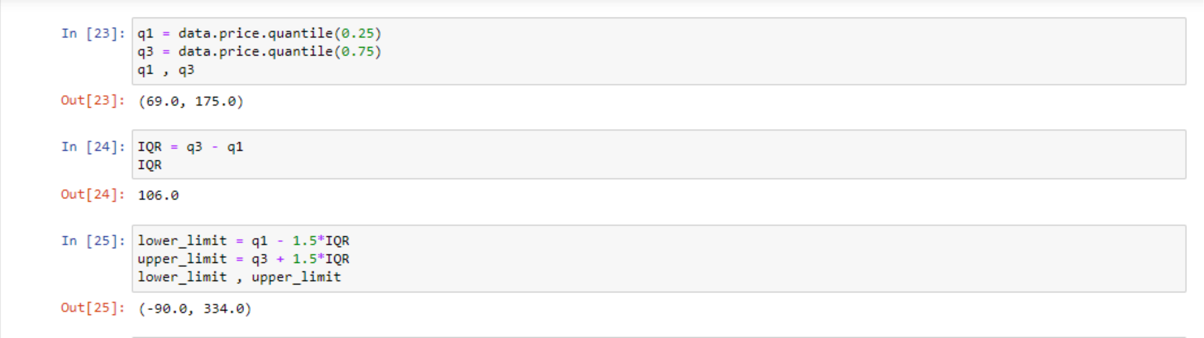
Handling Outliers

```
In [20]: price_out = data['price']
```

```
In [21]: plt.boxplot(price_out)
         fig = plt.figure(figsize =(10, 7))
         plt.show()
```

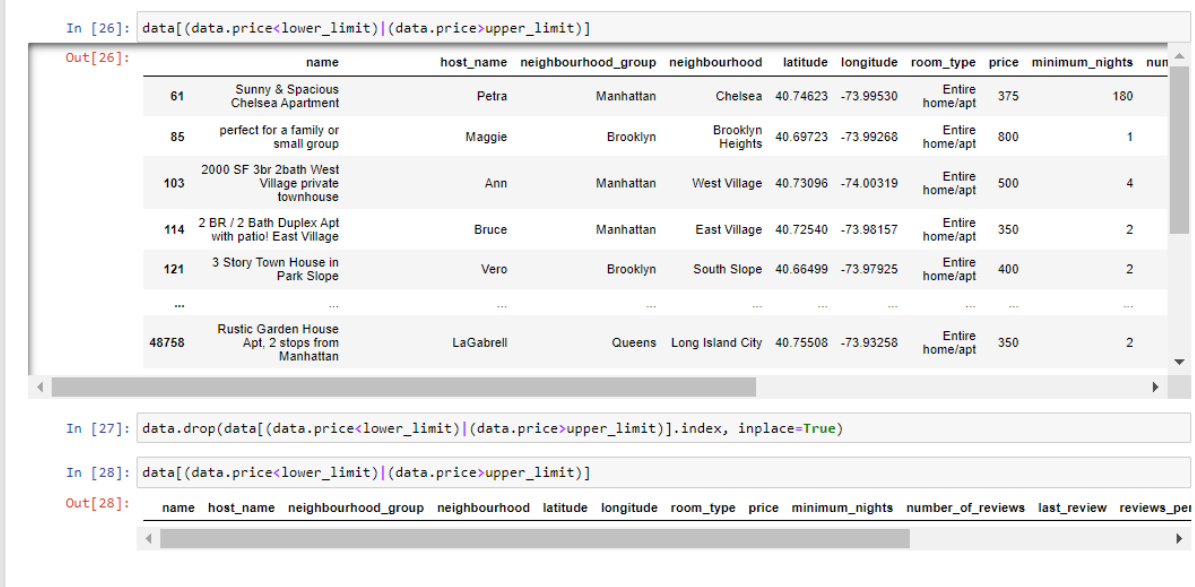
```
<Figure size 720x504 with 0 Axes>
```

There are some outliers in the price, we should clear them as it will disturb our values when we are doing statistical analysis. By using Inter Quartile range I will remove them.



```
In [23]: q1 = data.price.quantile(0.25)
         q3 = data.price.quantile(0.75)
         q1 , q3

Out[23]: (69.0, 175.0)

In [24]: IQR = q3 - q1
         IQR

Out[24]: 106.0

In [25]: lower_limit = q1 - 1.5*IQR
         upper_limit = q3 + 1.5*IQR
         lower_limit , upper_limit

Out[25]: (-90.0, 334.0)
```

Calculated the lower and upper range, Now lets drop them.



```
In [26]: data[(data.price<lower_limit)|(data.price>upper_limit)]
```

Out[26]:

	name	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	nun
61	Sunny & Spacious Chelsea Apartment	Petra	Manhattan	Chelsea	40.74623	-73.99530	Entire home/apt	375	180	
85	perfect for a family or small group	Maggie	Brooklyn	Brooklyn Heights	40.69723	-73.99268	Entire home/apt	800	1	
103	2000 SF 3br 2bath West Village private townhouse	Ann	Manhattan	West Village	40.73096	-74.00319	Entire home/apt	500	4	
114	2 BR / 2 Bath Duplex Apt with patio! East Village	Bruce	Manhattan	East Village	40.72540	-73.98157	Entire home/apt	350	2	
121	3 Story Town House in Park Slope	Vero	Brooklyn	South Slope	40.66499	-73.97925	Entire home/apt	400	2	
...	...	...	...	...	...	...	...	...	...	
48758	Rustic Garden House Apt, 2 stops from Manhattan	LaGabrell	Queens	Long Island City	40.75508	-73.93258	Entire home/apt	350	2	

```
In [27]: data.drop(data[(data.price<lower_limit)|(data.price>upper_limit)].index, inplace=True)
```

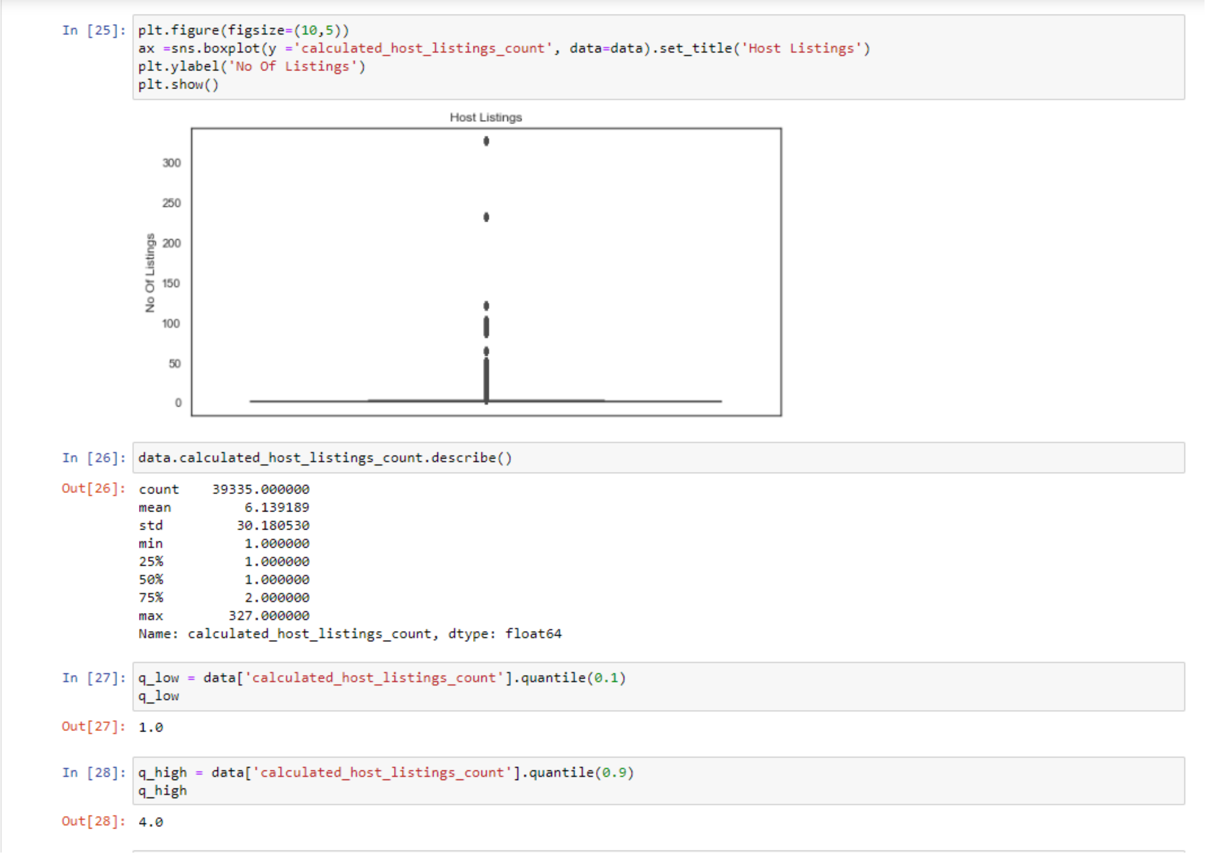
```
In [28]: data[(data.price<lower_limit)|(data.price>upper_limit)]
```

Out[28]:

name	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per

And the outliers were removed from the price column in data.

There are outliers in host listing count too .So, lets remove them.

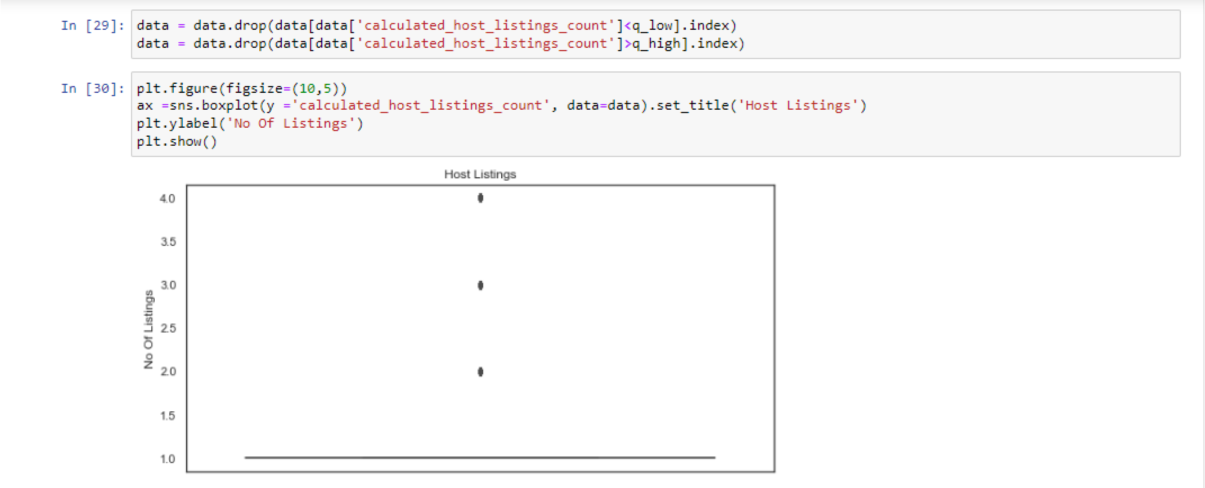


```
In [25]: plt.figure(figsize=(10,5))
         ax =sns.boxplot(y ='calculated_host_listings_count', data=data).set_title('Host Listings')
         plt.ylabel('No Of Listings')
         plt.show()
```

```
In [26]: data.calculated_host_listings_count.describe()

Out[26]: count    39335.000000
         mean         6.139189
         std         30.180530
         min          1.000000
         25%          1.000000
         50%          1.000000
         75%          2.000000
         max        327.000000
         Name: calculated_host_listings_count, dtype: float64
```

```
In [27]: q_low = data['calculated_host_listings_count'].quantile(0.1)
         q_low

Out[27]: 1.0
```

```
In [28]: q_high = data['calculated_host_listings_count'].quantile(0.9)
         q_high

Out[28]: 4.0
```

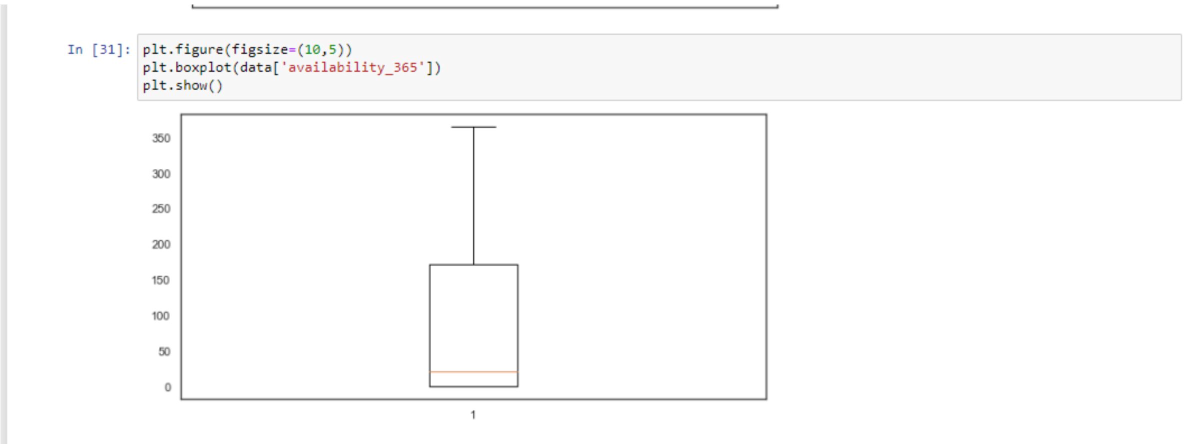


```
In [29]: data = data.drop(data[data['calculated_host_listings_count']<q_low].index)
         data = data.drop(data[data['calculated_host_listings_count']>q_high].index)
```

```
In [30]: plt.figure(figsize=(10,5))
         ax =sns.boxplot(y ='calculated_host_listings_count', data=data).set_title('Host Listings')
         plt.ylabel('No Of Listings')
         plt.show()
```

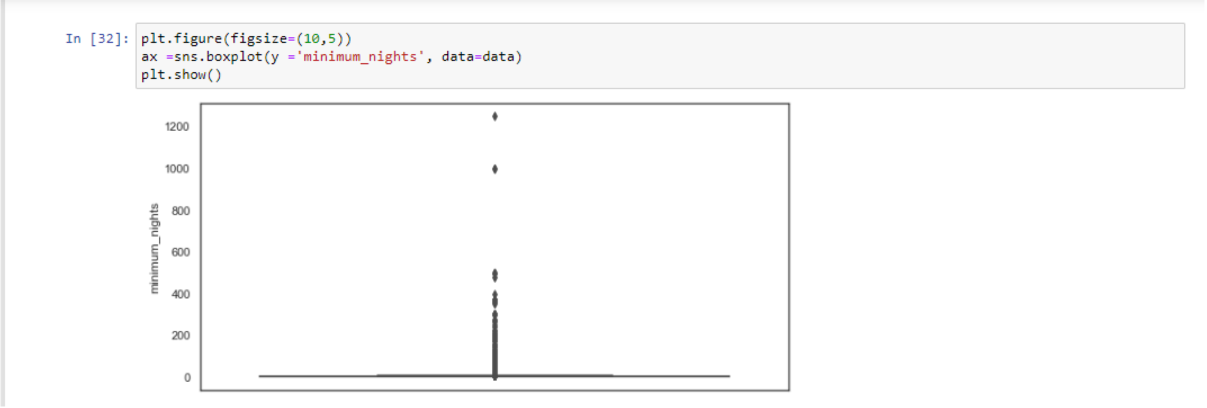
And the outliers were removed.

Let’s check for outliers in other columns too.



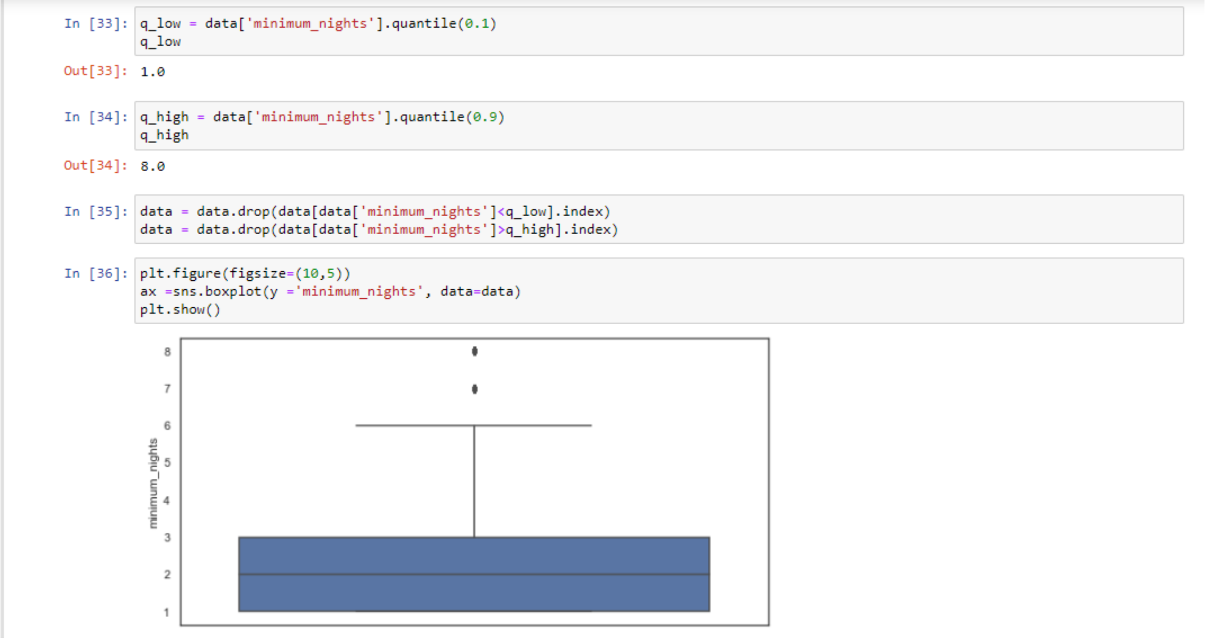
```
In [31]: plt.figure(figsize=(10,5))
         plt.boxplot(data['availability_365'])
         plt.show()
```

Seems like there are no outliers in availability days.



```
In [32]: plt.figure(figsize=(10,5))
         ax =sns.boxplot(y ='minimum_nights', data=data)
         plt.show()
```

There are outliers in minimum nights lets remove them.



```
In [33]: q_low = data['minimum_nights'].quantile(0.1)
         q_low

Out[33]: 1.0

In [34]: q_high = data['minimum_nights'].quantile(0.9)
         q_high

Out[34]: 8.0

In [35]: data = data.drop(data[data['minimum_nights']<q_low].index)
         data = data.drop(data[data['minimum_nights']>q_high].index)

In [36]: plt.figure(figsize=(10,5))
         ax =sns.boxplot(y ='minimum_nights', data=data)
         plt.show()
```

And removed.

Univariate Analysis

It is for analysing or visualizing the data of a single variable (as ‘Uni’ means single).

It mainly consists of following three types.

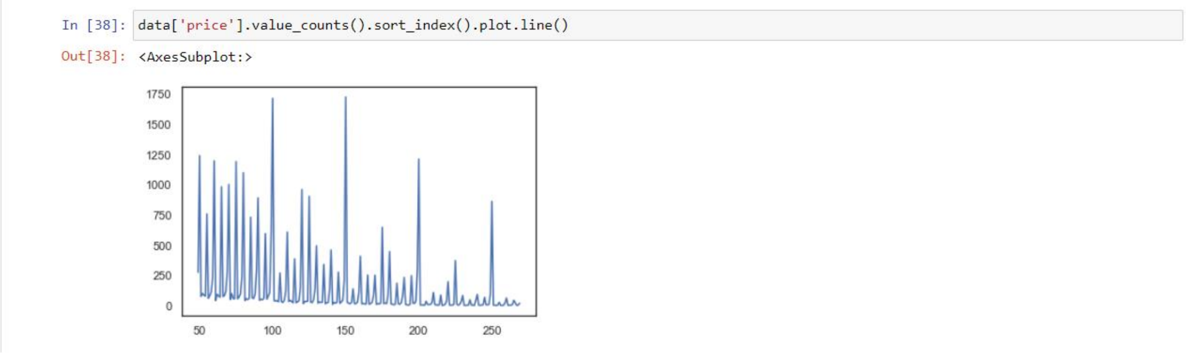
- Categorical Unordered Univariate analysis
- Categorical Ordered Univariate analysis
- Statistics on Numerical Features

There are several options for describing data with univariate data.

• Frequency Distribution Tables. • Bar Charts.

• Histograms. • Pie charts.

Line plot – It can be defined as a graph that displays data as points, showing the frequency of each value. Only for numerical data.



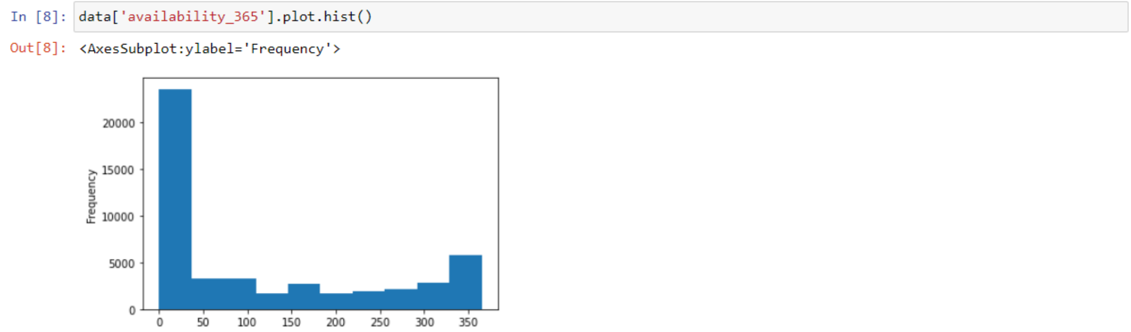
```
In [38]: data['price'].value_counts().sort_index().plot.line()
Out[38]: <AxesSubplot:>
```

Observation

Here the count is more between 0 to 150 which says that most of the price lies here for the hotels.

Histogram – Hist also shows the frequency of numerical data but in rectangular form

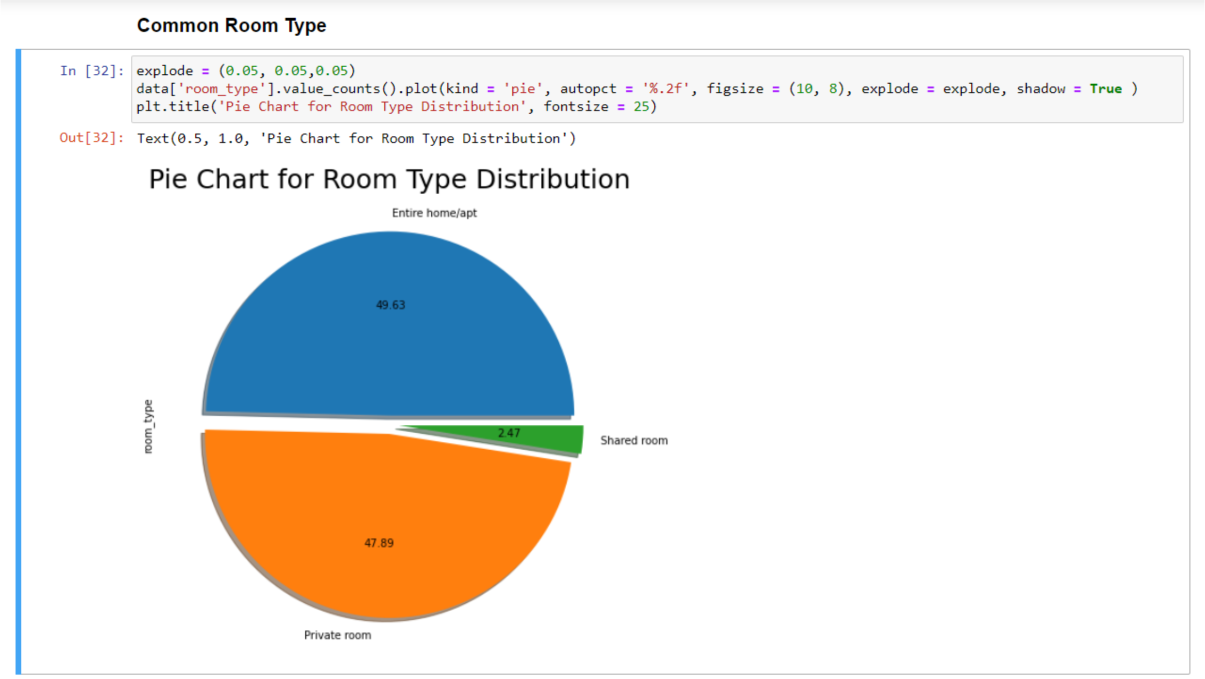
I did hist plot on the hotel availability column of 365 days and the most available days lies between 0 and 50 , where the density is more.



```
In [8]: data['availability_365'].plot.hist()
Out[8]: <AxesSubplot:ylabel='Frequency'>
```

Pie Chart - Pie charts are used in data handling and are circular charts divided up into segments which each represent a value. Pie charts are divided into sections (or 'slices') to represent values of different sizes.

Pie charts can be done by categorical data, here I did a pie chart on the most common room type preferred from three types of room(Private room, Entire Home/apt or Shared room)



Common Room Type

```
In [32]: explode = (0.05, 0.05,0.05)
         data['room_type'].value_counts().plot(kind = 'pie', autopct = '%.2f', figsize = (10, 8), explode = explode, shadow = True )
         plt.title('Pie Chart for Room Type Distribution', fontsize = 25)

Out[32]: Text(0.5, 1.0, 'Pie Chart for Room Type Distribution')
```

Observation

From the above pie chart , we can see clearly that most of the people are preferring Entire home/apt and their next choice is Private room and the least preferred room is shared room, maybe its because of price or some other reason we will see that in future analysis why they are preferring those rooms more.

Bivariate Analysis

It is for analysing or visualizing the data of a two variable (as ‘bi’ means double). To find the relation between two columns or variables.

The kind of bivariate analysis is dependent on the kind of attributes and variables that is used to analyse the data. The variables may be ordinal, categorical, or numeric. The independent variable is categorical like a brand of a pen. If the dependent and the independent variables are both ordinal which means that they have a ranking or position, then the rank correlation coefficient is measured.

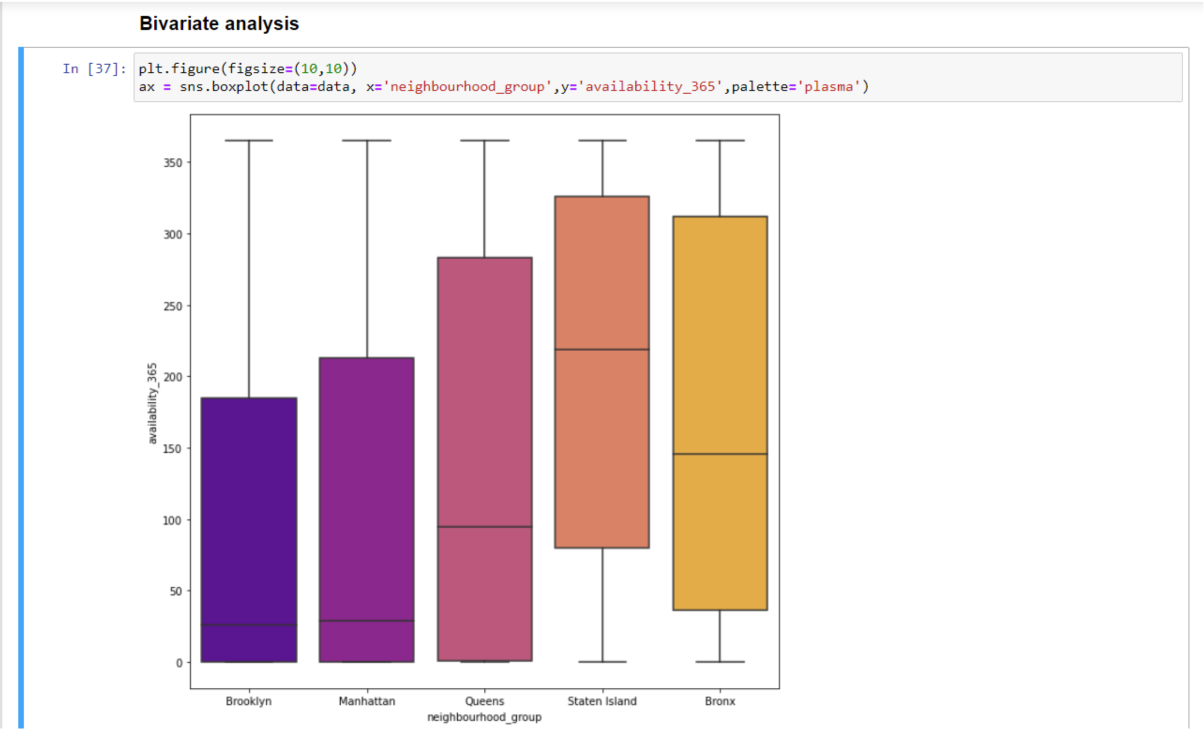
The type of data analysis done by bivariate for both numerical and categorical.

•Numerical and Numerical: In this kind of variable both the variables of the bivariate data which includes the dependent and the independent variable have a numerical value.

•Categorical and Categorical: When both the variables in the bivariate data are in static form then the data is interpreted, and statements and predictions are made from it.

•Numerical and Categorical: This is when one of the variables is numerical and the other is categorical.

Boxplot - A Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.



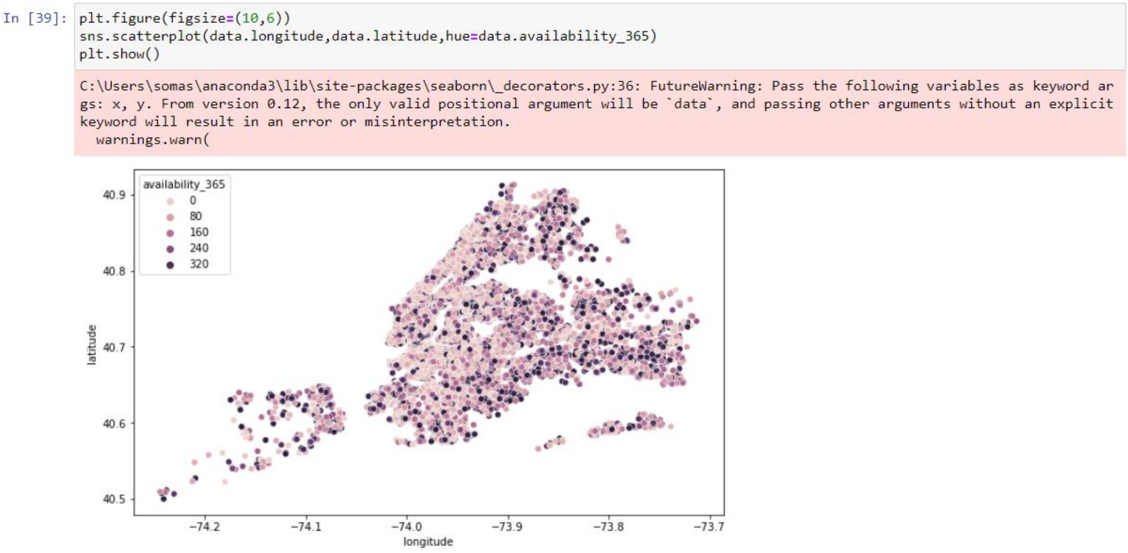
Bivariate analysis

```
In [37]: plt.figure(figsize=(10,10))
         ax = sns.boxplot(data=data, x='neighbourhood_group',y='availability_365',palette='plasma')
```

Observation

We did a boxplot using two columns one is neighbourhood_group and other is availability_365. From the above plot we can observe that Staten island and Bronx locations have more availability days average greater than 100 And in remaining places it has less availability rooms as it has less average.

Scatter Plot - A scatter plot is a diagram where each value in the data set is represented by a dot. Scatter plots are not a good option for categorical or nominal data, since these data are measured on a scale with specific values.



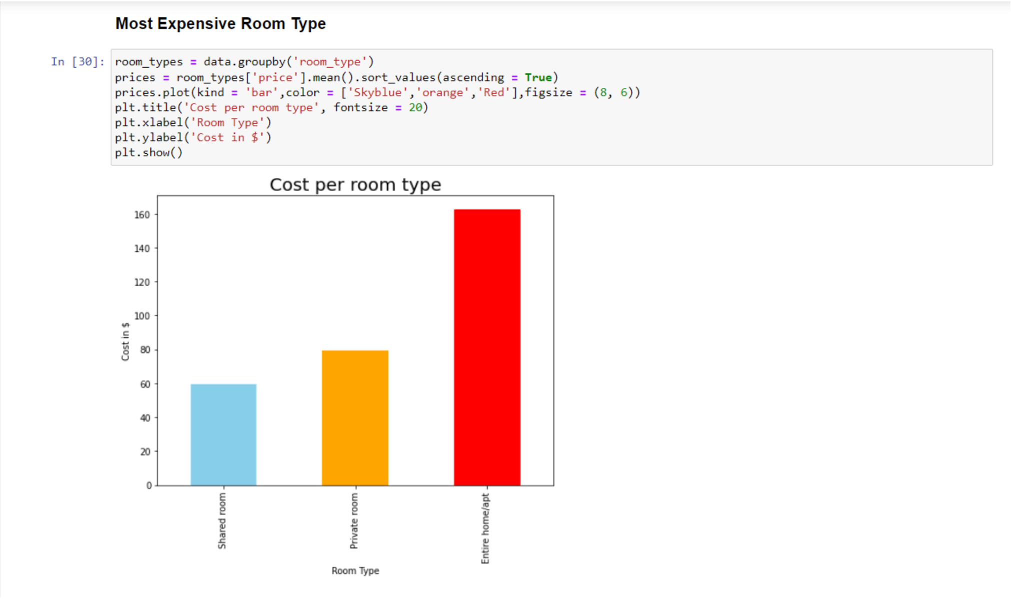
```
In [39]: plt.figure(figsize=(10,6))
         sns.scatterplot(data.longitude,data.latitude,hue=data.availability_365)
         plt.show()
```

```
C:\Users\somas\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Observation

The darker the colour in the above the more availability of rooms in 365 days, the lighter the colour the less availability of room in 365 days. Above plot is done with latitude and longitudes of New York dataset given and with availability_365 column.

Bar Chart - A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. Bar is for vertical and Barh is for horizontal plotting



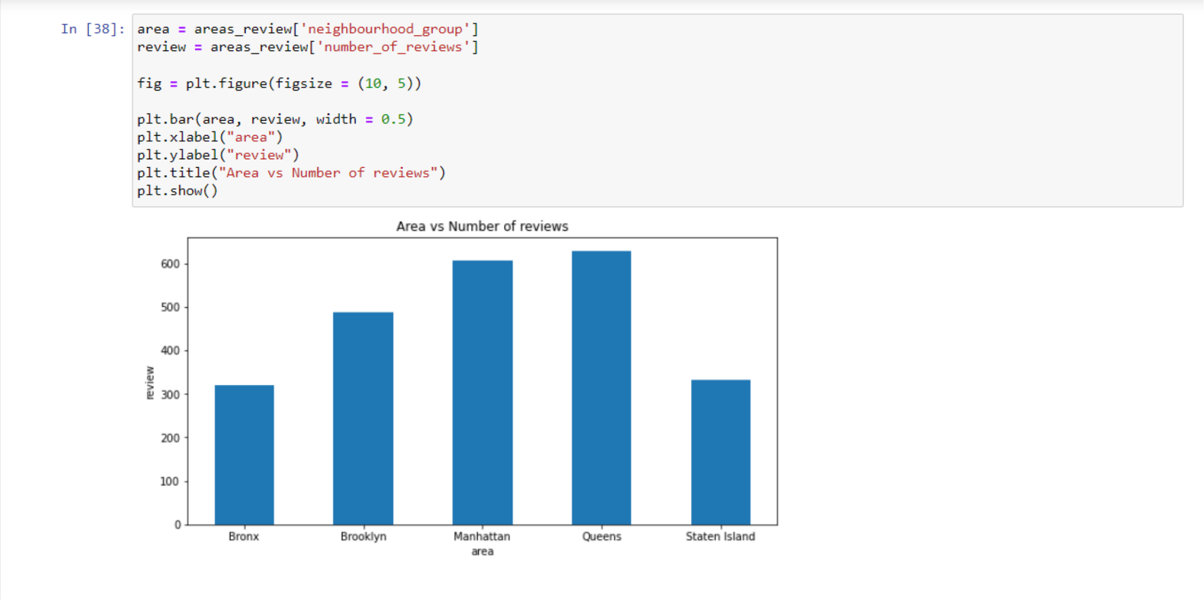
Most Expensive Room Type

In [30]:
```
room_types = data.groupby('room_type')
prices = room_types['price'].mean().sort_values(ascending = True)
prices.plot(kind = 'bar',color = ['Skyblue','orange','Red'],figsize = (8, 6))
plt.title('Cost per room type', fontsize = 20)
plt.xlabel('Room Type')
plt.ylabel('Cost in $')
plt.show()
```

Observation

Here I did a bar between room_type column and price column to check which room is costlier by using the mean. So, from that we can see Entire home/apt room type is costlier with the price of 160$ than the rest and then comes the private room after that comes the shared room these rooms lie between 60 -80$.

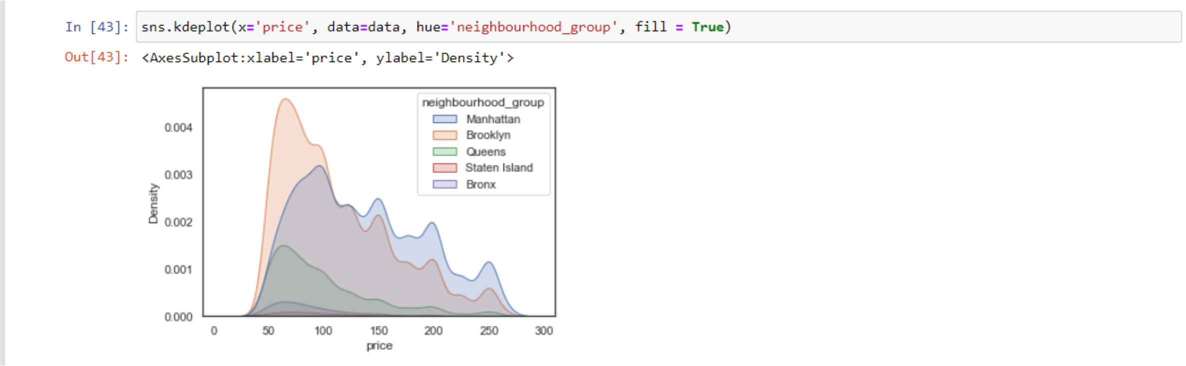
Here I did a bar chart again on another two different columns one is neighbourhood group and other is number of reviews to check the most reviewed area.



```
In [38]: area = areas_review['neighbourhood_group']
         review = areas_review['number_of_reviews']

         fig = plt.figure(figsize = (10, 5))

         plt.bar(area, review, width = 0.5)
         plt.xlabel("area")
         plt.ylabel("review")
         plt.title("Area vs Number of reviews")
         plt.show()
```

Queens area has more reviews than other with Manhattan in second and Brooklyn in third, By this we can say most tourists are visiting these areas in New York.

Kde Plot – Kde plot means Kernel Distribution Estimation which depicts the probability density function of the continuous or non-parametric data variables i.e. we can plot for the univariate, bivariate or multiple variables altogether. Here I used it for two column which is bivariate.

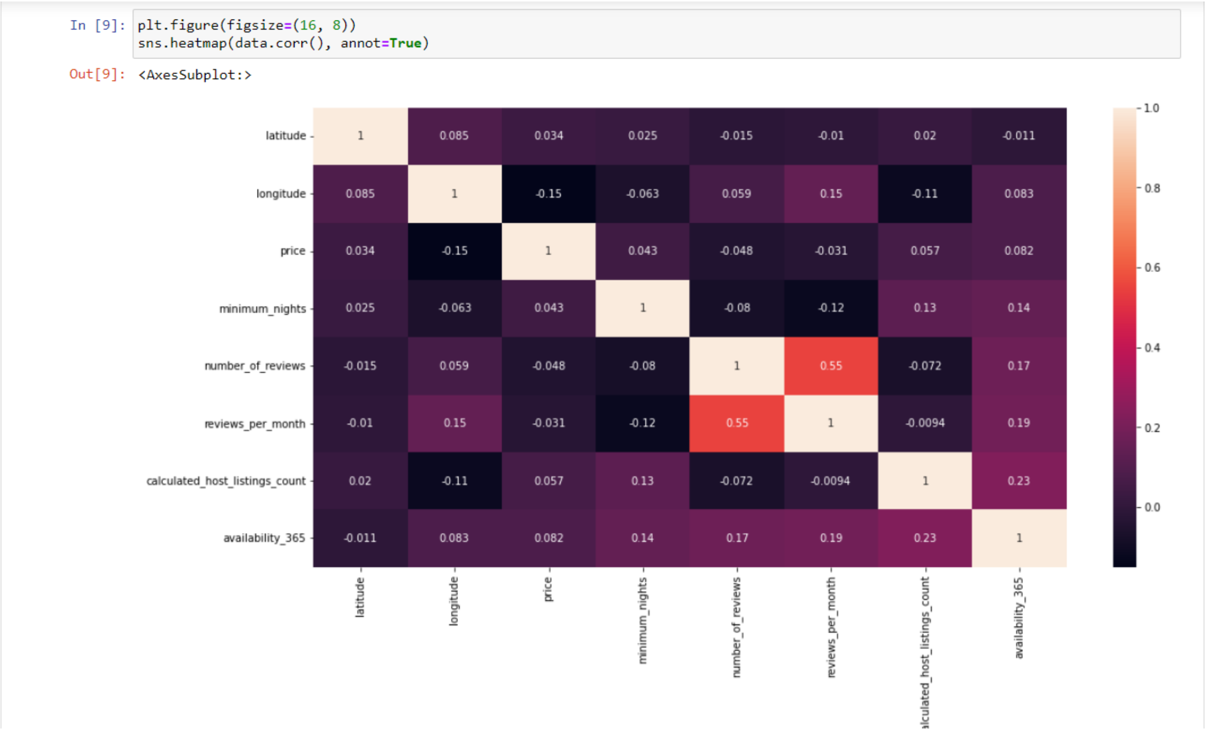


```
In [43]: sns.kdeplot(x='price', data=data, hue='neighbourhood_group', fill = True)
Out[43]: <AxesSubplot:xlabel='price', ylabel='Density'>
```

Observation

In this plot the neighbourhood group is represented by different colours to get the value, Brooklyn has highest density and Staten island has the lowest density.

The main question that the bivariate analysis answers are if there is correlation between the two variables, if the relationship is negative or positive and what is the degree or strength of the correlation.

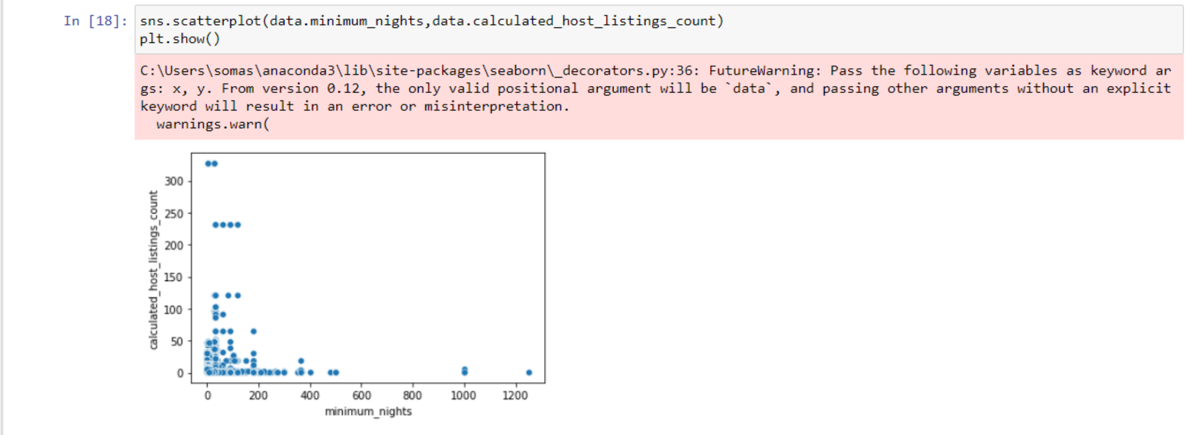


```
In [9]: plt.figure(figsize=(16, 8))
        sns.heatmap(data.corr(), annot=True)

Out[9]: <AxesSubplot:>
```

	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	alculated_host_listings_count	availability_365
latitude	1	0.085	0.034	0.025	-0.015	-0.01	0.02	-0.011
longitude	0.085	1	-0.15	-0.063	0.059	0.15	-0.11	0.083
price	0.034	-0.15	1	0.043	-0.048	-0.031	0.057	0.082
minimum_nights	0.025	-0.063	0.043	1	-0.08	-0.12	0.13	0.14
number_of_reviews	-0.015	0.059	-0.048	-0.08	1	0.55	-0.072	0.17
reviews_per_month	-0.01	0.15	-0.031	-0.12	0.55	1	-0.0094	0.19
calculated_host_listings_count	0.02	-0.11	0.057	0.13	-0.072	-0.0094	1	0.23
availability_365	-0.011	0.083	0.082	0.14	0.17	0.19	0.23	1

Observation

From the above correlation we can see that number of reviews and reviews per month having the highest positive correlation, We can do a bivariate analysis between them but those two are already reviews and percent of reviews per month and now the second highest is calculated host listings and minimum nights lets do a bivariate analysis using scatter plot.



```
In [18]: sns.scatterplot(data.minimum_nights,data.calculated_host_listings_count)
         plt.show()

C:\Users\somas\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword ar
gs: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

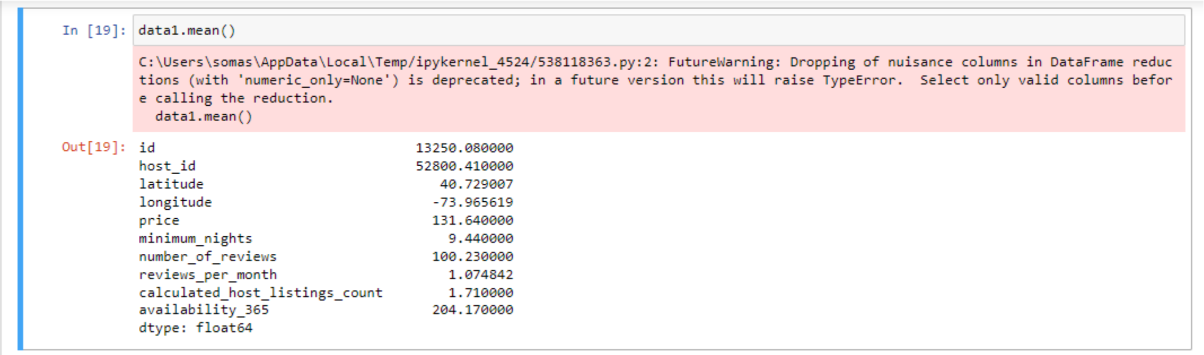
See there are less outliers and more relation between them . Above plot means the less the host came again the less the minimum nights.

<u>Statistical Analysis</u> - Python’s statistics is a built-in Python library for descriptive statistics. You can use it if your datasets are not too large or if you can't rely on importing other libraries. *S*tatistical modelling gives you the ability to asses, understand and make predictions about data, it is at the very bottom of inferential statistics and can be considered of those “must know” topics.The main two purposes of statistical analysis are to **describe** and to **investigate**:

To describe: estimate the moving average, impute missing data

To investigate: to search for a theoretical model that fits starting the observations we have, with the fundamentals we have like mean , median and mode etc..

From traditional analysis of variance and linear regression to exact methods and statistical visualization techniques, statistical programming is essential for making data-based decisions in every field.

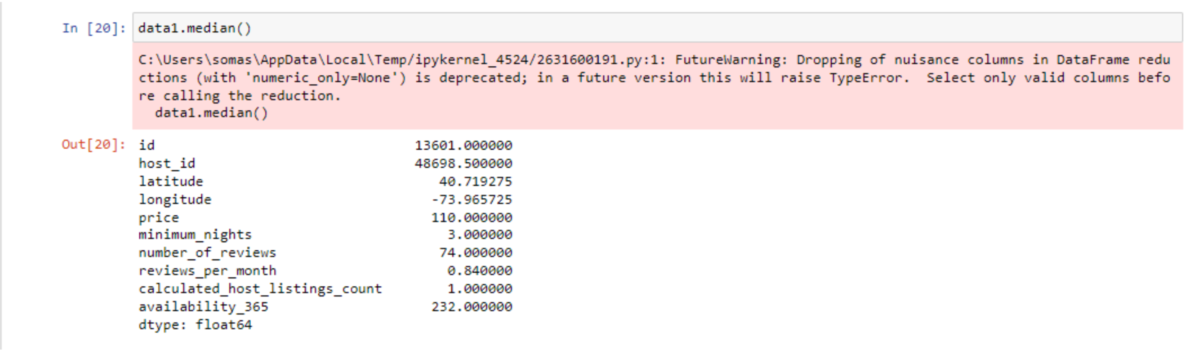


```
In [19]: data1.mean()

C:\Users\somas\AppData\Local\Temp/ipykernel_4524/538118363.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reduc
tions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns befor
e calling the reduction.
  data1.mean()

Out[19]: id                                13250.080000
         host_id                           52800.410000
         latitude                             40.729007
         longitude                           -73.965619
         price                               131.640000
         minimum_nights                        9.440000
         number_of_reviews                   100.230000
         reviews_per_month                     1.074842
         calculated_host_listings_count        1.710000
         availability_365                    204.170000
         dtype: float64
```

<u>Observation</u>

Here it shows the mean of the all the columns in the data. Its showing the values with e in them because it has more than 40,000 rows, all the mean values calculated resulting larger number. Mean means the average of all the values of values in columns

Here it shows the median of the all the columns in the data, median means middle value of the set data or the 50th percentile.

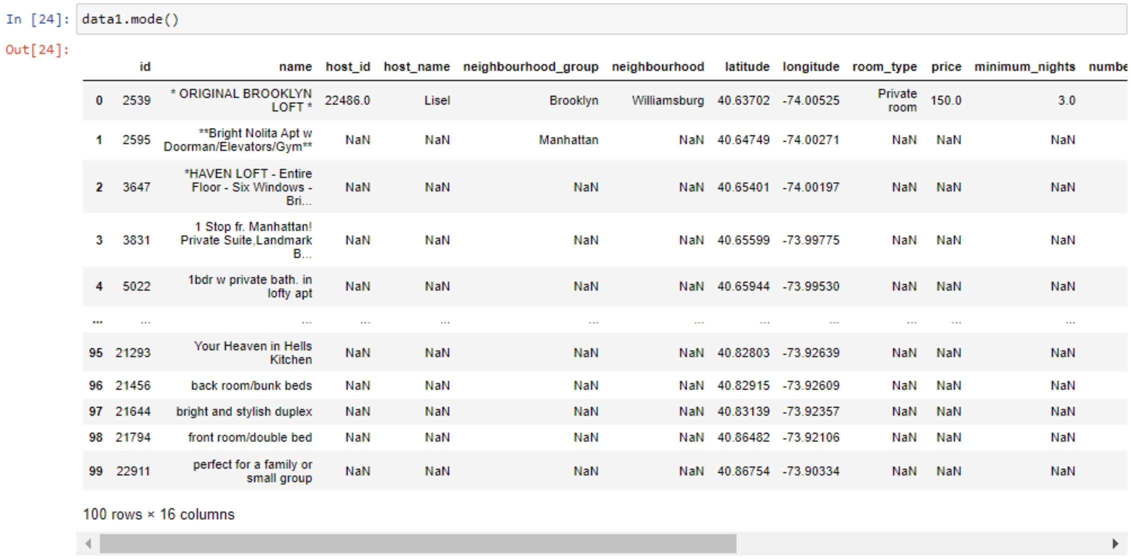


In [20]:
```
data1.median()
```

```
C:\Users\somas\AppData\Local\Temp/ipykernel_4524/2631600191.py:1: FutureWarning: Dropping of nuisance columns in DataFrame redu
ctions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns befo
re calling the reduction.
  data1.median()
```

Out[20]:
```
id                                 13601.000000
host_id                            48698.500000
latitude                              40.719275
longitude                            -73.965725
price                                110.000000
minimum_nights                         3.000000
number_of_reviews                     74.000000
reviews_per_month                      0.840000
calculated_host_listings_count         1.000000
availability_365                     232.000000
dtype: float64
```

Here it shows the mode of the all the columns in the data. Which means it shows the most repeated value in the column.

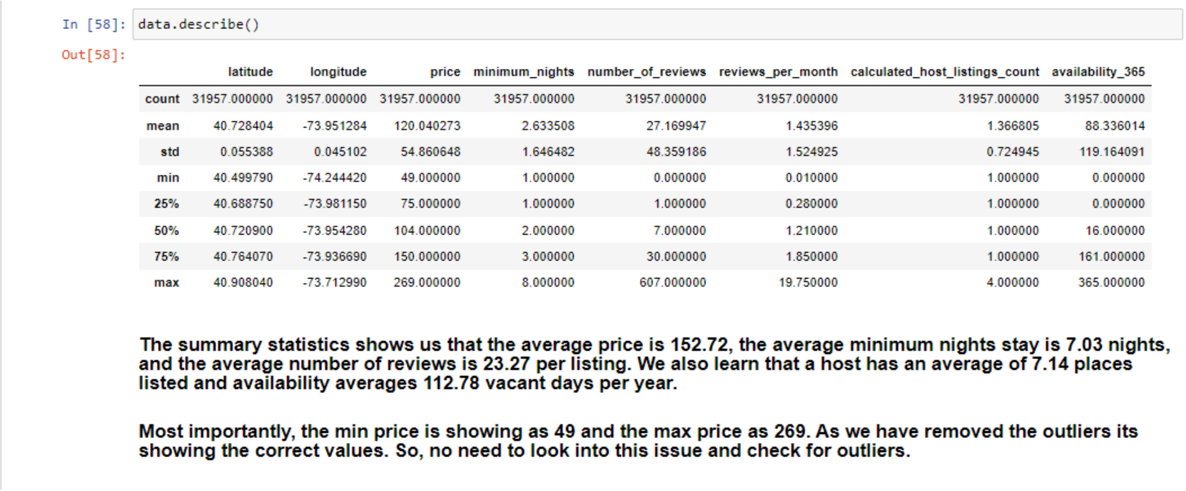


In [24]:
```
data1.mode()
```

Out[24]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	numbe
0	2539	* ORIGINAL BROOKLYN LOFT *	22486.0	Lisel	Brooklyn	Williamsburg	40.63702	-74.00525	Private room	150.0	3.0	
1	2595	**Bright Nolita Apt w Doorman/Elevators/Gym**	NaN	NaN	Manhattan	NaN	40.64749	-74.00271	NaN	NaN	NaN	
2	3647	*HAVEN LOFT - Entire Floor - Six Windows - Bri...	NaN	NaN	NaN	NaN	40.65401	-74.00197	NaN	NaN	NaN	
3	3831	1 Stop fr. Manhattan! Private Suite,Landmark B...	NaN	NaN	NaN	NaN	40.65599	-73.99775	NaN	NaN	NaN	
4	5022	1bdr w private bath. in lofty apt	NaN	NaN	NaN	NaN	40.65944	-73.99530	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
95	21293	Your Heaven in Hells Kitchen	NaN	NaN	NaN	NaN	40.82803	-73.92639	NaN	NaN	NaN	
96	21456	back room/bunk beds	NaN	NaN	NaN	NaN	40.82915	-73.92609	NaN	NaN	NaN	
97	21644	bright and stylish duplex	NaN	NaN	NaN	NaN	40.83139	-73.92357	NaN	NaN	NaN	
98	21794	front room/double bed	NaN	NaN	NaN	NaN	40.86482	-73.92106	NaN	NaN	NaN	
99	22911	perfect for a family or small group	NaN	NaN	NaN	NaN	40.86754	-73.90334	NaN	NaN	NaN	

100 rows × 16 columns



In [58]:
```
data.describe()
```

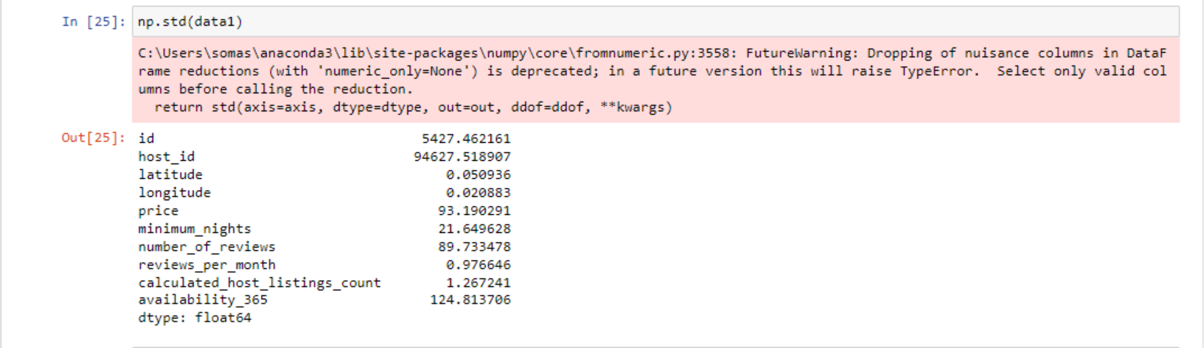
Out[58]:

	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	31957.000000	31957.000000	31957.000000	31957.000000	31957.000000	31957.000000	31957.000000	31957.000000
mean	40.728404	-73.951284	120.040273	2.633508	27.169947	1.435396	1.366805	88.336014
std	0.055388	0.045102	54.860648	1.646482	48.359186	1.524925	0.724945	119.164091
min	40.499790	-74.244420	49.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	40.688750	-73.981150	75.000000	1.000000	1.000000	0.280000	1.000000	0.000000
50%	40.720900	-73.954280	104.000000	2.000000	7.000000	1.210000	1.000000	16.000000
75%	40.764070	-73.936690	150.000000	3.000000	30.000000	1.850000	1.000000	161.000000
max	40.908040	-73.712990	269.000000	8.000000	607.000000	19.750000	4.000000	365.000000

The summary statistics shows us that the average price is 152.72, the average minimum nights stay is 7.03 nights, and the average number of reviews is 23.27 per listing. We also learn that a host has an average of 7.14 places listed and availability averages 112.78 vacant days per year.

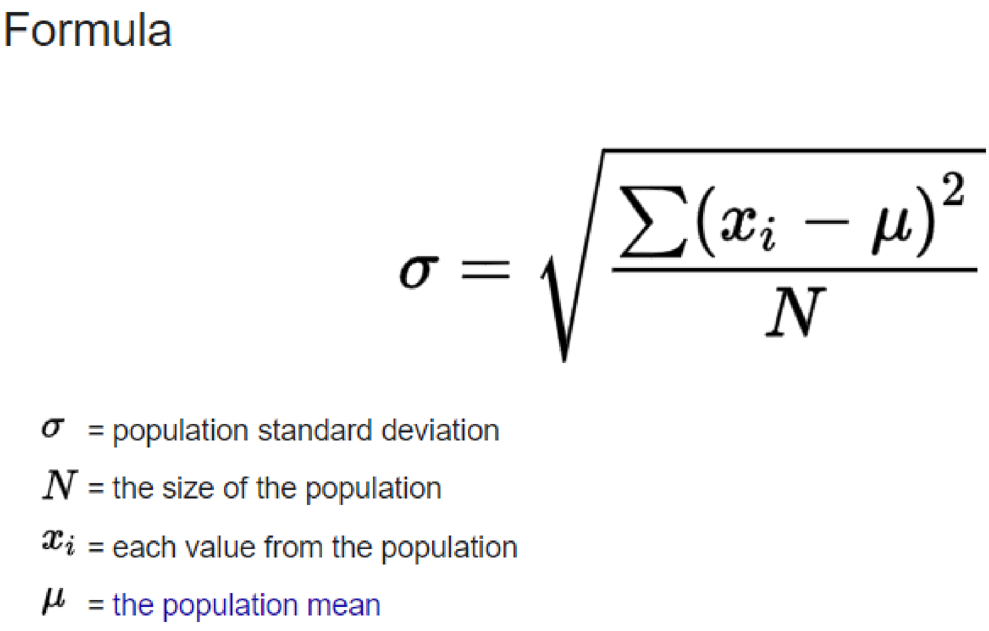
Most importantly, the min price is showing as 49 and the max price as 269. As we have removed the outliers its showing the correct values. So, no need to look into this issue and check for outliers.

Here it shows the standard deviation of the all the columns in the data. Which means the deviation of the data points from the mean of the set S.D =SQRT of variance.



```
In [25]: np.std(data1)

C:\Users\somas\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3558: FutureWarning: Dropping of nuisance columns in DataF
rame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid col
umns before calling the reduction.
  return std(axis=axis, dtype=dtype, out=out, ddof=ddof, **kwargs)

Out[25]: id                                 5427.462161
         host_id                           94627.518907
         latitude                              0.050936
         longitude                             0.020883
         price                                93.190291
         minimum_nights                       21.649628
         number_of_reviews                    89.733478
         reviews_per_month                     0.976646
         calculated_host_listings_count        1.267241
         availability_365                    124.813706
         dtype: float64
```

stdev() method calculates the standard deviation from a sample of data. Standard deviation is a measure of how spread out the numbers are. A large standard deviation indicates that the data is spread out, - a small standard deviation indicates that the data is clustered closely around the mean.



Formula

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation

N = the size of the population

x_i = each value from the population

μ = the population mean

Multivariate Analysis, Assumptions and Conclusions

Multivariate

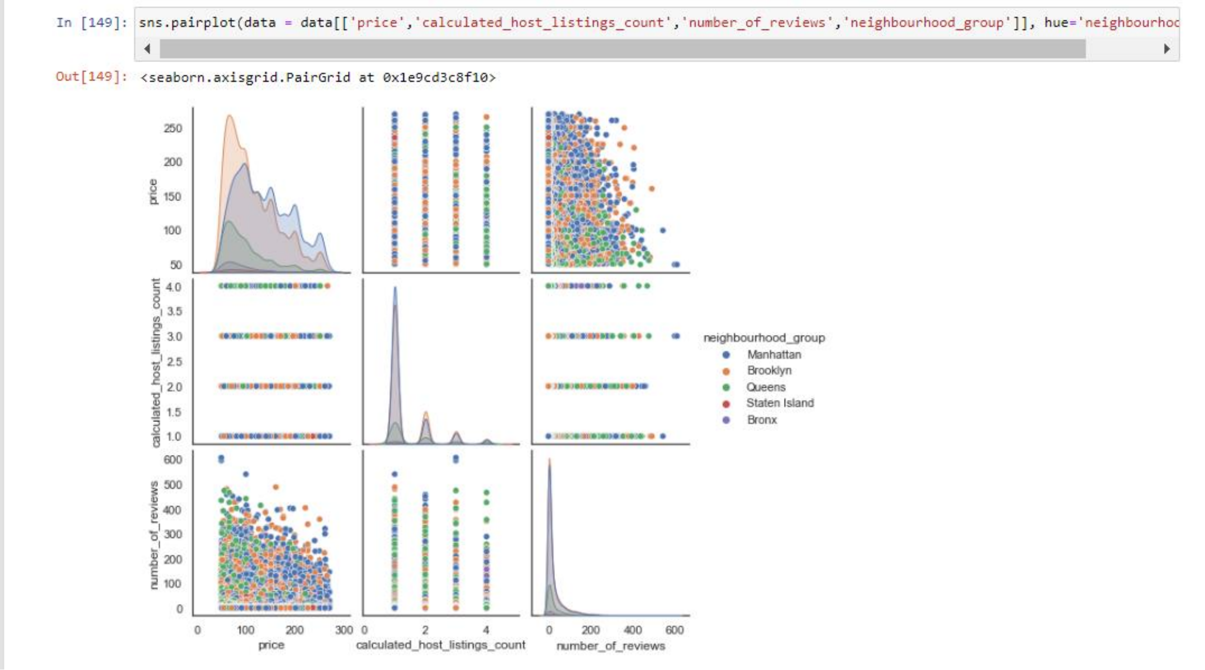
It is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analysed simultaneously with other variables.

The main advantage of multivariate analysis is that since it considers more than one factor of independent variables that influence the variability of dependent variables, the conclusion drawn is more accurate.

The conclusions are more realistic and nearer to the real-life situation.

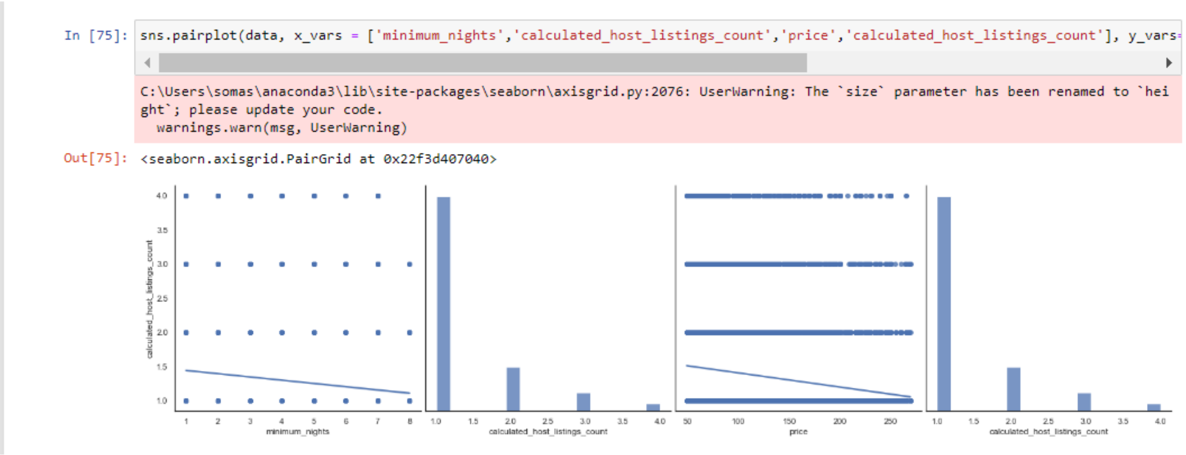
Pair plot: A pair plot a pairwise relationships in a dataset. The pair plot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.

It is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters



```
In [149]: sns.pairplot(data = data[['price','calculated_host_listings_count','number_of_reviews','neighbourhood_group']], hue='neighbourhoo
Out[149]: <seaborn.axisgrid.PairGrid at 0x1e9cd3c8f10>
```

As we can see from the above pair plot, it shows the relation between price, calculated host listing, number of reviews by comparing it with the neighbourhood.



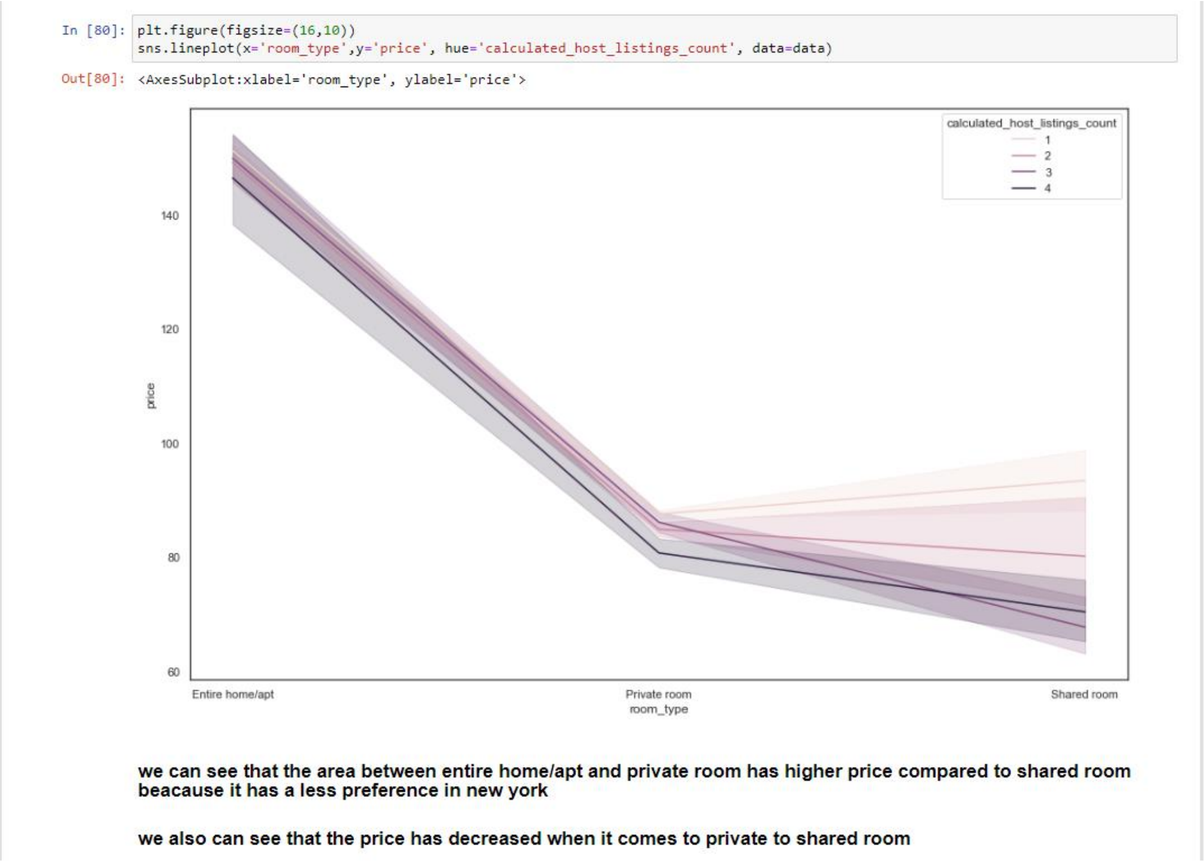
```
In [75]: sns.pairplot(data, x_vars = ['minimum_nights','calculated_host_listings_count','price','calculated_host_listings_count'], y_vars=
```

```
C:\Users\somas\anaconda3\lib\site-packages\seaborn\axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `hei
ght`; please update your code.
  warnings.warn(msg, UserWarning)
```

```
Out[75]: <seaborn.axisgrid.PairGrid at 0x22f3d407040>
```

And we again did the pair plot but different, now we have taken comparison with minimum nights. From above we can see the relation clearly between those columns.

Line Plot: A line plot is a graph that displays data using a number line. To create a line plot, first create a number line that includes all the values in the data set.



```
In [80]: plt.figure(figsize=(16,10))
         sns.lineplot(x='room_type',y='price', hue='calculated_host_listings_count', data=data)
```

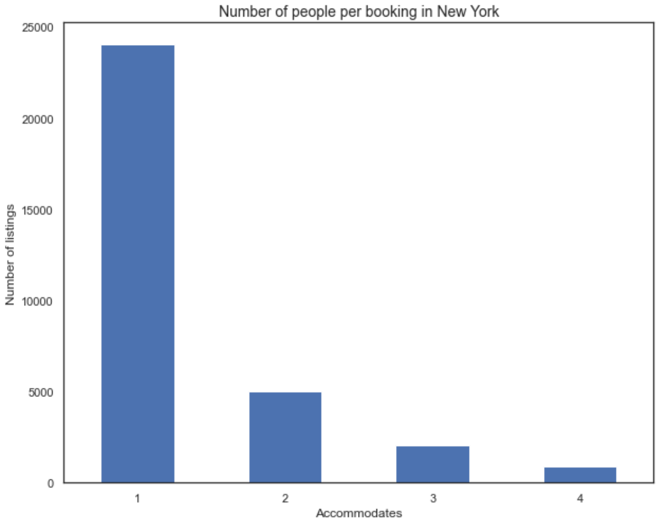
```
Out[80]: <AxesSubplot:xlabel='room_type', ylabel='price'>
```

we can see that the area between entire home/apt and private room has higher price compared to shared room beacause it has a less preference in new york

we also can see that the price has decreased when it comes to private to shared room

Assumptions and Conclusions

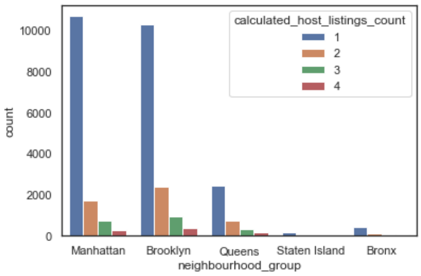
```
In [46]: Feq = data['calculated_host_listings_count'].value_counts().sort_index()
         Feq.plot.bar(figsize=(10,8), width=0.5, rot=0)
         plt.title('Number of people per booking in New York', fontsize=14)
         plt.ylabel('Number of listings', fontsize=12)
         plt.xlabel('Accommodates', fontsize=12)
         plt.show()
```



The highest number of accomodates is only one and the least is 4, which clearly indicates that is the reason for low listings on shared room.

```
In [54]: sns.countplot(data=data, x='neighbourhood_group', hue='calculated_host_listings_count')
```

```
Out[54]: <AxesSubplot:xlabel='neighbourhood_group', ylabel='count'>
```



Lets see which neighbourhood has more reviews

```
In [55]: areas_reviews = data.groupby(['neighbourhood_group'])['number_of_reviews'].max().reset_index().sort_values(by = 'number_of_reviev
         areas_reviews
```

Out[55]:

	neighbourhood_group	number_of_reviews
2	Manhattan	607
1	Brooklyn	488
3	Queens	474
0	Bronx	321
4	Staten Island	242

3. What can we learn about different areas , prices and reviews?

Predictions

In [47]:

```python
host_areas = data.groupby(['host_name','neighbourhood_group'])['calculated_host_listings_count'].max().reset_index()
host_areas.sort_values('calculated_host_listings_count',ascending = False).head(10)
```

Out[47]:

	host_name	neighbourhood_group	calculated_host_listings_count
2133	Chris And Zaneta	Brooklyn	4
8003	Mina	Queens	4
784	Anna	Brooklyn	4
4095	Gio	Manhattan	4
8495	Nick	Manhattan	4
10577	Sofia	Manhattan	4
2629	David	Manhattan	4
2630	David	Queens	4
1067	Askhat	Brooklyn	4
11920	Yaron	Bronx	4

The most of the listings are from Manhattan by Sonder, Blueground, Kara etc.,

In [48]:

```python
areas_review = data.groupby(['neighbourhood_group'])['number_of_reviews'].max().reset_index()
areas_review
```

Out[48]:

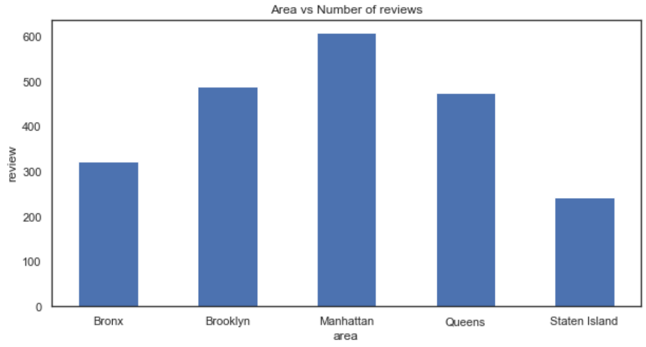
	neighbourhood_group	number_of_reviews
0	Bronx	321
1	Brooklyn	488
2	Manhattan	607
3	Queens	474
4	Staten Island	242

Here these neighbourhood groups have most no.of reviews which means they must be tourist places, because tourist places get most reviews

now lets see more clearly by plotting the above in a bar chart

In [49]:

```python
area = areas_review['neighbourhood_group']
review = areas_review['number_of_reviews']

fig = plt.figure(figsize = (10, 5))

plt.bar(area, review, width = 0.5)
plt.xlabel("area")
plt.ylabel("review")
plt.title("Area vs Number of reviews")
plt.show()
```



The most reviewed areas are Queens and Manhattan, By this we can say most tourists are visiting these areas in NewYork which means there will be a lot of traffic in the above areas

Lets see which neighbourhood has more reviews

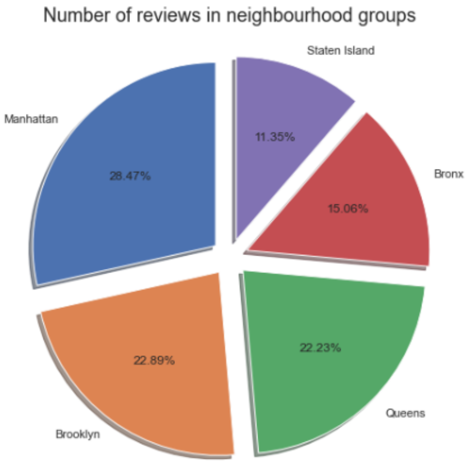
```
In [55]: areas_reviews = data.groupby(['neighbourhood_group'])['number_of_reviews'].max().reset_index().sort_values(by = 'number_of_reviev
         areas_reviews
```

Out[55]:

	neighbourhood_group	number_of_reviews
2	Manhattan	607
1	Brooklyn	488
3	Queens	474
0	Bronx	321
4	Staten Island	242

Lets visualize the above details in pie chart

```
In [56]: plt.rcParams['figure.figsize'] = (13,8)
         reviews = areas_reviews['number_of_reviews']
         plt.pie(reviews, labels = areas_reviews['neighbourhood_group'], autopct ='%0.2f%%', startangle =90, explode = [0.1,0.1,0.1,0.1,0.
         plt.title('Number of reviews in neighbourhood groups', {'fontsize': 18})
         plt.show()
```



Here, The share percentage of reviews for each neighbourhood group is depicted above. It can be stated that Queens, Manhattan have majority of reviews which implies that people are liking these neighbourhoods. And maximum revenue will be generated from these neighbourhood groups.

Lets see which are the busiest hosts from that we may find some relation.

In [62]:

```python
busy_hosts = data.groupby(['host_name','room_type','neighbourhood_group'])['number_of_reviews'].max().reset_index()
busy_hosts = busy_hosts.sort_values(by='number_of_reviews', ascending=False).head(10)
busy_hosts
```

Out[62]:

	host_name	room_type	neighbourhood_group	number_of_reviews
6689	Jj	Private room	Manhattan	607
2298	Carol	Private room	Manhattan	540
1294	Asa	Entire home/apt	Brooklyn	488
14446	Wanda	Private room	Brooklyn	480
8398	Linda	Private room	Queens	474
925	Angela	Private room	Queens	466
219	Agnes	Private room	Manhattan	458
6798	John	Private room	Manhattan	451
3481	Dennis & Naoko	Entire home/apt	Queens	441
10024	Miss Dy	Entire home/apt	Queens	434

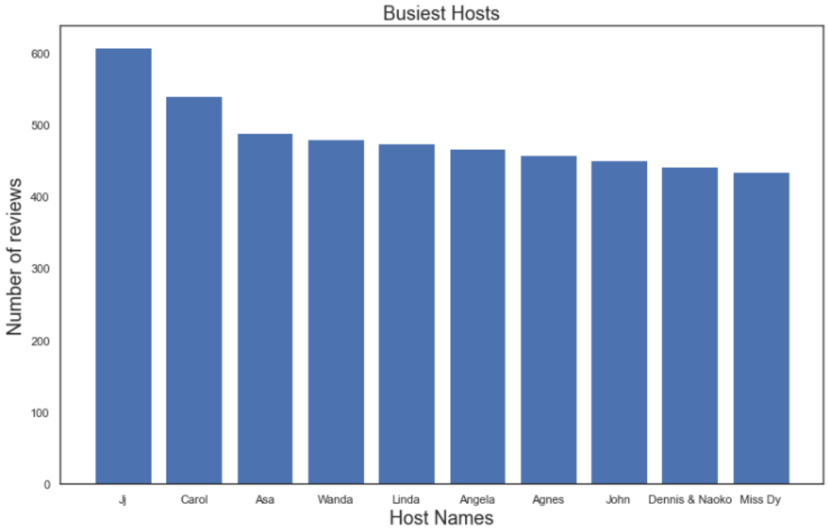
Lets plot them in bar chart

In [64]:

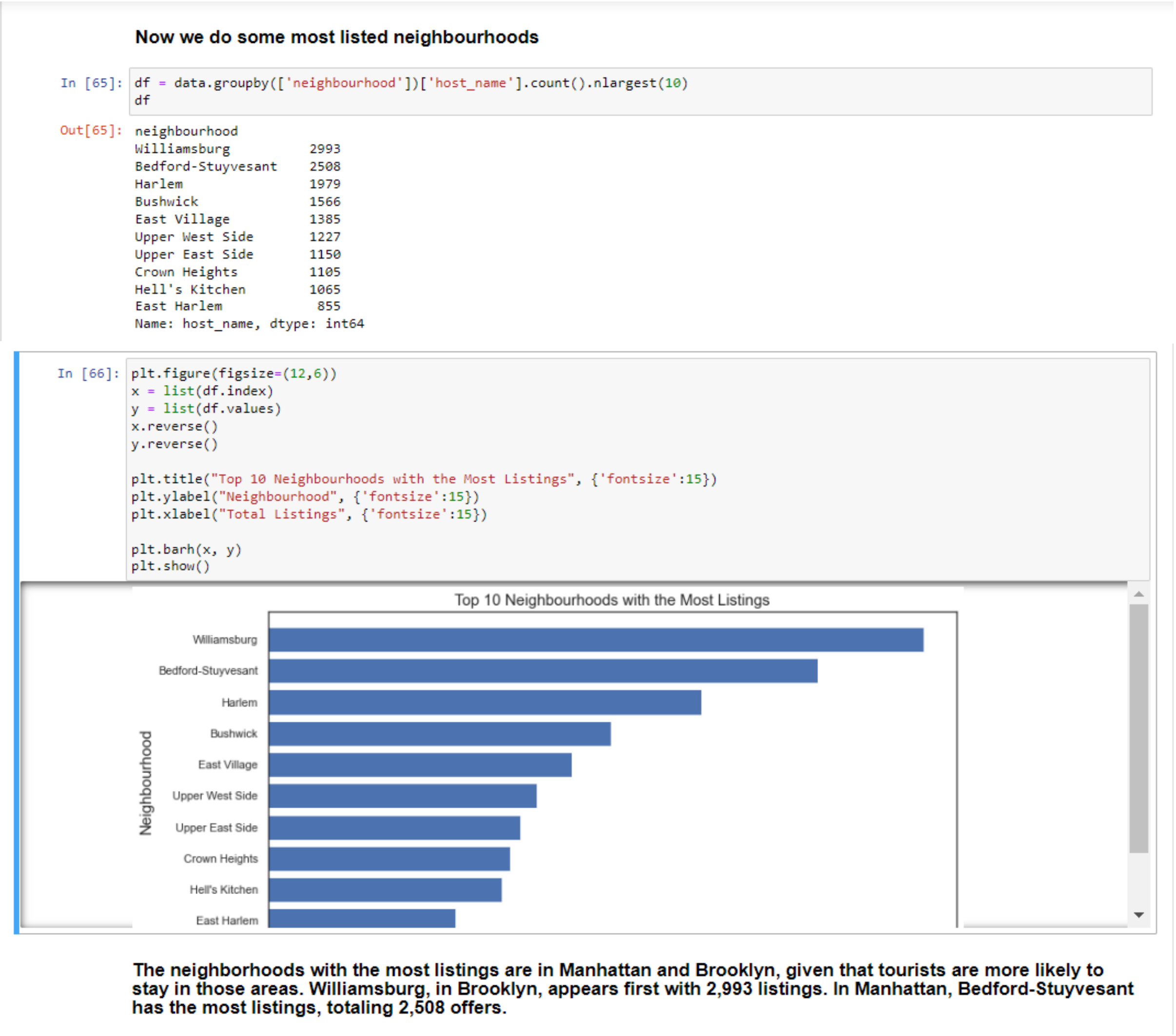
```python
host_name = busy_hosts['host_name']
reviews = busy_hosts['number_of_reviews']
plt.title('Busiest Hosts', {'fontsize':18})
plt.xlabel('Host Names',{'fontsize':18})
plt.ylabel('Number of reviews',{'fontsize':18})
plt.bar(host_name, reviews)
```

Out[64]:

```
<BarContainer object of 10 artists>
```



Above hosts are the most busiest and they mostly host entire home/apt or private room because we know from plotting the bar betweeen most preferred room type.



Now we do some most listed neighbourhoods

```
In [65]: df = data.groupby(['neighbourhood'])['host_name'].count().nlargest(10)
         df
```

```
Out[65]: neighbourhood
         Williamsburg          2993
         Bedford-Stuyvesant    2508
         Harlem                1979
         Bushwick              1566
         East Village          1385
         Upper West Side       1227
         Upper East Side       1150
         Crown Heights         1105
         Hell's Kitchen        1065
         East Harlem            855
         Name: host_name, dtype: int64
```

```
In [66]: plt.figure(figsize=(12,6))
         x = list(df.index)
         y = list(df.values)
         x.reverse()
         y.reverse()

         plt.title("Top 10 Neighbourhoods with the Most Listings", {'fontsize':15})
         plt.ylabel("Neighbourhood", {'fontsize':15})
         plt.xlabel("Total Listings", {'fontsize':15})

         plt.barh(x, y)
         plt.show()
```

The neighborhoods with the most listings are in Manhattan and Brooklyn, given that tourists are more likely to stay in those areas. Williamsburg, in Brooklyn, appears first with 2,993 listings. In Manhattan, Bedford-Stuyvesant has the most listings, totaling 2,508 offers.

Conclusion: Interesting insights from Analysis The above analysis helped us understand the New York Air Bnb dataset better. Following insights were drawn from it:

Private room is the most common listing type in all neighbourhood’s except Manhattan where Entire Home/apartment is the most common type.

Shared room is the least common type of listings.

Average price of listings is the highest for Manhattan followed by Brooklyn.

Bronx has the cheapest listings with an average price of 87.5 USD.

Average price is the highest for Entire home/apartment followed by private room and shared room.

The factors that impact listing prices are:

- room type
- neighbourhood group
- longitude
- availability_365
- minimum nights
- calculated_host_listings_count
- latitude
- number_of_reviews
- reviews_per_month

This information can be helpful for multiple categories of the society such as tourists, house owners, real estate agents etc. It can help customers decide which neighbourhood’s they should investigate depending on their needs and house owners can decide on the prices they set on the listings taking all these factors into consideration.

Through this exploratory data analysis and visualization, we gained several interesting insights into the Airbnb rental market. This Airbnb dataset for 2019 year appeared to be a very rich dataset with a variety of columns that allowed us to do deep data exploration on each significant column presented. After that, we proceeded with analysing boroughs and neighbourhood listing densities and what areas were more popular than another, their price variations, their availability as per room types. Also, we emphasized on key findings like room types and their preferred stays by guests, the top reviewed hosts and their listings.

I have used seaborn and matplotlib for creating all the visualizations.

Here is the link to my dataset - https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data .