DATA SCIENCE

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS
INSTITUTE OF TECHNOLOGY

EDA COMPONENT

TEAM MEMBERS:

NAME	PRN
ADITI SAXENA	20070123004
REWAA MITAL	20070122110
ADITYA KULEYAR	20070123003

EDA:

It is a data exploration technique that helps us to understand the various aspects of data.

OBJECTIVE:

Used to filter the data from redundancies.

STEPS:

- Understand the variables and and the dataset
- Then clean the data
- Third is to analyze the relationship between variables.

This dataset was **created by DGOMONOV** and was **created 3 years** ago(2019)and uploaded on Kaggle

Size of data: There are 48895 rows and 16 columns in our dataset. File size of 7.1Mb

HISTORY:

Since 2008, travelers and hosts have used Airbnb to expand their travel options and present a more unique and personal way to experience the world. Today, Airbnb is a unique service used by people around the world. Data analysts are becoming a crucial factor for the company which has served millions of listings through Airbnb. These inputs generate a lot of data that can be analyzed and used for security, business decisions, understanding customer and supplier behavior on the platform, implementing innovative add-on services, guiding marketing initiatives , And much more.

DOWNLOADED FROM:

https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data

This public dataset is part of Airbnb, and the original source can be found on this website.

IN THE DATASET:

This data file contains all the information needed to learn more about hosts, geographic availability, and metrics needed to make predictions and draw conclusions. The data also shows that there may be instances where a particular host has co-hosted another neighborhood's property/listing on Airbnb.

In basic inspection, a particular property name will have a particular host name hosted by the same person, but a particular host name can have multiple properties in an area. So host_name is a categorical variable here. Neighborhood groups (consisting of Manhattan, Brooklyn, Queens, Bronx, Staten Island), neighborhood and room type (private, shared, Entire home / apartment) also fall into this category.
~ id, latitude, longitude, price, minimum_nights, number_review, last_view, reviews_per_month, host_listing_count_calculated, 365_ availability are numeric variables.

EDA COMPONENT

IMPORTING THE DATA

jupyter Data Science CA3 Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [3]:
```
#upload dataset
ABdata=pd.read_csv('/Users/rewaamital/Desktop/AB_NYC.csv' )
```

In [4]:
```
ABdata #displaying data #displaying data
```

Out[4]:

rhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_ho
ington	40.64749	-73.97237	Private room	149	1	9	2018-10-19	0.21	
dtown	40.75362	-73.98377	Entire home/apt	225	1	45	2019-05-21	0.38	
Harlem	40.80902	-73.94190	Private room	150	3	0	NaN	NaN	
on Hill	40.68514	-73.95976	Entire home/apt	89	1	270	2019-07-05	4.64	
Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	2018-11-19	0.10	
...	...	...	...	...	...	...	...	...	
dford-vesant	40.67853	-73.94995	Private room	70	2	0	NaN	NaN	
shwick	40.70184	-73.93317	Private room	40	4	0	NaN	NaN	

UNDERSTANDING THE DATA

1) BASIC INFORMATION:

```
In [5]: #basic information
        ABdata.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
```

ABdata.info()-
This function will give us the basic information about the dataset.it counts all the rows and columns and provides the type id datatypes of the dataset.

2)NUMBER OF ROWS AND COLUMNS:

```
In [30]: #print number of rows and columns
         ABdata.shape

Out[30]: (48895, 16)
```

3)DESCRIPTIVE STATISTICS

```
In [6]: #Descriptive Statistics
        ABdata.describe()
```

Out[6]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.0
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.1
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.9
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.0
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.0
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.0
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.0
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.0

ABdata.describe()-

Its central issue is that it gives the mean , standard deviation, least deviation, inter quartile scope of the dataset regarding every one of the number fields and all out and it are not considered to string capabilities.

4) DATA TYPES OF COLUMNS:

```
In [19]: #tells us about the data types of columns
         ABdata.dtypes

Out[19]: id                                int64
         name                             object
         host_id                           int64
         host_name                        object
         neighbourhood_group              object
         neighbourhood                    object
         latitude                        float64
         longitude                       float64
         room_type                        object
         price                             int64
         minimum_nights                    int64
         number_of_reviews                 int64
         last_review                      object
         reviews_per_month                object
         calculated_host_listings_count    int64
         availability_365                  int64
         dtype: object
```

Abdata.dtypes-

This command print the datatypes of each column of the dataset and helps when the joining operation of two dataset is done.

5) FILTER THE DATA:

```
In [23]: #Filter the data and print first 5 values
         ABdata[ABdata['room_type']== 'Private room'].head()
```

Out[23]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0
6	5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	Private room	60	45	49
7	5178	Large Furnished Room Near B'way	8967	Shunichi	Manhattan	Hell's Kitchen	40.76489	-73.98493	Private room	79	2	430
8	5203	Cozy Clean Guest Room - Family Apt	7490	MaryEllen	Manhattan	Upper West Side	40.80178	-73.96723	Private room	79	2	118

This command will select the column room type and print all the rows which have a private room as data entry in it.The head property ensures that only the first 5 rows are printed.

CLEANING THE DATA

1)DUPLICATE VALUES:

```
In [7]: #Number of duplicate values in the dataset
        ABdata.duplicated().sum()

Out[7]: 0
```

ABdata.duplicated.sum()-

Function used to do the sum of duplicate values present if any. It will show the number of duplicate values if they are present in the data.

2)NUMBER UNIQUE VALUES:

```
In [31]: #print number of unique items in reach column
         ABdata.nunique()
```

```
Out[31]: id                                48895
         name                              47906
         host_id                           37457
         host_name                         11453
         neighbourhood_group                   5
         neighbourhood                       221
         latitude                          19048
         longitude                         14718
         room_type                             3
         price                               674
         minimum_nights                      109
         number_of_reviews                   394
         last_review                        1765
         reviews_per_month                   938
         calculated_host_listings_count       47
         availability_365                    366
         dtype: int64
```

3) UNIQUE VALUES OF COLUMNS:

```
In [8]: #find unique values in a particular column
        print(ABdata['neighbourhood_group'].unique())
```

```
['Brooklyn' 'Manhattan' 'Queens' 'Staten Island' 'Bronx']
```

```
In [9]: print(ABdata['room_type'].unique())
```

```
['Private room' 'Entire home/apt' 'Shared room']
```

We can find the number of unique values in the particular column using the unique() function in python.

4) NUMBER OF NULL VALUES IN COLUMNS

```
In [12]: ABdata.isnull().sum()
```

```
Out[12]: id                                     0
         name                                  16
         host_id                                0
         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
```

Finding the null values is the most important step in the EDA. ensuring the quality of data is paramount.

5) TO REPLACE ALL NULL VALUES

```
In [18]: #replace all null values
         ABdata.replace(np.nan,'0',inplace=True)
```

We use a replace()/fillna() function to replace all the null values with a specific data.

```
In [17]: ABdata.isnull().sum()

Out[17]: 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
```

As we can see wherever the null value was there it is filled with zero.

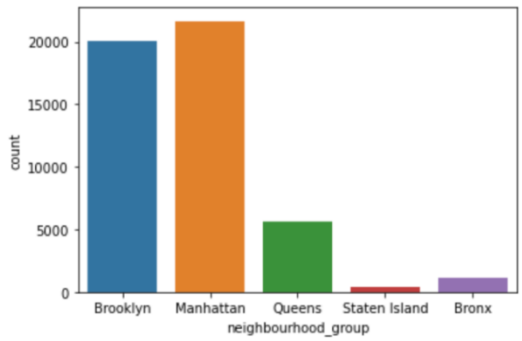
ANALYSIS OF RELATIONSHIP BETWEEN VARIABLES

1)GRAPH FOR UNIQUE COUNTS IN A COLUMN:

```
In [10]: #Visualize the unique counts
         sns.countplot(ABdata['neighbourhood_group'])
```

```
/Users/rewaamital/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the follow
ing variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing o
ther arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[10]: <AxesSubplot:xlabel='neighbourhood_group', ylabel='count'>
```

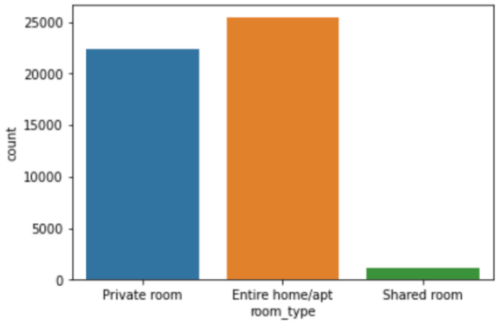


The countplot () method is used to display the observation counts in each categorical bin using bars.From the categorical boxplot above, we can conclude that Manhattan seems to be have more than 20000 Airbnb's in its city .On average, Brooklyn has almost 20000 Airbnb's followed by Queens with appox 5000 Air BNB's.

```
In [11]: sns.countplot(ABdata['room_type'])
```

```
/Users/rewaamital/opt/anaconda3/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the follow
ing variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing o
ther arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

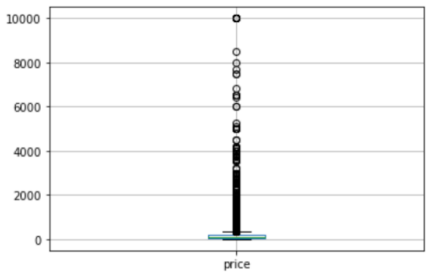
```
Out[11]: <AxesSubplot:xlabel='room_type', ylabel='count'>
```



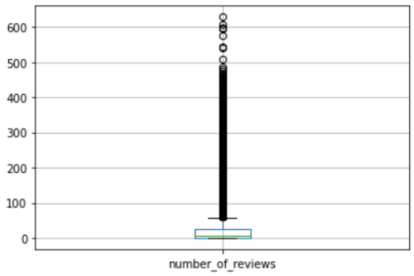
It clearly indicates most of the Airbnb's are Entire home/Apartment type, with a count of around 25000, followed by private rooms and then shared rooms.

2) BOXPLOT:

```
In [24]: #boxplot for the column 'price'
         ABdata[['price']].boxplot()

Out[24]: <AxesSubplot:>
```



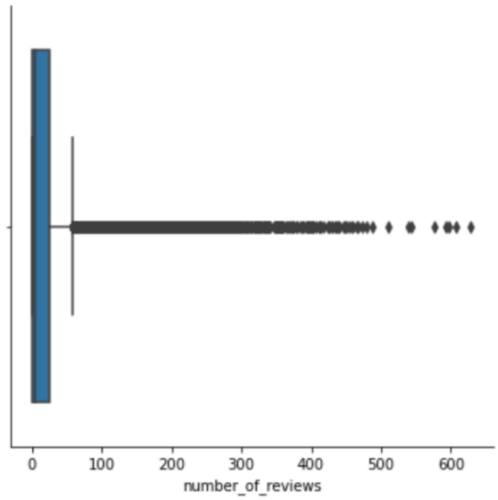

```
In [26]: #boxplot for the column 'number_of_reviews'
         ABdata[['number_of_reviews']].boxplot()

Out[26]: <AxesSubplot:>
```



A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the inter-quartile range.

This interprets that the interquartile range of the price , range from 0 to 200 dollars and the outliers range from 300 and go upto 10000 dollars.

```
In [39]: #boxplot
         sns.catplot(x='number_of_reviews', kind='box', data=ABdata)

Out[39]: <seaborn.axisgrid.FacetGrid at 0x7fe05fd569d0>
```



In the second graph we can see that the inter-quartile range of the number_of_reviews range from 0-40 and the whiskers extend from 60 and go upto more than 600 reviews.

3) CORRELATION PLOT AND MATRIX:

```
In [27]: #correlation matrix
         ABdata.corr()
```

Out[27]:

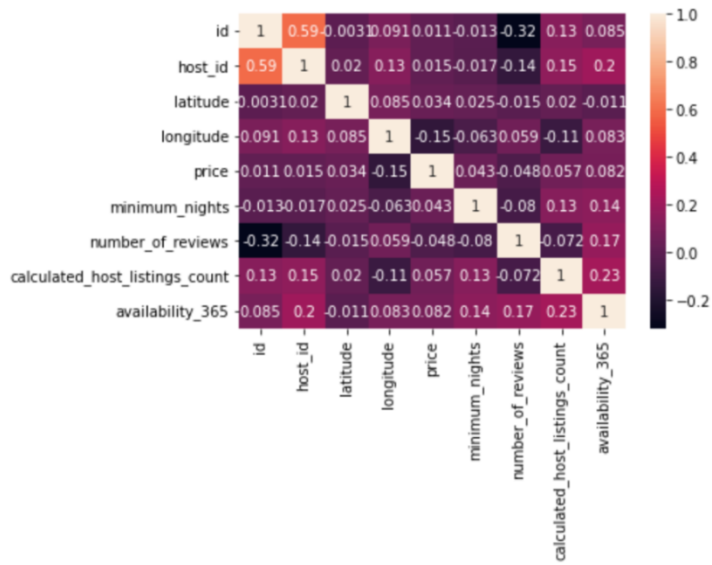
	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	calculated_host_listings_count	availabil
id	1.000000	0.588290	-0.003125	0.090908	0.010619	-0.013224	-0.319760	0.133272	0.
host_id	0.588290	1.000000	0.020224	0.127055	0.015309	-0.017364	-0.140106	0.154950	0.
latitude	-0.003125	0.020224	1.000000	0.084788	0.033939	0.024869	-0.015389	0.019517	-0.
longitude	0.090908	0.127055	0.084788	1.000000	-0.150019	-0.062747	0.059094	-0.114713	0.
price	0.010619	0.015309	0.033939	-0.150019	1.000000	0.042799	-0.047954	0.057472	0.
minimum_nights	-0.013224	-0.017364	0.024869	-0.062747	0.042799	1.000000	-0.080116	0.127960	0.
number_of_reviews	-0.319760	-0.140106	-0.015389	0.059094	-0.047954	-0.080116	1.000000	-0.072376	0.
calculated_host_listings_count	0.133272	0.154950	0.019517	-0.114713	0.057472	0.127960	-0.072376	1.000000	0.
availability_365	0.085468	0.203492	-0.010983	0.082731	0.081829	0.144303	0.172028	0.225701	1.

To find the correlation among the variables, we can make use of the correlation function. This will give you a fair idea of the correlation strength between different variables.

.This is the correlation matrix with the range from +1 to -1 where +1 is highly and positively correlated and -1 will be highly negatively correlated.

```
In [32]: #correlation plot
         sns.heatmap(ABdata.corr(),annot=True)
```

```
Out[32]: <AxesSubplot:>
```



Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred.

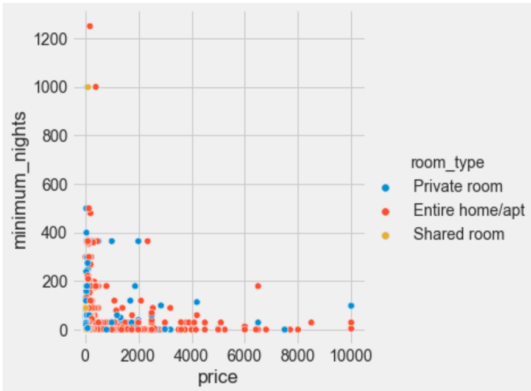
Using seaborn heatmap *strength* between the variables used is understood.

There's correlation among host_id to reveiws_per_month & availability_365 (sequential color bar is used between value and color). Also there's noticeable correlation between min_nights, no_of_listings_count & availability_365. Price also shows some correlation with availability_365 & host_listings_count.

no_of_reviews and reviews_per_month gives almost the same information. so we can carry out analysis with any of the two variables. Also, no_of_reviews is correlated to availability_365!

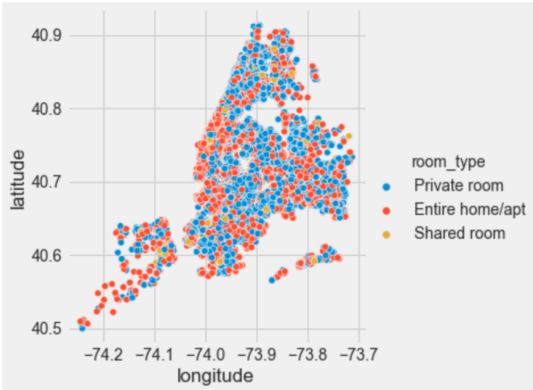
4) SCATTER PLOT:

```
In [14]: #scatter plot
         sns.relplot(x="price", y="minimum_nights", hue='room_type',data=ABdata)

Out[14]: <seaborn.axisgrid.FacetGrid at 0x7f829c134490>
```



By the scatterplot of price vs minimum nights we can conclude that hostd mostly hosted an entire home or apartment over a shared or private room. The price range of these properties were mostly between 0-2000 and the minimum nights for which these properties, between the price range of 0-2000, were booked for 0-200 nights. We can also see that the Shared rooms are priced lesser as compared to most of the private rooms or apartments. The properties which were priced at higher amount were booked for least number of nights usually.

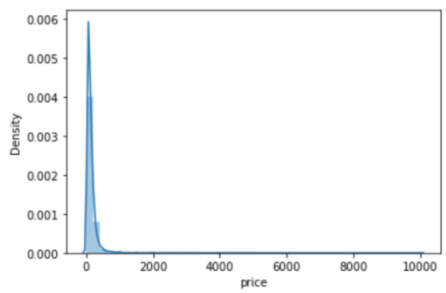
```
In [7]: #scatter plot
        sns.relplot(x="longitude", y="latitude", hue='room_type',data=ABdata)

Out[7]: <seaborn.axisgrid.FacetGrid at 0x7f82b2f53400>
```



By the scatterplot of longitude vs latitude we can conclude that most of the Airbnb's were between -74.0 to -73.75 longitude and 40.6 to 40.75 latitude.

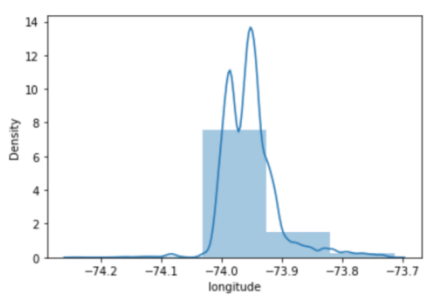
5) HISTOGRAM(WITH BINS):

```
In [37]: #histogram
         sns.distplot(ABdata['price'])

/Users/rewaamital/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot`
is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f
igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

Out[37]: <AxesSubplot:xlabel='price', ylabel='Density'>
```



Histograms represent the data distribution by forming bins along the range of the data and then drawing bars to show the number of observations that fall in each bin,In this graph it is shown that mainly the price is hiked between the range from 0-1000

```
In [38]: #histogram with bins
         sns.distplot(ABdata['longitude'], bins=5)

/Users/rewaamital/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot`
is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f
igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

Out[38]: <AxesSubplot:xlabel='longitude', ylabel='Density'>
```

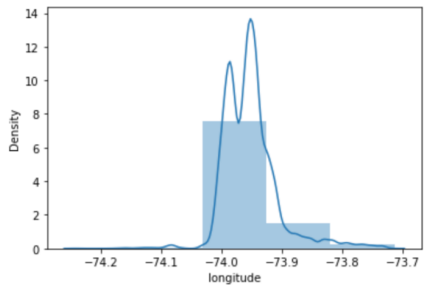


This graph interprets that the density rises from the longitudes ranging from -74.0 to -73.9 and it gradually decreases.

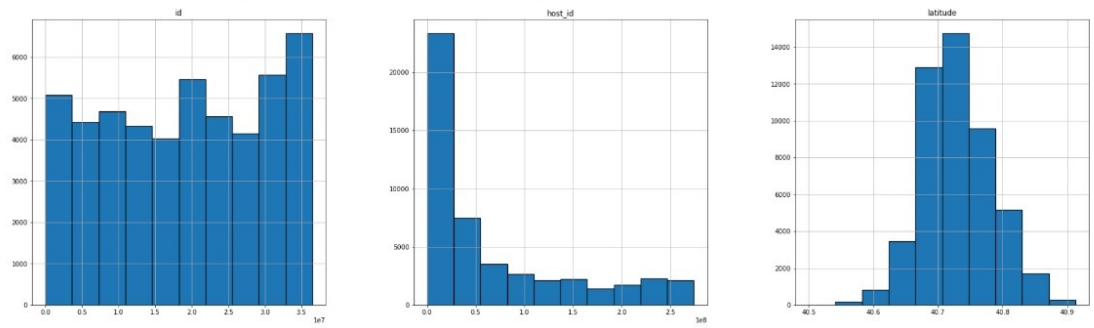
```
In [38]: #histogram with bins
         sns.distplot(ABdata['longitude'], bins=5)
```

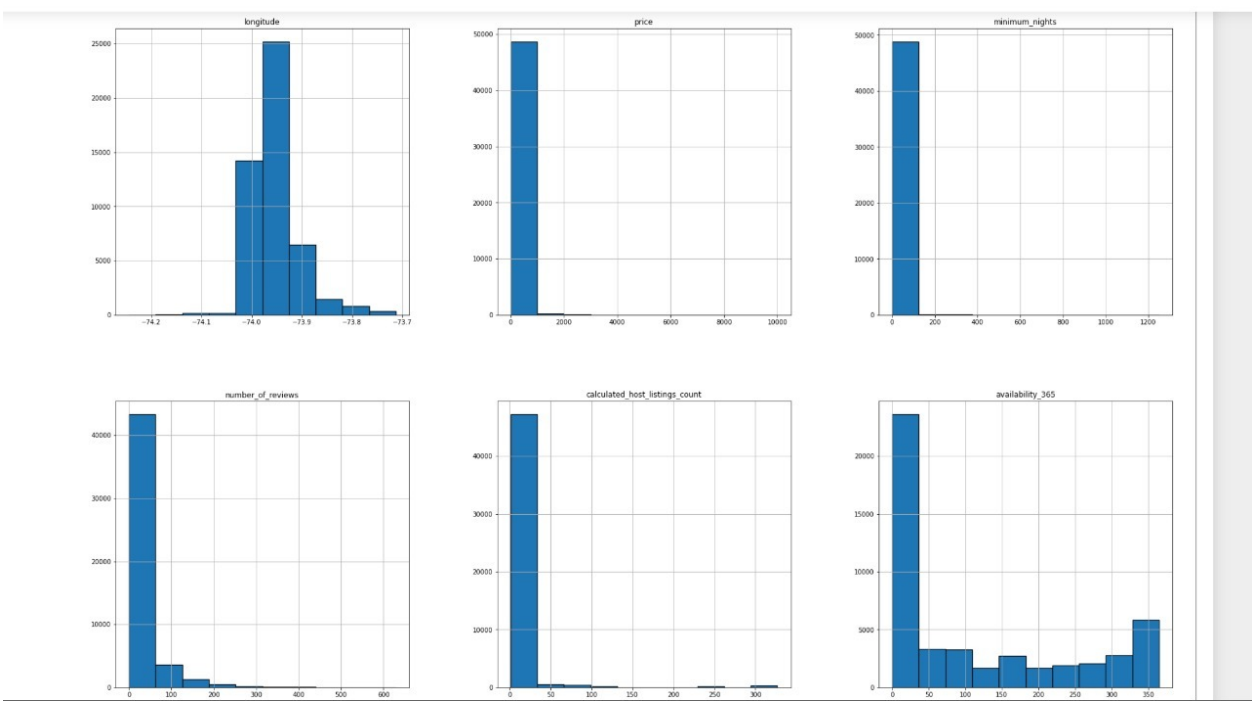
```
/Users/rewaamital/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot`
is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f
igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[38]: <AxesSubplot:xlabel='longitude', ylabel='Density'>
```




```
In [60]: ABdata.hist(edgecolor = "black", linewidth=1.2, figsize=(30,30));
```





Visualizing the distribution for every "feature".

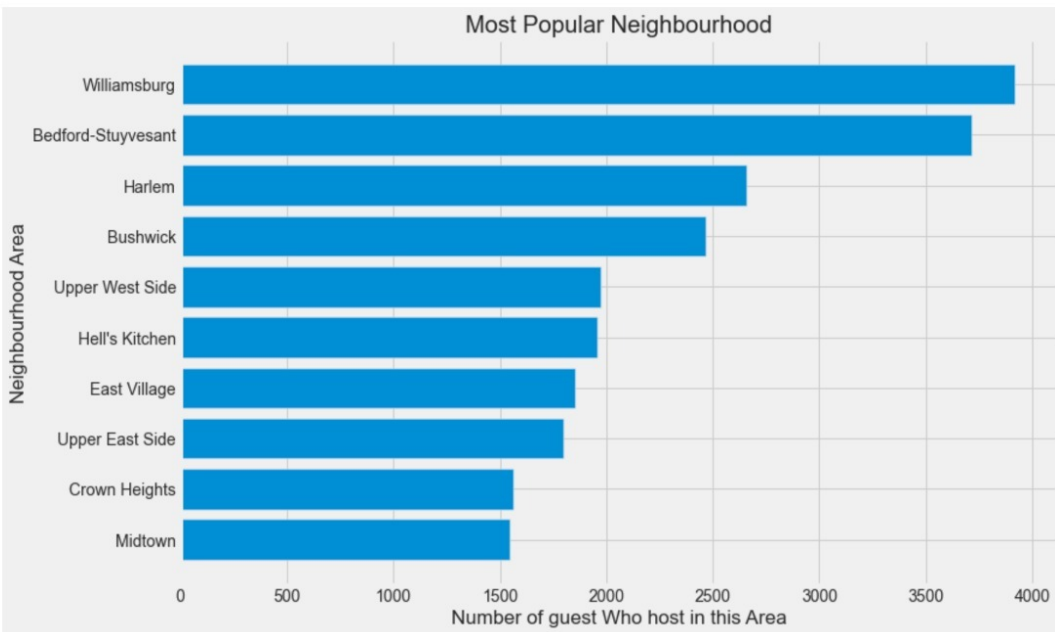
It displays the no.of airbnbs with respect to the given parameters.

6) Bar Graph:

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.

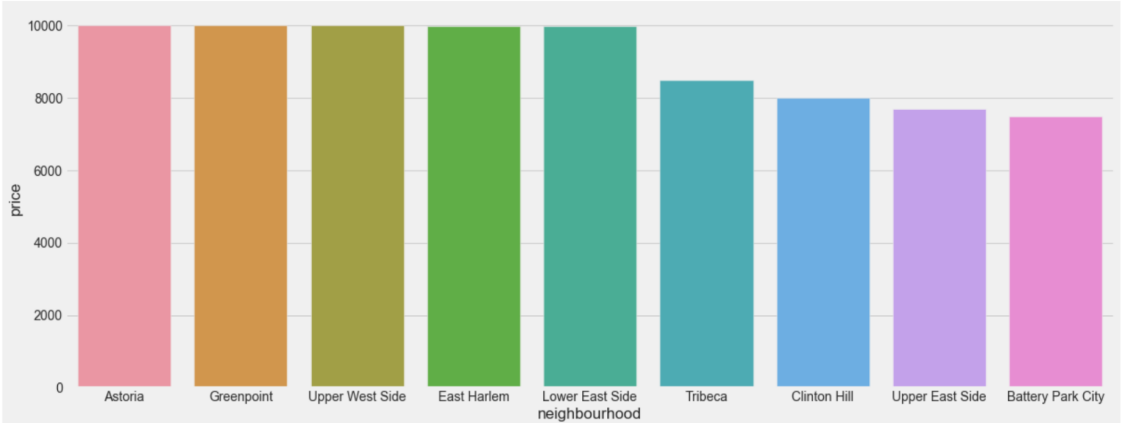
```
In [62]: data = ABdata.neighbourhood.value_counts()[:10]
         plt.figure(figsize=(12, 8))
         x = list(data.index)
         y = list(data.values)
         x.reverse()
         y.reverse()

         plt.title("Most Popular Neighbourhood")
         plt.ylabel("Neighbourhood Area")
         plt.xlabel("Number of guest Who host in this Area")

         plt.barh(x, y)
```

```
Out[62]: <BarContainer object of 10 artists>
```



From this graph we can conclude most of the guests host in Willamsburg followed by Bedford-Stuyvesant and then Harlem.

```
In [22]: plt.figure(figsize =(18,7))
         sns.barplot(x="neighbourhood", y="price", data=ABdata.nlargest(10,['price']))
Out[22]: <AxesSubplot:xlabel='neighbourhood', ylabel='price'>
```



Plot showing top 10 neighbourhoods with highest hotel prices.

CONCLUSION-

We have imported a dataset from a csv document after which all the data of the dataset is collected.then we have shifted the information, cleaned and supplanted every one of the invalid qualities with the proper worth required. The statistical examination of the dataset is finished and afterward through the graphical investigation and from the conclusions derived from it the Airbnb company can use this analysis to see which areas in New York have a large amount of hosted properties and which neighbourhood need more of them. We can see which property has more availability throughout the year, which property do needs more customers and do their promotions accordingly. We could even see which property the hosts prefer to host, which was apartments in this case, and target their audience accordingly as well.

References-

https://www.youtube.com/watch?v=-o3AxdVcUtQ

https://www.simplypsychology.org/boxplots.html

https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data/code?datasetId=268833&searchQuery=EDA