import the required libraries

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

In [2]:
```
#loading the dataset
data = pd.read_csv('/home/tamanna/Downloads/AB_NYC_2019.csv')
```

In [3]:
```
data
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbou
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensi
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Mi
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	H
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clint
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East H
...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Be Stuy
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bus
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	H
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's K
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's K

48895 rows × 16 columns

Exploring the data

In [4]:

```
data.head()
```

Out[4]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	lati
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.6
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.7
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.8
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.6
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.7

In [5]:
```
data.tail()
```

Out[5]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhc
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedfo Stuyves
48891	36485057	Affordable room in Bushwick/ East Williamsburg	6570630	Marisol	Brooklyn	Bushw
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harl
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitcl
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitcl

In [6]:
```
pd.concat([data.head(), data.tail()])
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbou
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensi
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Mi
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	H
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clint
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East H
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Be Stuy
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bus
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	H
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's K
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's K

In [7]:

```
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 [8]:
```
data.shape
```

Out[8]:
```
(48895, 16)
```

In [9]:
```
data.describe()
```

Out[9]:

	id	host_id	latitude	longitude	price	minimu
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	4889
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	125

Handling duplicate values

In [10]:
```
data.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
data.drop_duplicates(inplace = True)
```

Handling Missing Values

```
In [12]: data.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
```

```
In [13]: # First Method imputing mean value
         data['reviews_per_month'].fillna(data['reviews_per_month'].mean(), inplace =
```

```
In [14]: # Second Method dropping na values
         data.dropna(inplace = True)
```

```
In [15]: data.isnull().sum()
```

```
Out[15]: 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
```

Detecting and Removing Outliers

```
In [16]: # z-score and interquartile range both are the criteria to identify an outli
         # ways to find outlier -
         #using scatter plot - distribution of x and y
```

```
#box plot
#using z score
#using iqr range

#Interquantile range steps
#75% - 25%

#steps

#arrange the data in increasing order
#calculate first(q1) and third quatile(q3)
#find interquartile range(q3-q1)
#find lower bound q1*1.5
#find upper bound q3*1.5

#anything outside lower or upper bound is an outlier.
```

In [17]:
```
des_stat = data['price'].describe(percentiles = [.25, .75])
```

In [18]:
```
des_stat
```

Out[18]:
```
count    38821.000000
mean       142.332526
std        196.994756
min          0.000000
25%         69.000000
50%        101.000000
75%        170.000000
max      10000.000000
Name: price, dtype: float64
```

In [19]:
```
# find ist and 3rd quantile
q1, q3 = np.percentile(data['price'], [25, 75])
```

In [20]:
```
print(q1, q3)
```

```
69.0 170.0
```

In [21]:
```
# find IQR
iqr_value = q3 - q1
print(iqr_value)
```

```
101.0
```

In [22]:
```
lower_bound_val = q1 - (1.5 * iqr_value)
upper_bound_val = q3 + (1.5 * iqr_value)
```

In [23]:
```
print(lower_bound_val, upper_bound_val)
```

```
-82.5 321.5
```

In [24]:
```
new_data = data[(data['price'] >= lower_bound_val) & (data['price'] <= upper
```

In [25]:
```
new_data
```

Out[25]:

	id	name	host_id	host_name	neighbourhood_group	neighbourho
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensingt
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtov
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton H
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harle
5	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray H
...	...	...	...	...	...	
48782	36425863	Lovely Privet Bedroom with Privet Restroom	83554966	Rusaa	Manhattan	Upper East Si
48790	36427429	No.2 with queen size bed	257683179	H Ai	Queens	Flushi
48799	36438336	Seas The Moment	211644523	Ben	Staten Island	Great K
48805	36442252	1B-1B apartment near by Metro	273841667	Blaine	Bronx	Mott Hav
48852	36455809	Cozy Private Room in Bushwick, Brooklyn	74162901	Christine	Brooklyn	Bushw

36744 rows × 16 columns

outliers removed

Standardization

In [26]:
```
#Standardization
from sklearn.preprocessing import StandardScaler
```

In [27]:
```
scaling = StandardScaler()
```

In [28]:
```
column_to_standardize = data['reviews_per_month'].values.reshape(-1, 1)
```

In [29]:
```
standradize_values = scaling.fit_transform(column_to_standardize)
```

In [30]:
```
data['reviews_per_month'] = standradize_values.flatten()
```

In [31]:
```
data
```

Out[31]:

	id	name	host_id	host_name	neighbourhood_group	neighbourho
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensingt
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtov
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton H
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harle
5	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray H
...	...	...	...	...	...	
48782	36425863	Lovely Privet Bedroom with Privet Restroom	83554966	Rusaa	Manhattan	Upper East Si
48790	36427429	No.2 with queen size bed	257683179	H Ai	Queens	Flushi
48799	36438336	Seas The Moment	211644523	Ben	Staten Island	Great K
48805	36442252	1B-1B apartment near by Metro	273841667	Blaine	Bronx	Mott Hav
48852	36455809	Cozy Private Room in Bushwick, Brooklyn	74162901	Christine	Brooklyn	Bushw

38821 rows × 16 columns

Now data is cleaned to do further process.