Knowledge Discovery & Data Mining Lab-02

Name: Gurvinder Kaur Matharu
PRN: 20190802077

AIM:

Analyzing statistical description, effects of outliers, missing values and noise in a given dataset.

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data = pd.read_csv('AB_NYC_2019.csv')
```

In [3]:

```
data.head()
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80

In [4]:

```
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]:
```
data.describe() # statistical description and 5-point summary
```

Out[5]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	3
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	

In [6]:
```
data.shape # number of rows and columns in the dataset
```

Out[6]:
```
(48895, 16)
```

In [7]:
```
data.isnull().sum() # no. of missing values present in the dataset
```

Out[7]:
```
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 [8]:
```
df = data
```

Detecting Outliers from the 'price' column of the dataset

Outliers are the extreme values on the low and the high side of the data.
Using the Interquartile Range Method:

In [9]:
```
q1=df['price'].quantile(0.25)
q2=df['price'].quantile(0.5)
q3=df['price'].quantile(0.75)
iqr = q3-q1
```

In [10]:
```
lc = q1 - 1.5*iqr
uc = q3 + 1.5*iqr
```

In [11]:
```
lc
```

Out[11]:
```
-90.0
```

In [12]:
```
uc
```

Out[12]:
```
334.0
```

In [13]:
```python
# identifying outliers
df[df['price']>uc]
```

Out[13]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitu
61	15396	Sunny & Spacious Chelsea Apartment	60278	Petra	Manhattan	Chelsea	40.746
85	19601	perfect for a family or small group	74303	Maggie	Brooklyn	Brooklyn Heights	40.697
103	23686	2000 SF 3br 2bath West Village private townhouse	93790	Ann	Manhattan	West Village	40.730
114	26933	2 BR / 2 Bath Duplex Apt with patio! East Village	72062	Bruce	Manhattan	East Village	40.725
121	27659	3 Story Town House in Park Slope	119588	Vero	Brooklyn	South Slope	40.664
...	...	...	...	...	...	...	
48758	36420289	Rustic Garden House Apt, 2 stops from Manhattan	73211393	LaGabrell	Queens	Long Island City	40.755
48833	36450896	Brand New 3-Bed Apt in the Best Location of FiDi	29741813	Yue	Manhattan	Financial District	40.706
48839	36452721	Massage Spa. Stay overnight. Authors Artist dr...	274079964	Richard	Brooklyn	Sheepshead Bay	40.598
48842	36453160	LUXURY MANHATTAN PENTHOUSE+HUDSON RIVER+EMPIRE...	224171371	LuxuryApartmentsByAmber	Manhattan	Chelsea	40.752
48856	36457700	Large 3 bed, 2 bath , garden , bbq , all you need	66993395	Thomas	Brooklyn	Bedford-Stuyvesant	40.688

2972 rows × 16 columns

In [14]:
```python
df[(df['price']<uc) & (df['price']>lc)]
```

Out[14]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	ro
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	

Using the Standard Deviation Method:

In [15]:

```
mean = np.mean(data['price'])
std = np.std(data['price'])
print('Mean:', mean)
print('Standard Deviation:', std)
x = std*3
lower = mean-x
upper = mean+x
outliers = [i for i in data['price'] if i < lower or i > upper]
print('No. of outliers found in the price column of the dataset: ',len(outliers))
```

```
Mean: 152.7206871868289
Standard Deviation: 240.1517139194169
No. of outliers found in the price column of the dataset:  388
```

In [16]:

```
data[data['price']>upper]
```

Out[16]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	roo
496	174966	Luxury 2Bed/2.5Bath Central Park View	836168	Henry	Manhattan	Upper West Side	40.77350	-73.98697	h
762	273190	6 Bedroom Landmark West Village Townhouse	605463	West Village	Manhattan	West Village	40.73301	-74.00268	h
946	363673	Beautiful 3 bedroom in Manhattan	256239	Tracey	Manhattan	Upper West Side	40.80142	-73.96931	
1105	468613	$ (Phone number hidden by Airbnb) weeks - room f	2325861	Cynthia	Manhattan	Lower East Side	40.72152	-73.99279	
1414	634353	Luxury 1Bed with Central Park Views	836168	Henry	Manhattan	Upper West Side	40.77428	-73.98594	h
...	...	...	...	...	...	...	...	...	
48301	36186719	Private Bedroom in the Heart of Chelsea!	268920555	Terrence Jake	Manhattan	Chelsea	40.74531	-73.99454	
48304	36189195	Next to Times Square/Javits/MSG! Amazing 1BR!	270214015	Rogelio	Manhattan	Hell's Kitchen	40.75533	-73.99866	h
48305	36189257	2BR Near Museum Mile! Upper East Side!	272166348	Mary Rotsen	Manhattan	Upper East Side	40.78132	-73.95262	h
48523	36308562	Tasteful & Trendy Brooklyn Brownstone, near Train	217732163	Sandy	Brooklyn	Bedford-Stuyvesant	40.68767	-73.95805	h
48535	36311055	Stunning & Stylish Brooklyn Luxury, near Train	245712163	Urvashi	Brooklyn	Bedford-Stuyvesant	40.68245	-73.93417	h

388 rows × 16 columns

```
In [17]: data[(data['price']<upper) & (data['price']>lower)]  # without outliers
```

Out[17]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Pr
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	E hom
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Pr
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	E hom
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	E hom
...	...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Pr
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Pr
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	E hom
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Sh
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Pr

48507 rows × 16 columns

CONCLUSION:

The outliers were identified from the given dataset by using the quantile method and the standard deviation method. Missing values were also identified from the datastet. The statistical description of the dataset was also looked upon.