METHODOLOGY DOCUMENT

1. Problem Statement

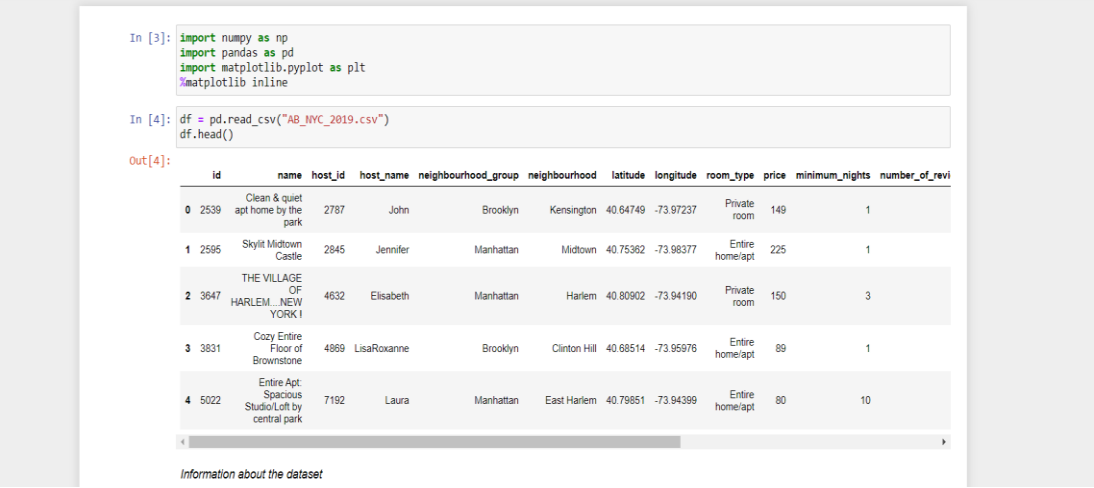
Suppose that you are working as a data analyst at Airbnb. For the past few months, Airbnb has seen a major decline in revenue. Now that the restrictions have started lifting and people have started to travel more, Airbnb wants to make sure that it is fully prepared for this change.

The different leaders at Airbnb want to understand some important insights based on various attributes in the dataset so as to increase the revenue such as -

1. Which type of hosts to acquire more and where?
2. The categorisation of customers based on their preferences.
    - What are the neighbourhoods they need to target?
    - What is the pricing ranges preferred by customers?
    - The various kinds of properties that exist w.r.t. customer preferences.
    - Adjustments in the existing properties to make it more customer-oriented.
3. What are the most popular localities and properties in New York currently?
4. How to get unpopular properties more traction? and so on...

Important Steps done to achieve the outcome in the assignment

- Importing the necessary libraries and the dataset that is provided.



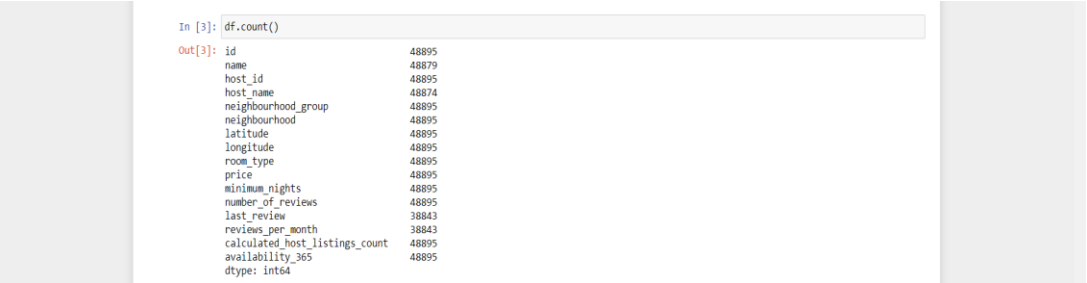
```
In [3]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

In [4]: df = pd.read_csv("AB_NYC_2019.csv")
        df.head()
```

Out[4]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_revi
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

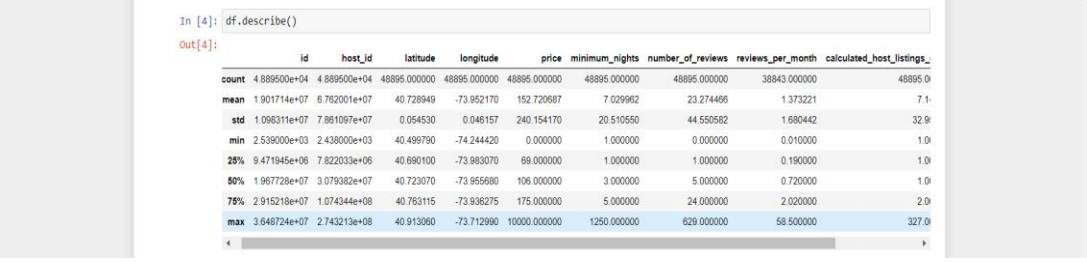
Information about the dataset

- Checking the count of the data that is present in each column



```
In [3]: df.count()

Out[3]: id                                48895
        name                              48879
        host_id                           48895
        host_name                         48874
        neighbourhood_group               48895
        neighbourhood                     48895
        latitude                          48895
        longitude                         48895
        room_type                         48895
        price                             48895
        minimum_nights                    48895
        number_of_reviews                 48895
        last_review                       38843
        reviews_per_month                 38843
        calculated_host_listings_count    48895
        availability_365                  48895
        dtype: int64
```

- Checking the describe of the data



```
In [4]: df.describe()
```

Out[4]:

	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

- Cleaning of data

Cleaning of Data

```
In [9]: df.dropna(subset=['last_review','reviews_per_month'],axis=0,inplace=True)
        df.count()
```

```
Out[9]: id                                38843
        name                              38837
        host_id                           38843
        host_name                         38827
        neighbourhood_group               38843
        neighbourhood                     38843
        latitude                          38843
        longitude                         38843
        room_type                         38843
        price                             38843
        minimum_nights                    38843
        number_of_reviews                 38843
        last_review                       38843
        reviews_per_month                 38843
        calculated_host_listings_count    38843
        availability_365                  38843
        dtype: int64
```

- Segmentation of data for visualization of type of rooms

Segmentation of data for visualiztion of type of rooms

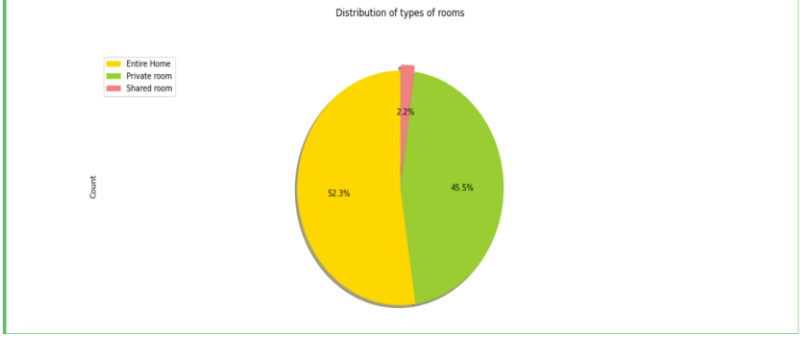
```
In [8]: df_type_of_rooms = df['room_type'].value_counts()
        df_type_of_rooms = pd.DataFrame({'Room Type': df_type_of_rooms.index,'Count': df_type_of_rooms.values})
        df_type_of_rooms
```

Out[8]:

	Room Type	Count
0	Entire home/apt	20332
1	Private room	17665
2	Shared room	846

- Pie Chart Visualization of type of rooms

```
In [11]: df_type_of_rooms['Count'].plot(kind='pie', figsize=(15,6), autopct="%1.1f%%", startangle=90, shadow=True, labels=None,
                                        colors=['gold','yellowgreen','lightcoral'], explode=[0,0,0.05])
         plt.title('Distribution of types of rooms',y=1.12)
         plt.axis('equal')
         plt.legend(labels=['Entire Home','Private room','Shared room'],loc='upper left')
         plt.show()
```



- Checking the count based on neighborhood_group

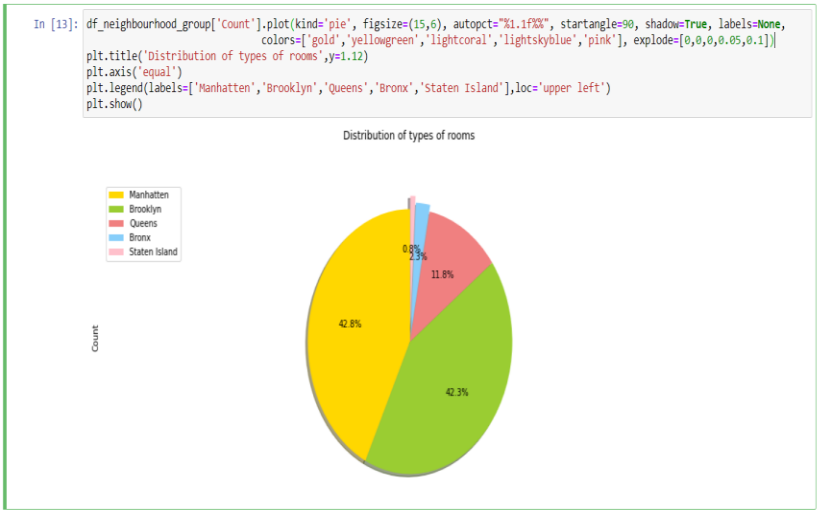
```
In [10]: df_neighbourhood_group = df['neighbourhood_group'].value_counts()
         df_neighbourhood_group = pd.DataFrame({'Neighbourhood': df_neighbourhood_group.index,'Count': df_neighbourhood_group.values})
         df_neighbourhood_group
```

Out[10]:

	Neighbourhood	Count
0	Manhattan	16632
1	Brooklyn	16447
2	Queens	4574
3	Bronx	876
4	Staten Island	314

- Pie chart visualization on distribution of types of rooms based on neighborhood_group

```
In [13]: df_neighbourhood_group['Count'].plot(kind='pie', figsize=(15,6), autopct="%1.1f%%", startangle=90, shadow=True, labels=None,
                                              colors=['gold','yellowgreen','lightcoral','lightskyblue','pink'], explode=[0,0,0,0.05,0.1])
         plt.title('Distribution of types of rooms',y=1.12)
         plt.axis('equal')
         plt.legend(labels=['Manhatten','Brooklyn','Queens','Bronx','Staten Island'],loc='upper left')
         plt.show()
```



- Checking different prices in different neighborhood_group

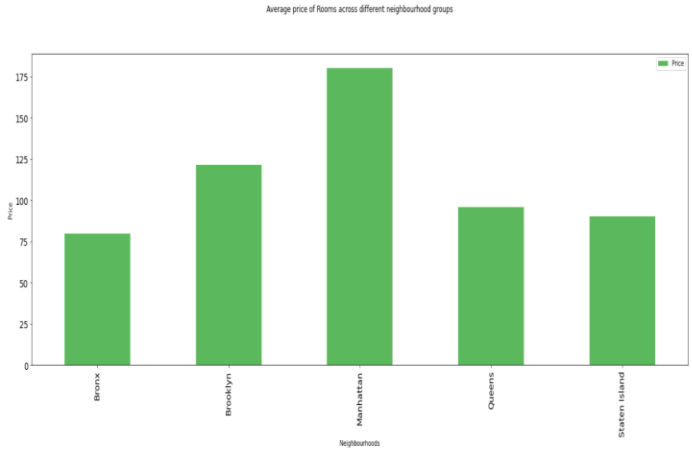
```
In [12]: df_neighbourhood_group_price = df.groupby(['neighbourhood_group']).mean()
         df_neighbourhood_group_price = df_neighbourhood_group_price['price']
         df_neighbourhood_group_price = pd.DataFrame({'Neighbourhood': df_neighbourhood_group_price.index,'Price': df_neighbourhood_group_
         df_neighbourhood_group_price.set_index('Neighbourhood')
```

Out[12]:

Neighbourhood	Price
Bronx	79.553653
Brooklyn	121.448714
Manhattan	180.052489
Queens	95.762571
Staten Island	89.964968

- Average Price of Rooms across different neighborhood_group

```
In [33]: df_neighbourhood_group_price.plot(kind='bar', figsize=(24,8), width=0.5, fontsize=14, color = ['#5cb85c'])
         plt.title('Average price of Rooms across different neighbourhood groups',y=1.12)
         plt.xticks(df_neighbourhood_group_price.index,df_neighbourhood_group_price.Neighbourhood)
         plt.xlabel('Neighbourhoods')
         plt.ylabel('Price')
         plt.show()
```



- Number of each type of rooms in each Neighborhood_group

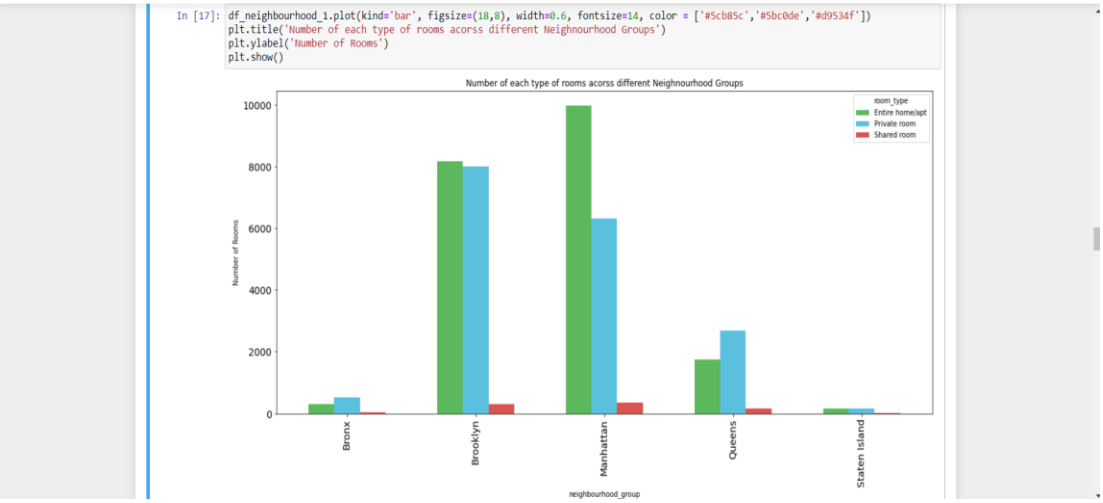
Number of each type of rooms in each Neighbourhood group

```
In [14]: df_neighbourhood_1 = df.groupby(['neighbourhood_group','room_type'])
         df_neighbourhood_1 = df_neighbourhood_1['id'].count()
         df_neighbourhood_1 = df_neighbourhood_1.unstack(level=1)
         df_neighbourhood_1
```

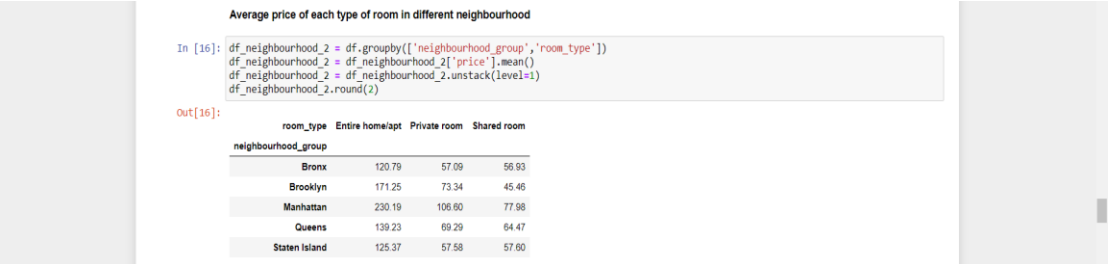
Out[14]:

room_type	Entire home/apt	Private room	Shared room
neighbourhood_group			
Bronx	309	524	43
Brooklyn	8164	7993	290
Manhattan	9967	6309	356
Queens	1742	2680	152
Staten Island	150	159	5

- Number of each type of rooms across different Neighborhood_group



- Average price of each type of room in different neighborhood



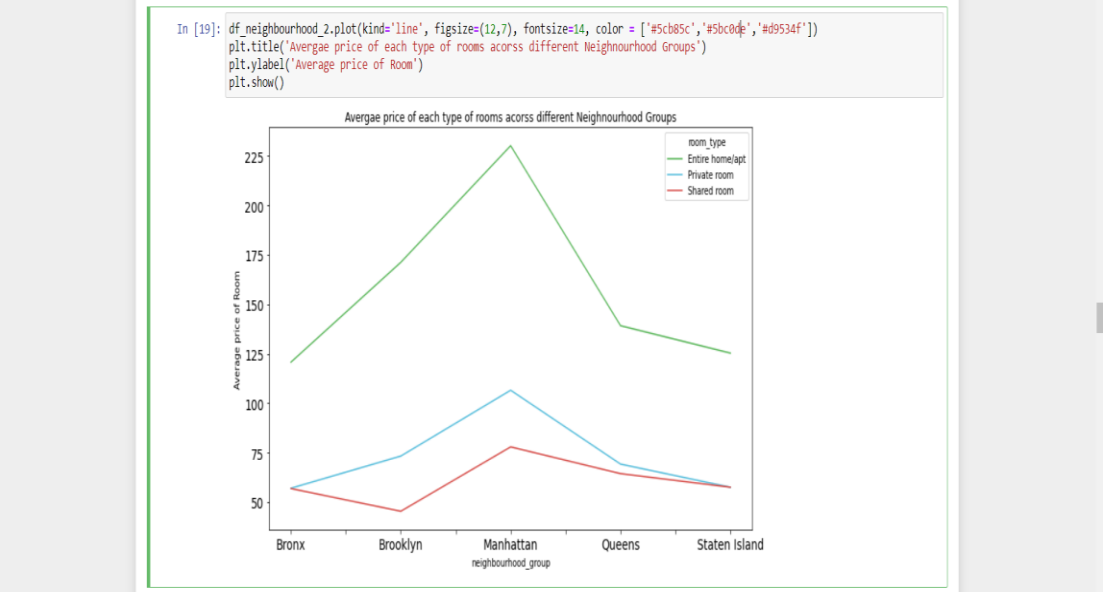
Average price of each type of room in different neighbourhood

```
In [16]: df_neighbourhood_2 = df.groupby(['neighbourhood_group','room_type'])
df_neighbourhood_2 = df_neighbourhood_2['price'].mean()
df_neighbourhood_2 = df_neighbourhood_2.unstack(level=1)
df_neighbourhood_2.round(2)
```

Out[16]:

room_type	Entire home/apt	Private room	Shared room
neighbourhood_group			
Bronx	120.79	57.09	56.93
Brooklyn	171.25	73.34	45.46
Manhattan	230.19	106.60	77.98
Queens	139.23	69.29	64.47
Staten Island	125.37	57.58	57.60

- Average price of each type of rooms across different Neighborhood_group



- Maximum price of each type of room across different neighborhoods

Maximum price of each type of room across different neighbourhoods

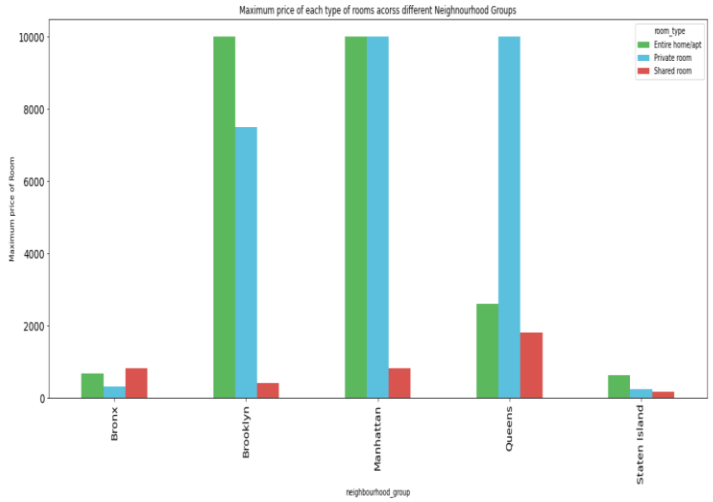
```
In [18]: df_neighbourhood_3 = df.groupby(['neighbourhood_group','room_type'])
         df_neighbourhood_3 = df_neighbourhood_3['price'].max()
         df_neighbourhood_3 = df_neighbourhood_3.unstack(level=1)
         df_neighbourhood_3
```

Out[18]:

neighbourhood_group	Entire home/apt	Private room	Shared room
Bronx	670	300	800
Brooklyn	10000	7500	400
Manhattan	9999	9999	800
Queens	2600	10000	1800
Staten Island	625	235	150

- Maximum price of each type of rooms across different Neighborhood_group

```
In [21]: df_neighbourhood_3.plot(kind='bar', figsize=(20,8), fontsize=14, color = ['#5cb85c','#5bc0de','#d9534f'])
         plt.title('Maximum price of each type of rooms acorss different Neighnourhood Groups')
         plt.ylabel('Maximum price of Room')
         plt.show()
```



- Checking for the neighborhood groups that are available and which shows up the most

```
In [22]: # Checking for the neighborhood groups that are available and which shows up the most

         print(df.neighbourhood_group.unique())
         print('\nNeighborhood Group Count:',len(df.neighbourhood_group.unique()))

         # There are five neighborhood groups available

         print('\n',df.groupby('neighbourhood_group')['id'].count())

         # Manhattan shows up the most, with 16632
```

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

Neighborhood Group Count: 5

 neighbourhood_group
Bronx             876
Brooklyn        16447
Manhattan       16632
Queens           4574
Staten Island     314
Name: id, dtype: int64
```

- Checking which hosts are the busiest based on their reviews

```
In [23]: # Checking which hosts are the busiest based on their reviews

air_bnb_busyhost= df.groupby(['host_id', 'host_name', 'calculated_host_listings_count'], as_index=False)['reviews_per_month'].sum
air_bnb_busyhost.sort_values('reviews_per_month', ascending=False).head()

# If we analyse data based on a monthly number of reviews between all listings of a certain host, Sonder has a
# total of 397.56 reviews per month followed by Row with 111.72
```

Out[23]:

	host_id	host_name	calculated_host_listings_count	reviews_per_month
28206	219517861	Sonder (NYC)	327	397.56
29160	244361589	Row NYC	9	111.72
28842	232251881	Lakshmee	8	80.63
15362	26432133	Danielle	5	68.02
25201	137274917	David	12	62.89

- Checking which are the top 5 hosts have the highest total price

```
In [24]: # Checking which are the top 5 hosts have the highest total price

df_total_price = df.groupby(['host_id', 'host_name', 'calculated_host_listings_count'], as_index=False)['price'].sum()
df_total_price.sort_values('price', ascending=False).head()

# Based on the total price/night between all the listings a single host has data shows that Sonder is leading with
# a total of $55920 followed by Blueground, Sally, Red Awning, and Kara.
```

Out[24]:

	host_id	host_name	calculated_host_listings_count	price
28206	219517861	Sonder (NYC)	327	55920
27692	205031545	Red Awning	49	24194
1724	836168	Henry	11	19500
2122	1177497	Jessica	11	14850
12011	16098958	Jeremy & Laura	96	12080

- Checking who currently are having no (zero) availability with a review count of 100 or more

```
In [25]: # Checking who currently are having no (zero) availability with a review count of 100 or more
# pd.set_option('display.max_rows',None)

zero_availability = df.loc[(df.availability_365==0) & (df.number_of_reviews>=100)]
zero_availability.count()
zero_availability.get(key=['host_name', "availability_365", 'number_of_reviews'])
# There are 162 entries that have zero availability and more than 100 reviews.
```

Out[25]:

	host_name	availability_365	number_of_reviews
8	MaryEllen	0	118
94	Christiana	0	168
132	Sol	0	193
174	Coral	0	114
180	Doug	0	206
...	...	...	...
29581	Kathleen	0	103
30461	Janet	0	119
31250	Albert	0	102
32670	Stephany	0	131
35014	Mariluz	0	112

162 rows × 3 columns

- Checking which host has the highest total of prices and where are they located

```
In [26]: # Checking which host has the highest total of prices and where are they located

highest_total_prices = df.groupby(['host_id', 'host_name', 'neighbourhood_group'], as_index=False)['price'].sum()
highest_total_prices.sort_values('price', ascending=False).head(1)

# Sonder has the highest total of prices and is located in Manhattan
```

Out[26]:

	host_id	host_name	neighbourhood_group	price
28280	219517861	Sonder (NYC)	Manhattan	55920

- Checking when did Danielle from Queens last receive a review

```
In [27]: # Checking when did Danielle from Queens last receive a review

hey_danielle = df.loc[(df.host_name=='Danielle') & (df.neighbourhood_group=='Queens')]
hey_danielle.sort_values(by=['last_review', 'host_id'], ascending=False).groupby('host_id').head()

# There are 4 different Danielles in Queens. However the latest review for a Danielle in our database was for host_id 26432133
# and it was posted on 2019-07-08. Danielle from Queend/Long Island City got a review on 2019-06-20,
# Danielle from Queens/Astoria got her latest review on 2018-01-02 and the last Danielle in our database
# did not get any reviews.
```

Out[27]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_
33861	26814763	One bedroom with full bed / 1 stop from Manhattan	201647469	Danielle	Queens	Long Island City	40.74565	-73.94699	Private room	108	2	
22469	18173787	Cute Tiny Room Family Home by LGA NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76380	-73.87238	Private room	48	1	
21517	17222454	Sun Room Family Home LGA Airport NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76367	-73.87088	Private room	48	1	
20403	16276632	Cozy Room Family Home LGA Airport NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76335	-73.87007	Private room	48	1	
		Bed in Family Home							Shared			

33861	26814763	with full bed / 1 stop from Manhattan	201647469	Danielle	Queens	Long Island City	40.74565	-73.94699	room	108	2	
22469	18173787	Cute Tiny Room Family Home by LGA NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76380	-73.87238	Private room	48	1	
21517	17222454	Sun Room Family Home LGA Airport NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76367	-73.87088	Private room	48	1	
20403	16276632	Cozy Room Family Home LGA Airport NO CLEANING FEE	26432133	Danielle	Queens	East Elmhurst	40.76335	-73.87007	Private room	48	1	
22068	17754072	Bed in Family Home Near LGA Airport	26432133	Danielle	Queens	East Elmhurst	40.76389	-73.87155	Shared room	38	1	
7086	5115372	Comfy Room Family Home LGA Airport NO CLEANING...	26432133	Danielle	Queens	East Elmhurst	40.76374	-73.87103	Private room	54	1	
27021	21386105	Quiet & clean 1br haven with balcony near the ...	154256662	Danielle	Queens	Astoria	40.77134	-73.92424	Entire home/apt	250	3	

- Checking which hosts had the most listings

```
In [28]: # Checking which hosts had the most listings

         highest_total_prices = df.groupby(['host_id', 'host_name'], as_index=False)['calculated_host_listings_count'].count()
         highest_total_prices.sort_values('calculated_host_listings_count', ascending=False).head(2)

         # Sonder has a calculated host listings count of 207
```

Out[28]:

	host_id	host_name	calculated_host_listings_count
28206	219517861	Sonder (NYC)	207
20808	61391963	Corporate Housing	79

- Checking for how many listings have completely "open availability"

```
In [29]: # Checking for how many listings have completely "open availability"

         complete_availability= df.loc[(df.availability_365==365)]
         complete_availability.count()

         # there are 841 entries that have a 365 days availability
```

Out[29]:

```
id                                841
name                              841
host_id                           841
host_name                         841
neighbourhood_group               841
neighbourhood                     841
latitude                          841
longitude                         841
room_type                         841
price                             841
minimum_nights                    841
number_of_reviews                 841
last_review                       841
reviews_per_month                 841
calculated_host_listings_count    841
availability_365                  841
dtype: int64
```

- Checking which room_types have the highest review numbers

```
In [30]: # Checking which room_types have the highest review numbers

         reviews_by_roomtype = df.groupby(['room_type'], as_index=False)['number_of_reviews'].sum()
         reviews_by_roomtype.sort_values('number_of_reviews', ascending=False)
         # The Entire home/apt has the highest review numbers and is followed by the private room type.
```

Out[30]:

	room_type	number_of_reviews
0	Entire home/apt	580403
1	Private room	538346
2	Shared room	19256

Conclusion

- There are 5 different neighbourhood groups present in our database: 1.Manhattan 2.Bronx 3.Brooklyn 4.Queens 5.Staten Island

- The one that shows up the most is Manhattan with over 16 thousand entries.

- There are three room types listed: 1.Private room 2.Shared room 3. Entire Home/apt.

- We plotted a pie chart for visualising different types of rooms and we found out that most of the people offered either an Entire home/apt or a private room. We plotted a pie chart for distribution of different types of rooms and we found out that more than half of the houses are based in either Manhatten or Brooklyn.

- Then checked the average price of Rooms across different neighbourhood groups and found out that Manhattan has the highest average price of 175$. Then we found out the maximum price of each type of rooms acorss different Neighnourhood Groups and Manhattan leads with room_type Entire home/apt and private room.

- Then we checked for the neighborhood groups that are available and which shows up the most and found out that , there were 5 neighborhood groups available and Manhattan shows up the most, with 16632. If we analyze the same data based on the average number of reviews per month between all listings of a certain host, Sonder has a total of 397.56 reviews per month followed by Row NYC with 111.72.

- Looking at the price by neighbourhood group: The highest average price for a room in NYC is in Manhattan with a little over 196 $/night. The highest total price for listings in NYC are in Manhattan almost double the total price in Brooklyn.

- Based on the total price of all the listings we have a TOP 5 hosts in NYC: Sonder is leading with a total of $55920 followed by Blueground, Sally, Red Awning, and Kara.

- We analyzed the entries with zero available days in 2019 and more than 100 reviews for the year and found out that there are 162 entries that satisfy these conditions.

- By summing the total prices per host we found that "Sonder NYC" has the highest total of prices (55920) and is located in Manhattan. Also, Sonder is the leader in the total calculated listings count with 207 entries.

- We looked into a specific case of Danielle from Queens and her listings sorted by the date of the last review. We found that there are 4 different Danielles in Queens. However, the latest review for a Danielle in our database was for Danielle with the host_id 26432133 and it was posted on 2019-07-08. Danielle from Queend/Long Island City got her latest review on 2019-06-20, Danielle from Queens/Astoria got her latest review on 2018-01-02 and the last Danielle in our database did not get any reviews.

- Finally, we looked into how many entries had a year round availability and found that 841 properties were available for 365 days in 2019, also we found that by the total number of reviews left on the platform, the "Entire home/apt" option is leading by far in this category.