AIRBNB NYC, CASE STUDY METHODOLOGY:

Prepared by:

SreeLakshmi Vaddi

Maria Peter

Krishna S

Tools used:

Google Colab

Tableau

MS PPT

Problem background:

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.

Dataset used for methodology – AB_NYC_2019

Importing necessary libraries and data reading:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

```
airbnb = pd.read_csv("AB_NYC_2019.csv")
airbnb.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_mont
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	19-10-2018	0.2
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	21-05-2019	0.3
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	NaN	Na
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	05-07-2019	4.6
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	19-11-2018	0.1

Information on datatypes:

```
airbnb.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
```

There are no null values in any of the columns.

Creating features to categorize the columns:

```
def availability_365_categories_function(row):
    Categorizes the "minimum_nights" column into 5 categories
    if row <= 1:
        return 'very Low'
    elif row <= 100:
        return 'Low'
    elif row <= 200 :
        return 'Medium'
    elif (row <= 300):
        return 'High'
    else:
        return 'very High'
```

```
def minimum_night_categories_function(row):
    Categorizes the "minimum_nights" column into 5 categories
    if row <= 1:
        return 'very Low'
    elif row <= 3:
        return 'Low'
    elif row <= 5 :
        return 'Medium'
    elif (row <= 7):
        return 'High'
    else:
        return 'very High'
```

```
def number_of_reviews_categories_function(row):
    Categorizes the "number_of_reviews" column into 5 categories
    if row <= 1:
        return 'very Low'
    elif row <= 5:
        return 'Low'
    elif row <= 10 :
        return 'Medium'
    elif (row <= 30):
        return 'High'
    else:
        return 'very High'
```

```
def price_categories_function(row):
    Categorizes the "number_of_reviews" column into 5 categories
    if row <= 1:
        return 'very Low'
    elif row <= 4:
        return 'Low'
    elif row <= 15 :
        return 'Medium'
    elif (row <= 100):
        return 'High'
    else:
        return 'very High'
```

Check for missing values in the columns:

```
airbnb.isnull().sum()
```

```
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
availability_365_categories         0
minimum_night_categories            0
number_of_reviews_categories        0
price_categories                    0
dtype: int64
```

The columns last_review & reviews_per_month has 20.5% missing values, columns name and host_name has 0.3% and 0.4% missing values respectively.

Missing value treatment:

```
## Dropping columns that are insignificant for analysis
airbnb.drop(['last_review'], axis=1, inplace=True)
```

```
## Replacing null values in reviews_per_month with 0.0
airbnb.fillna({'reviews_per_month':0.0}, inplace=True)
```

```
## Replacing null values in name as unknown
airbnb['name'].fillna('unknown',inplace=True)
```

```
## Check for duplicate data
airbnb_dup = airbnb.duplicated()
print(airbnb_dup.sum())
airbnb[airbnb_dup]
```

There are no duplicate values.

Univariate Analysis:

```
## Top 5 listings on Airbnb in NYC
listing_count=airbnb['name'].value_counts()[:5].reset_index()
listing_count.rename(columns={'index':'Listing on Airbnb','name':'Total_listing'},inplace=True)
listing_count
```

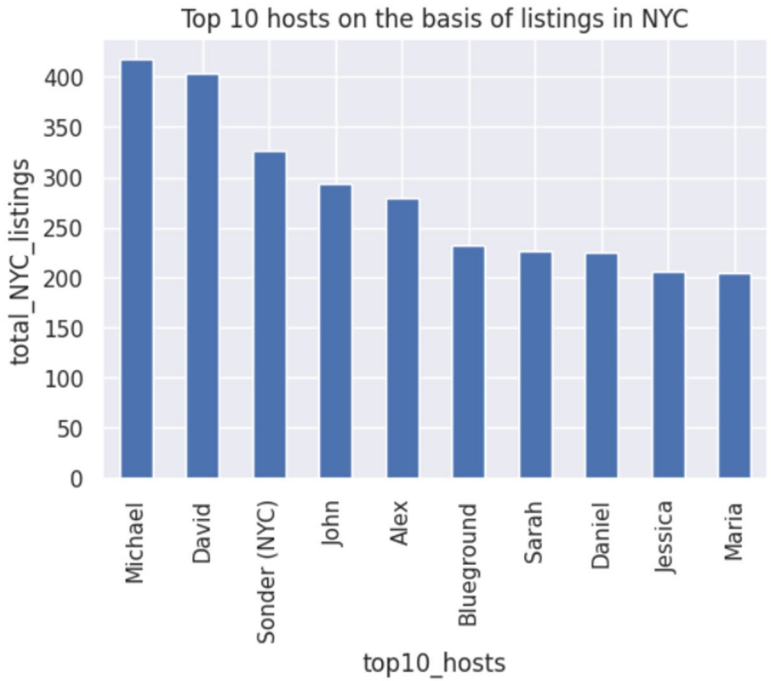
	Listing on Airbnb	Total_listing
0	Hillside Hotel	18
1	Home away from home	17
2	unknown	16
3	New york Multi-unit building	16
4	Brooklyn Apartment	12

```
## Top 10 hosts on the basis of no of listing in NYC
top_10_hosts=airbnb['host_name'].value_counts()[:10]
top_10_hosts
```

```
Michael          417
David            403
Sonder (NYC)     327
John             294
Alex             279
Blueground       232
Sarah            227
Daniel           226
Jessica          205
Maria            204
Name: host_name, dtype: int64
```

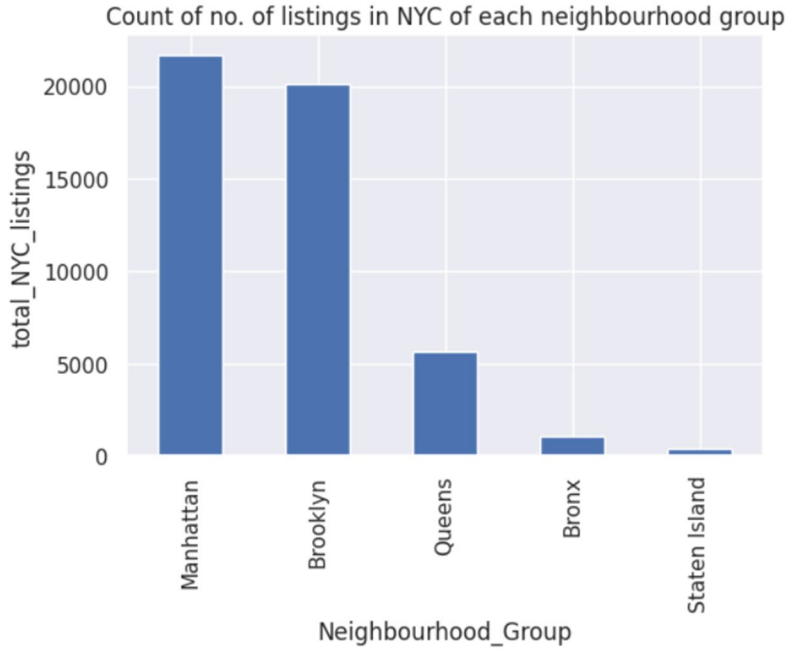
```
top_10_hosts.plot(kind='bar',color='b')
plt.xlabel('top10_hosts')
plt.ylabel('total_NYC_listings')
plt.title('Top 10 hosts on the basis of listings in NYC')
```

```
Text(0.5, 1.0, 'Top 10 hosts on the basis of listings in NYC')
```



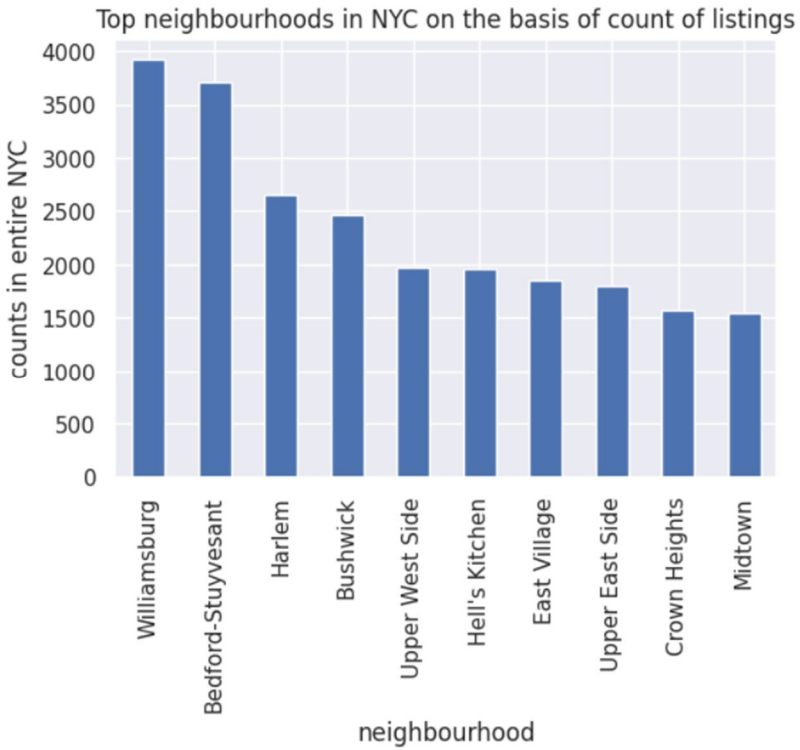
```
airbnb['neighbourhood_group'].value_counts().plot(kind='bar',color='b')
plt.xlabel('Neighbourhood_Group')
plt.ylabel('total_NYC_listings')
plt.title('Count of no. of listings in NYC of each neighbourhood group')
```

```
Text(0.5, 1.0, 'Count of no. of listings in NYC of each neighbourhood group')
```



'Manhattan' has the highest no of listings in entire NYC.

```
## Check for top 10 neighbourhoods on the basis of no of listings in NYC
top_10_neigbours= airbnb['neighbourhood'].value_counts()[:10]
plt.figure(figsize=(6,4))
top_10_neigbours.plot(kind='bar',color='b')
plt.xlabel('neighbourhood')
plt.ylabel('counts in entire NYC')
plt.title('Top neighbourhoods in NYC on the basis of count of listings')
```

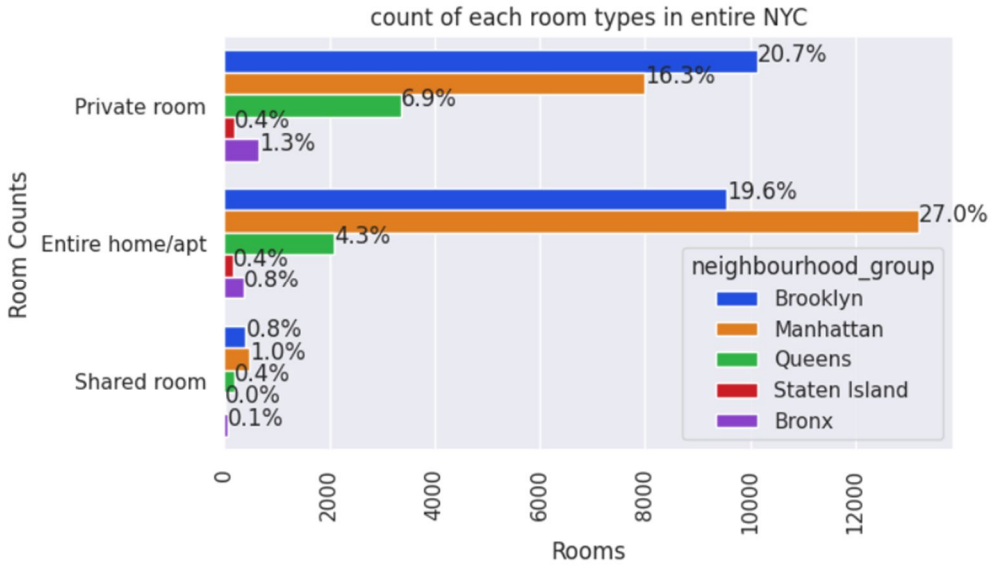

Top neighbourhoods in NYC on the basis of count of listings
counts in entire NYC
4000
3500
3000
2500
2000
1500
1000
500
0
Williamsburg
Bedford-Stuyvesant
Harlem
Bushwick
Upper West Side
Hell's Kitchen
East Village
Upper East Side
Crown Heights
Midtown
neighbourhood

```
plt.rcParams['figure.figsize'] = (7,4)
ax= sns.countplot(y='room_type',hue='neighbourhood_group',data=airbnb,palette='bright')

total = len(airbnb['room_type'])
for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_width()/total)
        x = p.get_x() + p.get_width() + 0.01
        y = p.get_y() + p.get_height()/2
        ax.annotate(percentage, (x, y))

plt.title('count of each room types in entire NYC')
plt.xlabel('Rooms')
plt.xticks(rotation=90)
plt.ylabel('Room Counts')

plt.show()
```

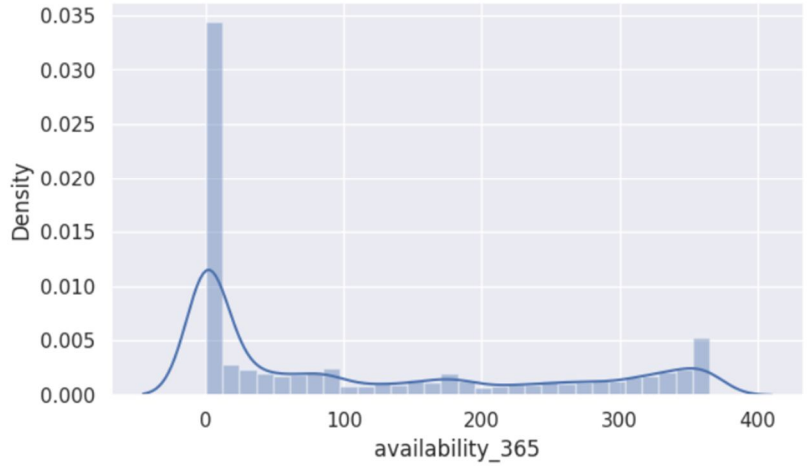


Manhattan has more listed properties with entire home/apt with 27% of total listed properties followed by Brooklyn with 19.6%

Private rooms are more in Brooklyn with 20.7% of the total listed properties followed by Manhattan with 16.3%

```
## Distribution of listings availability at NYC
sns.distplot(airbnb['availability_365'])
```

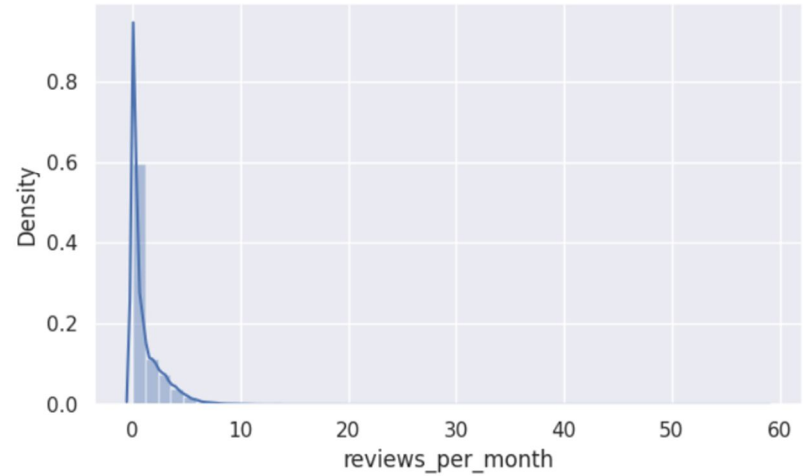
```
<Axes: xlabel='availability_365', ylabel='Density'>
```



It can be inferred that availability_365 is uniformly distributed with availability ranging from 0 to 370.

```
## Look at the distribution of reviews_per_month
fig, ax = plt.subplots(figsize=(7,4))
sns.distplot(airbnb['reviews_per_month'])
```

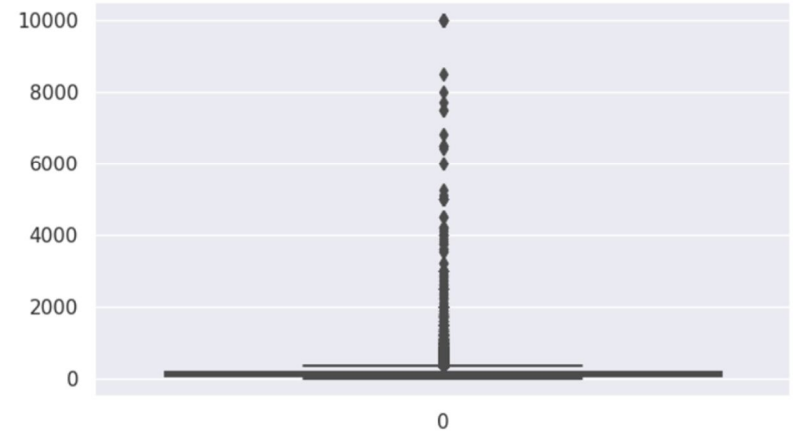
```
<Axes: xlabel='reviews_per_month', ylabel='Density'>
```



Handling the outliers:

```
## Check outliers in price
sns.boxplot(airbnb['price'])
```

```
<Axes: >
```




```
## Outliers treatment
min_threshold,max_threshold= airbnb.price.quantile([0.01,0.999])
min_threshold,max_threshold
```

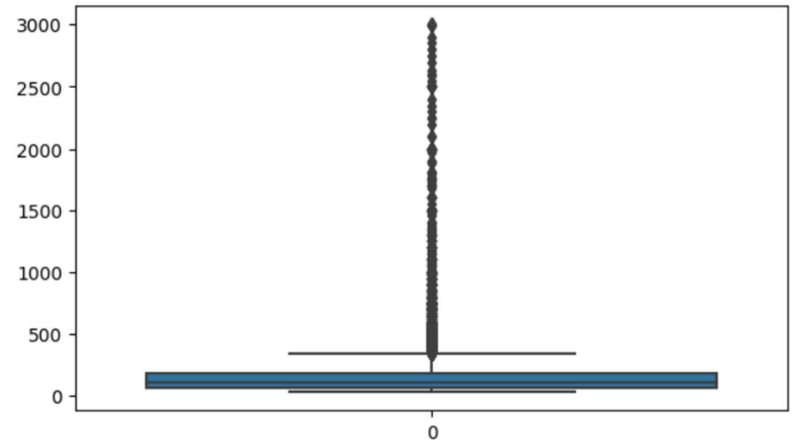
```
(30.0, 3000.0)
```

```
## Treatment of prices less than min threshold
airbnb[airbnb.price<min_threshold]
```

```
## Treatment of prices less than max threshold
airbnb[airbnb.price>max_threshold]
```

```
## Check for outliers again
sns.boxplot(airbnb_newp['price'])
```

```
<Axes: >
```



The outliers are still observed.

Outlier treatment using IQR:

```
## Outliers treatment using IQR
Q1 = airbnb.quantile(0.25)
Q3 = airbnb.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
id                                1.968023e+07
host_id                           9.961239e+07
latitude                          7.301500e-02
longitude                         4.679500e-02
price                             1.060000e+02
minimum_nights                    4.000000e+00
number_of_reviews                 2.300000e+01
reviews_per_month                 1.540000e+00
calculated_host_listings_count    1.000000e+00
availability_365                  2.270000e+02
dtype: float64
```

```
## Removal of outliers
def outlier_treatment(datacolumn):
 sorted(datacolumn)
 Q1,Q3 = np.percentile(datacolumn , [25,75])
 IQR = Q3 - Q1
 lower_range = Q1 - (1.5 * IQR)
 upper_range = Q3 + (1.5 * IQR)
 return lower_range,upper_range
```

```
lower_bound,upper_bound = outlier_treatment(airbnb['price'])
```

```
## Lower and upper range outliers
airbnb[(airbnb.price < lower_bound) | (airbnb.price > upper_bound)]
```

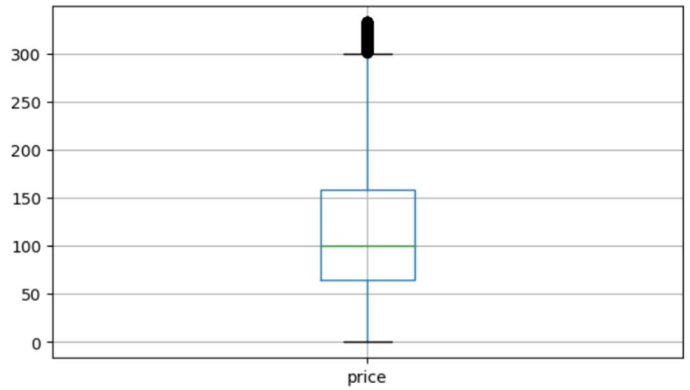
	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
61	15396	Sunny & Spacious Chelsea Apartment	60278	Petra	Manhattan	Chelsea	40.74623	-73.99530	Entire home/apt	375	180	5
85	19601	perfect for a family or small group	74303	Maggie	Brooklyn	Brooklyn Heights	40.69723	-73.99268	Entire home/apt	800	1	25
103	23686	2000 SF 3br 2bath West Village private townhouse	93790	Ann	Manhattan	West Village	40.73096	-74.00319	Entire home/apt	500	4	46
114	26933	2 BR / 2 Bath Duplex Apt with patio! East Village	72062	Bruce	Manhattan	East Village	40.72540	-73.98157	Entire home/apt	350	2	7
121	27659	3 Story Town House in Park Slope	119588	Vero	Brooklyn	South Slope	40.66499	-73.97925	Entire home/apt	400	2	16
...	...	...	...	...	...	...	...	...	...	...	...	...

```
## subsetting those rows having data points greater than lower range and lesser than upper range respectively
airbnb_newp1 = airbnb[(airbnb.price>lower_bound) & (airbnb.price<upper_bound)]
airbnb_newp1
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_month
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	0.21
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	0.38
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	0.00
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	4.64
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	0.10
...	...	...	...	...	...	...	...	...	...	...	...	...	...

```
## Check for outliers again
airbnb_newp1.boxplot(column='price')
```

```
<Axes: >
```

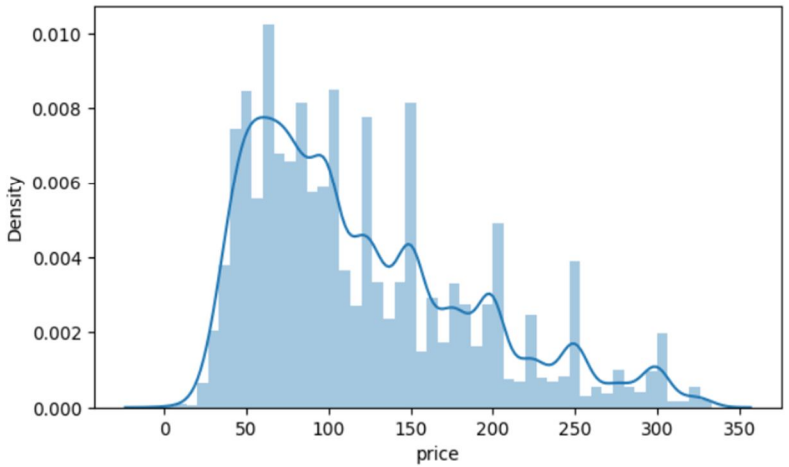


The plot looks good after outlier treatment.

Understanding few insights based on various attributes in the dataset:

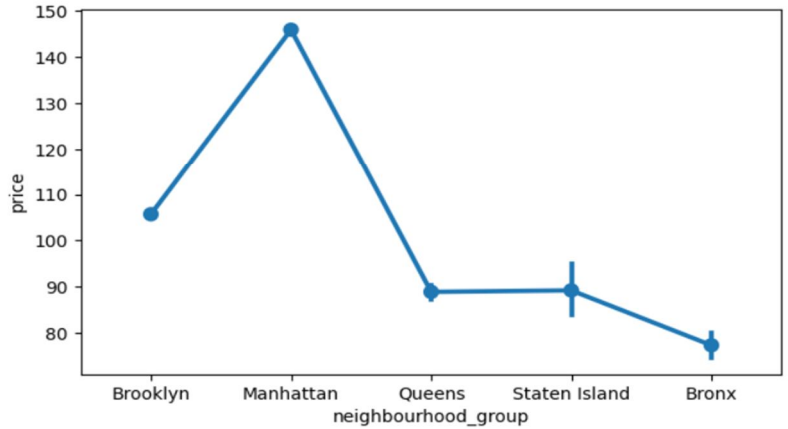
```
## Check for the distribution of prices
sns.distplot(airbnb_newp1['price'])
```

```
<Axes: xlabel='price', ylabel='Density'>
```




```
## look for average price in neighbourhood
from numpy import mean
sns.pointplot(x = 'neighbourhood_group', y='price', data=airbnb_newp1, estimator=mean)
```

```
<Axes: xlabel='neighbourhood_group', ylabel='price'>
```



Manhattan is the costliest in terms of average price ranging to around 150 dollars.

```
## Let's check the expensive listings with respect to prices leading in entire NYC
airbnb_newp.nlargest(5,'price')[['name','neighbourhood_group','neighbourhood','host_name','room_type']]
```

	name	neighbourhood_group	neighbourhood	host_name	room_type
38498	LUXURIOUS 5 bedroom, 4.5 bath home	Manhattan	Upper West Side	Lisa	Entire home/apt
48304	Next to Times Square/Javits/MSG! Amazing 1BR!	Manhattan	Hell's Kitchen	Rogelio	Entire home/apt
46533	Amazing Chelsea 4BR Loft!	Manhattan	Chelsea	Viberlyn	Entire home/apt
30824	Designer's Beautiful 2BR Apartment in NOLITA/SOHO	Manhattan	Nolita	Ilo And Richard	Entire home/apt
22992	Modern Townhouse for Photo, Film & Daytime Ev...	Manhattan	Upper West Side	Lanie	Entire home/apt

All top 5 listings belong to Manhattan.

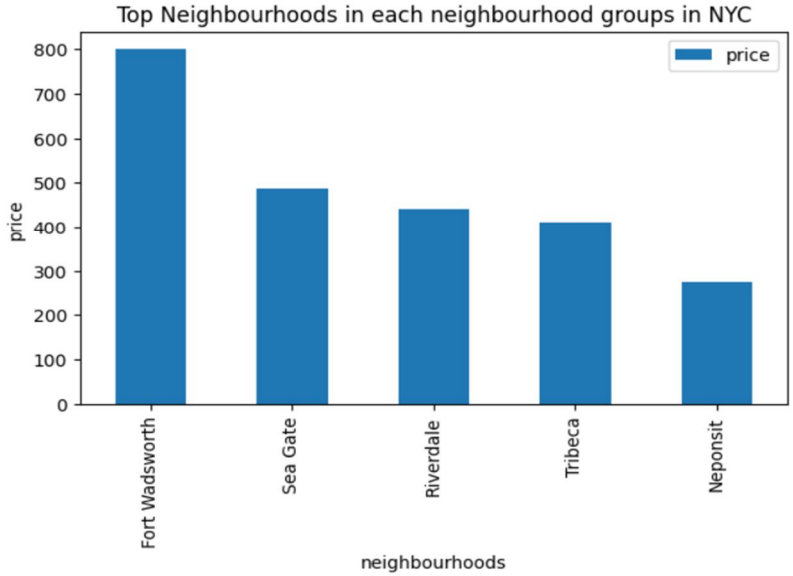
```
## Let's check the cheapest listings with respect to prices leading in entire NYC
airbnb_newp.sort_values(by='price',ascending=True)[['name','neighbourhood_group','neighbourhood','host_name','room_type']][:5]
```

	name	neighbourhood_group	neighbourhood	host_name	room_type
12516	cute and cozy room in brooklyn	Brooklyn	Bedford-Stuyvesant	Ornella	Private room
7864	Comfortable and Large Room	Brooklyn	Flatbush	Kay	Private room
29967	Large bed room share bathroom	Queens	Elmhurst	Cha	Private room
39100	15 minutes From Times Square!!	Manhattan	Washington Heights	Ari	Private room
28700	Cozy room in Loft Apartment - Brooklyn	Queens	Ridgewood	Estefani	Private room

The cheapest listing belongs to Brooklyn.

```
top_neighbour.plot.bar(x='neighbourhood', rot=90, title='Top Neighbourhoods in each neighbourhood groups in NYC')
plt.xlabel('neighbourhoods')
plt.ylabel('price')
```

```
Text(0, 0.5, 'price')
```

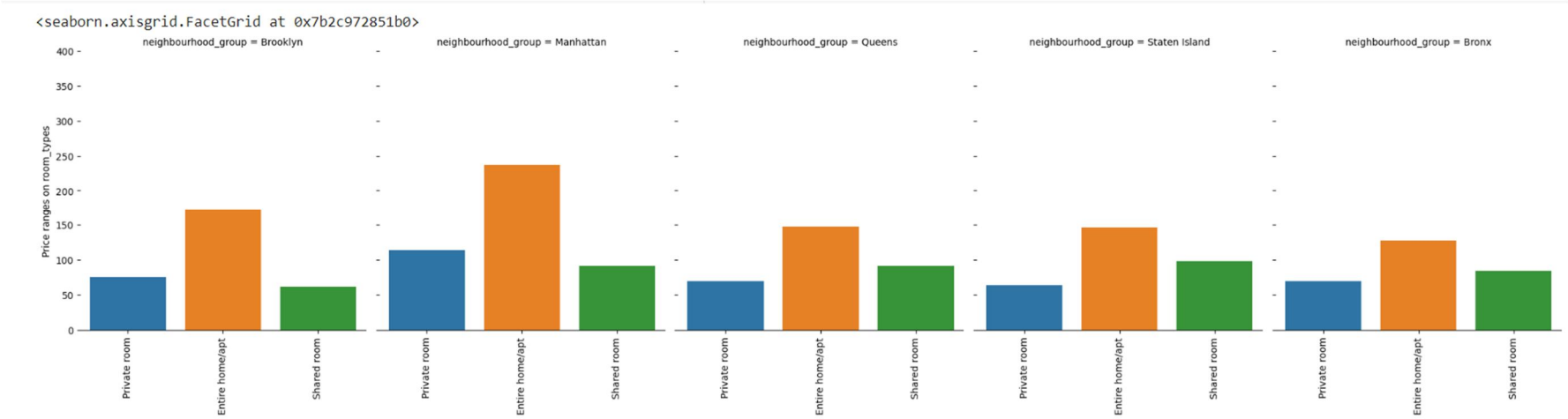


Comparison of Room_types with price on different neighbourhood groups

```
airbnb_newp['room_type'].unique()
```

```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

```
g = sns.catplot(x="room_type", y="price", col="neighbourhood_group",
                data=airbnb_newp, saturation=.8,
                kind="bar",ci=None,  aspect=.9)
(g.set_axis_labels("", "Price ranges on room_types")
  .set_xticklabels(["Private room", "Entire home/apt", "Shared room"],rotation=90)
  .set(ylim=(0, 400))
  .despine(left=True))
```

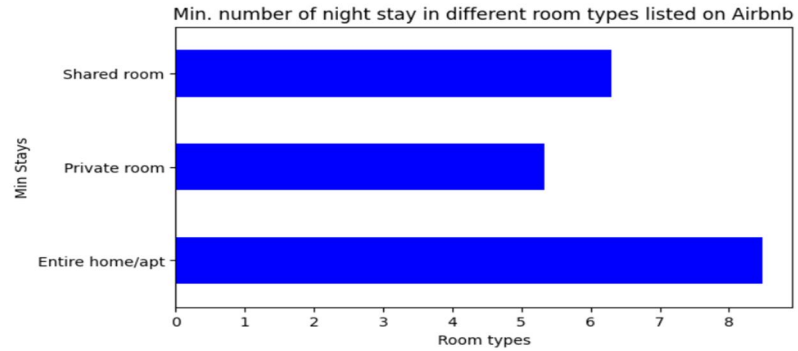


We can conclude that listings with Entire home/apt room_type is mostly preferred which is followed by private rooms in NYC.

Manhattan has the highest price for room types with Entire home/apt ranging up to 240 USD/night, followed by Private room with 110 USD/night.

```
## Check for the number of nights stayed in each room type
airbnb_newp.groupby('room_type')['minimum_nights'].mean().plot(kind='barh',color='b')
plt.title('Min. number of night stay in different room types listed on Airbnb')
plt.xlabel('Room types')
plt.ylabel('Min Stays')
```

```
Text(0, 0.5, 'Min Stays')
```



People prefer living in an entire home/apt on an average of more than 8 nights followed by guests who stayed in shared room where average stay is 6-7 nights.

```
top10_reviewed_listings= airbnb.nlargest(10,'reviews_per_month')
top10_reviewed_listings[['name','reviews_per_month','neighbourhood_group']]
```

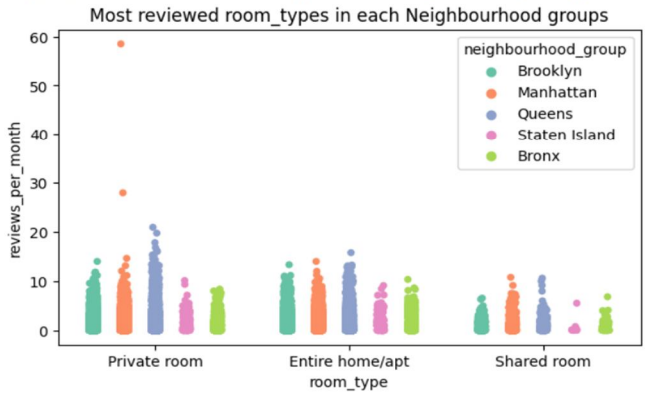
	name	reviews_per_month	neighbourhood_group
42075	Enjoy great views of the City in our Deluxe Room!	58.50	Manhattan
42076	Great Room in the heart of Times Square!	27.95	Manhattan
38870	Lou's Palace-So much for so little	20.94	Queens
27287	JFK Comfort.5 Mins from JFK Private Bedroom & ...	19.75	Queens
28651	JFK 2 Comfort 5 Mins from JFK Private Bedroom	17.82	Queens
29628	JFK 3 Comfort 5 Mins from JFK Private Bedroom	16.81	Queens
20403	Cozy Room Family Home LGA Airport NO CLEANING FEE	16.22	Queens
22469	Cute Tiny Room Family Home by LGA NO CLEANING FEE	16.03	Queens
36238	"For Heaven Cakes"	15.78	Queens
40297	Studio Apartment 6 minutes from JFK Airport	15.32	Queens

Enjoy great views of the City in our Deluxe Room! in Manhattan is the top reviewed listing with 58.50 score followed by Great Room in the heart of Times Square! with 27.95 in Manhattan.

We can also infer that Manhattan is the best suited place for a comfortable stay but with high prices.

```
## monthly reviews with room types in each neighbourhood groups
f,ax = plt.subplots(figsize=(7,4))
ax= sns.stripplot(x='room_type',y='reviews_per_month',hue='neighbourhood_group',dodge=True,data=airbnb,palette='Set2')
ax.set_title('Most reviewed room_types in each Neighbourhood groups')
```

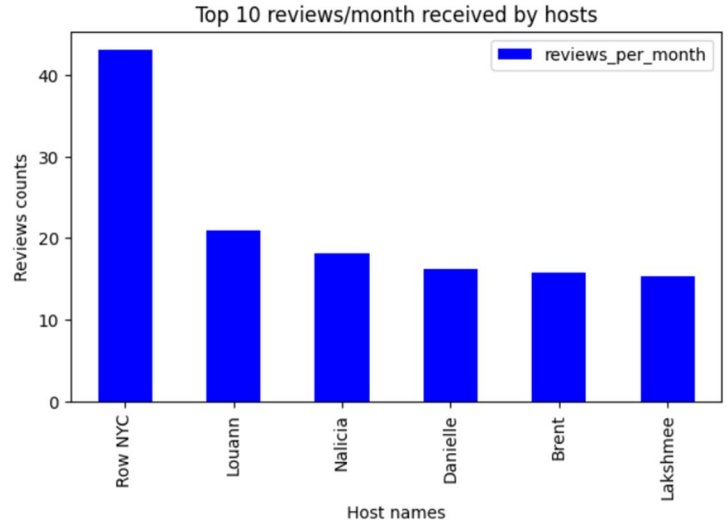
```
Text(0.5, 1.0, 'Most reviewed room_types in each Neighbourhood groups')
```



Private room received the most no of reviews/month, shared rooms received lesser reviews compared to other room types.

```
airbnb_reviews=top10_reviewed_listings.groupby('host_name')['reviews_per_month'].mean()
airbnb_reviews=airbnb_reviews.reset_index().sort_values(by='reviews_per_month',ascending=False)
airbnb_reviews.plot(x='host_name',y='reviews_per_month',kind='bar',color='b')
plt.ylabel('Reviews counts')
plt.xlabel('Host names')
plt.title('Top 10 reviews/month received by hosts')
```

```
Text(0.5, 1.0, 'Top 10 reviews/month received by hosts')
```



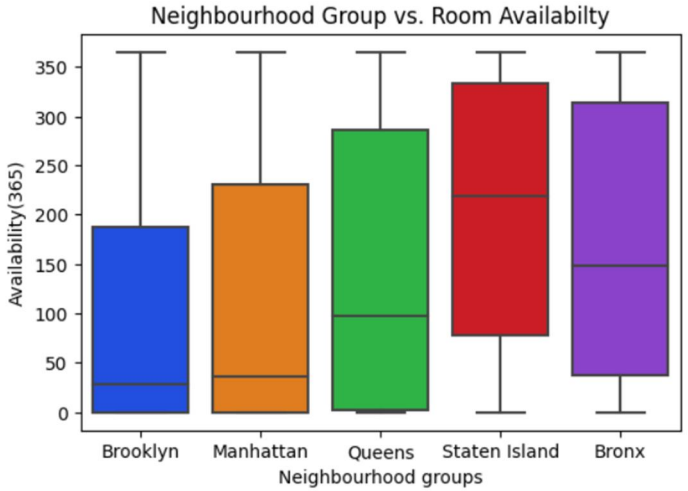
Row NYC has the most reviewed host with more than 40 reviews/month on average.

```
## Check for hosts with most no of listings in NYC
host_most_listings= airbnb.groupby(['host_name','neighbourhood_group'])['calculated_host_listings_count'].sum().reset_index()
most_listings= host_most_listings.nlargest(10,'calculated_host_listings_count')
most_listings
```

	host_name	neighbourhood_group	calculated_host_listings_count
13217	Sonder (NYC)	Manhattan	106929
1834	Blueground	Manhattan	53360
7275	Kara	Manhattan	14669
6540	Jeremy & Laura	Manhattan	9216
13216	Sonder	Manhattan	9216
2901	Corporate Housing	Manhattan	8281
7480	Kazuya	Queens	8137
7546	Ken	Manhattan	7500
11399	Pranjal	Manhattan	4225
9856	Mike	Manhattan	2824

Sonder(NYC) tops the most expensive neighbourhood group in NYC with 106929 properties which is then followed by Blueground and Kara respectively.

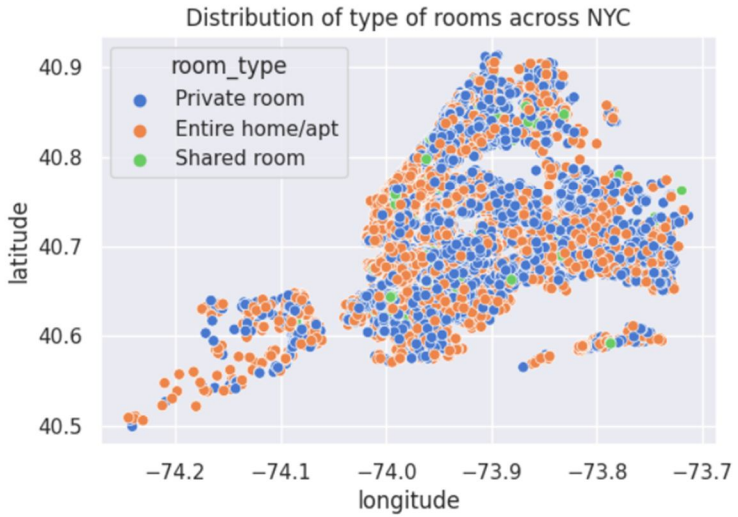
```
f,ax = plt.subplots(figsize=(6,4))
ax=sns.boxplot(x='neighbourhood_group',y='availability_365',data=airbnb,palette="bright")
plt.title("Neighbourhood Group vs. Room Availabilty")
plt.xlabel('Neighbourhood groups')
plt.ylabel('Availability(365)')
plt.show()
```



listings in Staten Island is available throughout the year for more than 300 days

```
sns.set(rc={"figure.figsize": (6,4)})
ax= sns.scatterplot(x=airbnb_newp.longitude, y=airbnb_newp.latitude,hue=airbnb.room_type,palette='muted')
ax.set_title('Distribution of type of rooms across NYC')
```

```
Text(0.5, 1.0, 'Distribution of type of rooms across NYC')
```



Shared rooms are very few in NYC compared to private and Entire home/apt.