EXPLORATORY DATA ANALYSIS OF AIRBNB DATASET

Presented by :- Sourav Dutta

Introduction

The Airbnb dataset provides a wealth of information about property listings, including details such as location, price, availability, and ratings. By utilizing the power of pandas, matplotlib, and seaborn, we can gain valuable insights from this data to better understand the trends and patterns within the Airbnb ecosystem.Pandas will be used to efficiently manipulate and clean the dataset, making it suitable for analysis. Matplotlib and seaborn will then be employed to create visualizations that help us visualize the data and extract meaningful information.

By the end, we aim to have a comprehensive understanding of the Airbnb market, which can be valuable for both hosts and guests alike.

UNDERSTAND THE GIVEN VARIABLES

Listing_id :- This is a unique identifier for each listing in the dataset.

Listing_name :- This is the name or title of the listing, as it appears on the Airbnb website.

Host_id :- This is a unique identifier for each host in the dataset.

Host_name :- This is the name of the host as it appears on the Airbnb website.

Neighbourhood_group :- This is a grouping of neighborhoods in New York City, such as Manhattan or Brooklyn.

Neighbourhood :- This is the specific neighborhood in which the listing is located.

Latitude :- This is the geographic latitude of the listing.

Longitude :- This is the geographic longitude of the listing.

Room_type :- This is the type of room or property being offered, such as an entire home, private room, shared room.

Price :- This is the nightly price for the listing, in US dollars.

Minimum_nights :- This is the minimum number of nights that a guest must stay at the listing.

Total_reviews :- This is the total number of reviews that the listing has received.

Reviews_per_month :- This is the average number of reviews that the listing receives per month.

Availability_365 :- This is the number of days in the next 365 days that the listing is available for booking.

PROBLEM STATEMENTS

(1) Find Distribution Of Airbnb Bookings Price Range.

(2) Find Total Listing/Property count in Each Neighborhood Group in NYC.

(3) Find Average Price Of listings/property in each Neighborhood Groups and also Neighborhoods.

(4) Find Top Neighborhoods and Hosts by Listing/property in entire NYC.

(5) Find the Number Of Active Hosts Per Location by Each Neighborhood Groups.

(6) Find Total Counts Of Each Room Types in entire NYC.

(7) Find Stay Requirement counts by Minimum Nights.

(8) Find the total numbers of Reviews and Maximum Reviews by Each Neighborhood Group.

(9) Find Most reviewed room type in Neighborhood groups per month.

(10) Find Best location listing/property location for travelers.

(11) Find Best Location Listing/Property Location For Travelers and Hosts.

IMPORT LIBRARIES

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

DATA LOADING

```
In [2]: air=pd.read_csv("D:/Projects/Airbnb NYC 2019.csv")
        air
```

Out[2]:

	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
...	...	...	...	...	...	...	...	...	...	...
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90

48895 rows × 16 columns

Data Exploration & Data Cleaning

```
In [3]: air.head().T
```

Out[3]:

	0	1	2	3	4
id	2539	2595	3647	3831	5022
name	Clean & quiet apt home by the park	Skylit Midtown Castle	THE VILLAGE OF HARLEM....NEW YORK !	Cozy Entire Floor of Brownstone	Entire Apt: Spacious Studio/Loft by central park
host_id	2787	2845	4632	4869	7192
host_name	John	Jennifer	Elisabeth	LisaRoxanne	Laura
neighbourhood_group	Brooklyn	Manhattan	Manhattan	Brooklyn	Manhattan
neighbourhood	Kensington	Midtown	Harlem	Clinton Hill	East Harlem
latitude	40.64749	40.75362	40.80902	40.68514	40.79851
longitude	-73.97237	-73.98377	-73.9419	-73.95976	-73.94399
room_type	Private room	Entire home/apt	Private room	Entire home/apt	Entire home/apt
price	149	225	150	89	80
minimum_nights	1	1	3	1	10
number_of_reviews	9	45	0	270	9
last_review	19-10-2018	21-05-2019	NaN	05-07-2019	19-11-2018
reviews_per_month	0.21	0.38	NaN	4.64	0.1
calculated_host_listings_count	6	2	1	1	1
availability_365	365	355	365	194	0

Check all the Columns name

In [4]:
```
air.columns
```

Out[4]:
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365'],
      dtype='object')
```

Re-name few columns for better understanding

In [5]:
```
airbnb=air.rename(columns={"id":"listing_id","name":"listing_name","number_of_reviews":"total_reviews","calcul
airbnb.head(2)
```

Out[5]:

	listing_id	listing_name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_
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	

Some Information about the Dataset.

checking shape of this dataset

In [6]:
```
airbnb.shape
```

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

basic information about the dataset

```
In [7]: airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   listing_id           48895 non-null  int64  
 1   listing_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  total_reviews        48895 non-null  int64  
 12  last_review          38843 non-null  object 
 13  reviews_per_month    38843 non-null  float64
 14  Host_listing_count   48895 non-null  int64  
 15  availability_365     48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

check duplicate rows in dataset

```
In [8]: airbnb.duplicated().sum()
```

```
Out[8]: 0
```

checking null values of each columns

In [9]:
```
airbnb.isnull().sum()
```

Out[9]:
```
listing_id                0
listing_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
total_reviews             0
last_review           10052
reviews_per_month     10052
Host_listing_count        0
availability_365          0
dtype: int64
```

Here we can see "host_name" and "listing_name" are not that much of null values, so first we are good to fill those with some substitutes in both the columns first.

In [10]:

```
airbnb["listing_name"].fillna("unknown",inplace=True)
airbnb["host_name"].fillna("unknown",inplace=True)
airbnb.isnull().sum()
```

Out[10]:

```
listing_id                 0
listing_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
total_reviews              0
last_review            10052
reviews_per_month      10052
Host_listing_count         0
availability_365           0
dtype: int64
```

now, the columns last_review and reviews_per_month have total 10052 null values each.

last_review column is not required for our analysis as compared to number_of_reviews & reviews_per_month. We're good to drop this column.

In [11]:

```
airbnb=airbnb.drop(['last_review'], axis=1)
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   listing_id           48895 non-null  int64  
 1   listing_name         48895 non-null  object 
 2   host_id              48895 non-null  int64  
 3   host_name            48895 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  total_reviews        48895 non-null  int64  
 12  reviews_per_month    38843 non-null  float64
 13  Host_listing_count   48895 non-null  int64  
 14  availability_365     48895 non-null  int64  
dtypes: float64(3), int64(7), object(5)
memory usage: 5.6+ MB
```

listing_id also not that much of important for our analysis but i dont remove because of listing_id and listing_name is pair and removing listing_id it still wont make much difference.

The reviews_per_month column also containing null values and we can simple put 0 reviews by replacing NAN's i think this is make sense

In [12]:

```
airbnb['reviews_per_month']=airbnb['reviews_per_month'].replace(to_replace=np.nan,value=0).astype('int64')
airbnb['reviews_per_month'].isnull().sum()
```

Out[12]:

```
0
```

In [13]:
```
airbnb.sample(5)
```

Out[13]:

	listing_id	listing_name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	mi
26109	20820417	The Penthouse	66096509	Eric	Manhattan	East Harlem	40.79398	-73.94159	Entire home/apt	300	
32366	25328081	Basic Little Italy 3	181074926	Billy	Manhattan	Little Italy	40.71766	-73.99827	Private room	110	
46305	35189990	Spacious 5 BED, 2 Full BATH (15 mins to Manhat...	264990100	Yonaton	Brooklyn	Crown Heights	40.66566	-73.93282	Entire home/apt	395	
34061	26989176	Modern, Cozy Apt Share for Young Professionals	2822805	Ollie	Brooklyn	Bedford-Stuyvesant	40.68585	-73.94919	Private room	60	
33674	26670812	Lucky home 温馨如家	88576580	Anna	Queens	Flushing	40.75432	-73.82273	Private room	60	

Check Unique Value for some Columns

In [14]:
```
airbnb['host_id'].nunique()
```

Out[14]:
```
37457
```

In [15]:
```
airbnb['listing_id'].nunique()
```

Out[15]:
```
48895
```

In [16]:
```
airbnb['neighbourhood'].nunique()
```

Out[16]:
```
221
```

In [17]:
```
airbnb['host_name'].nunique()
```

Out[17]:
```
11453
```

Describe the Dataset.

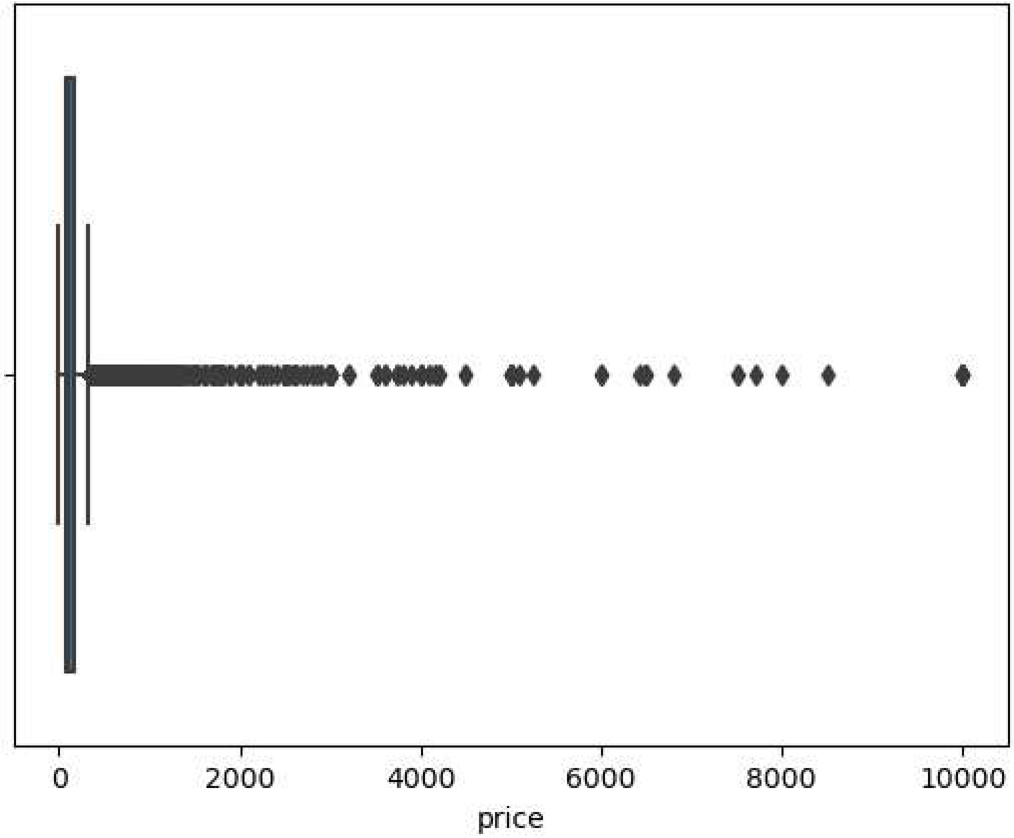
In [18]:
```
airbnb.describe()
```

Out[18]:

	listing_id	host_id	latitude	longitude	price	minimum_nights	total_reviews	reviews_per_month	Host_
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	0.806258	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.502767	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	1.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.000000	

In [19]:
```
sns.boxplot(x= airbnb['price'])
airbnb.shape
```

Out[19]:
```
(48895, 15)
```



In [20]:

```
def iqr_technique(DFcolumn):
  Q1 = np.percentile(DFcolumn, 25)
  Q3 = np.percentile(DFcolumn, 75)
  IQR = Q3 - Q1
  lower_range = Q1 - (1.5 * IQR)
  upper_range = Q3 + (1.5 * IQR)                           # interquantile range

  return lower_range,upper_range
```

In [21]:

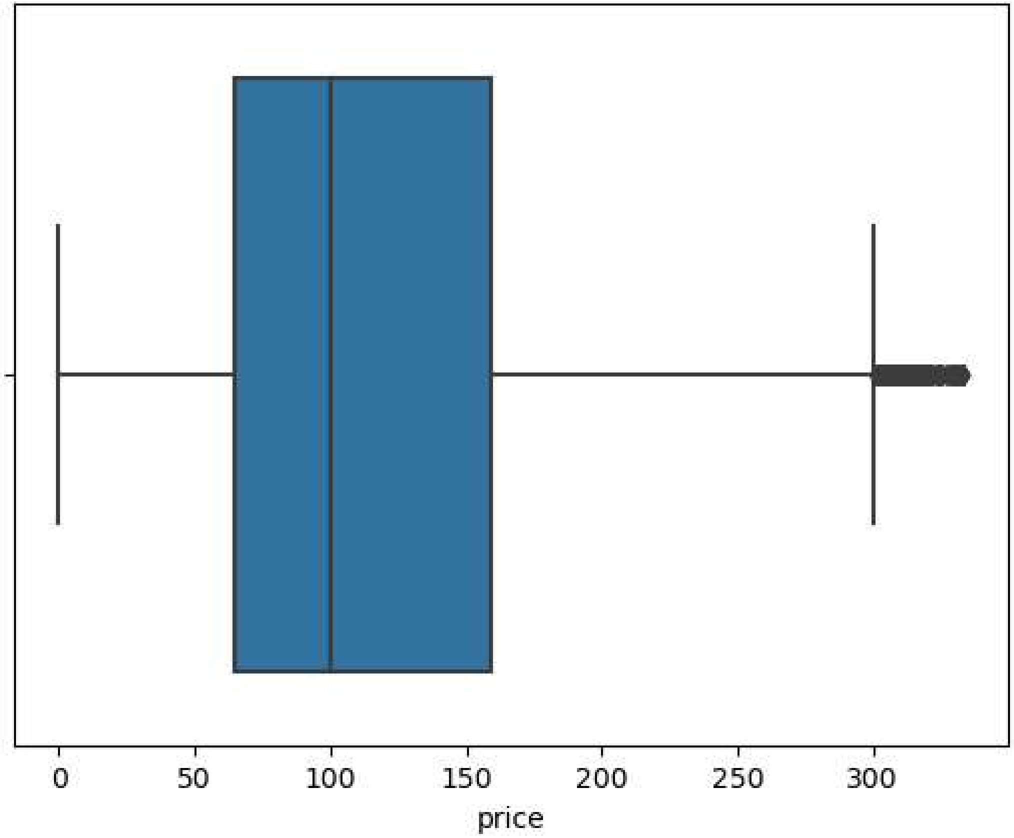
```
lower_bound,upper_bound = iqr_technique(airbnb['price'])

airbnb = airbnb[(airbnb.price>lower_bound) & (airbnb.price<upper_bound)]
airbnb.head(5)
```

Out[21]:

	listing_id	listing_name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minim
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 [22]: sns.boxplot(x= airbnb['price'])
         airbnb.shape
```

```
Out[22]: (45918, 15)
```



Find the Distribution Of Airbnb Bookings Price Range

```
In [23]: airbnb.groupby(['listing_id']).sum()
         airbnb=airbnb.sort_values(['price'],ascending=True)
         airbnb.head(10)
```

Out[23]:

	listing_id	listing_name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	n
26841	21291569	Coliving in Brooklyn! Modern design / Shared room	101970559	Sergii	Brooklyn	Bushwick	40.69211	-73.90670	Shared room	0	
25433	20333471	★Hostel Style Room | Ideal Traveling Buddies★	131697576	Anisha	Bronx	East Morrisania	40.83296	-73.88668	Private room	0	
25634	20523843	MARTIAL LOFT 3: REDEMPTION (upstairs, 2nd room)	15787004	Martial Loft	Brooklyn	Bushwick	40.69467	-73.92433	Private room	0	
25753	20608117	Sunny, Quiet Room in Greenpoint	1641537	Lauren	Brooklyn	Greenpoint	40.72462	-73.94072	Private room	0	
25778	20624541	Modern apartment in the heart of Williamsburg	10132166	Aymeric	Brooklyn	Williamsburg	40.70838	-73.94645	Entire home/apt	0	
25794	20639628	Spacious comfortable master bedroom with nice ...	86327101	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68173	-73.91342	Private room	0	
25795	20639792	Contemporary bedroom in brownstone with nice view	86327101	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68279	-73.91170	Private room	0	
25796	20639914	Cozy yet spacious private brownstone bedroom	86327101	Adeyemi	Brooklyn	Bedford-Stuyvesant	40.68258	-73.91284	Private room	0	
26866	21304320	Best Coliving space ever! Shared room.	101970559	Sergii	Brooklyn	Bushwick	40.69166	-73.90928	Shared room	0	
26259	20933849	the best you can find	13709292	Qiuchi	Manhattan	Murray Hill	40.75091	-73.97597	Entire home/apt	0	

Ploting a Histogram to show Distribution of Price range

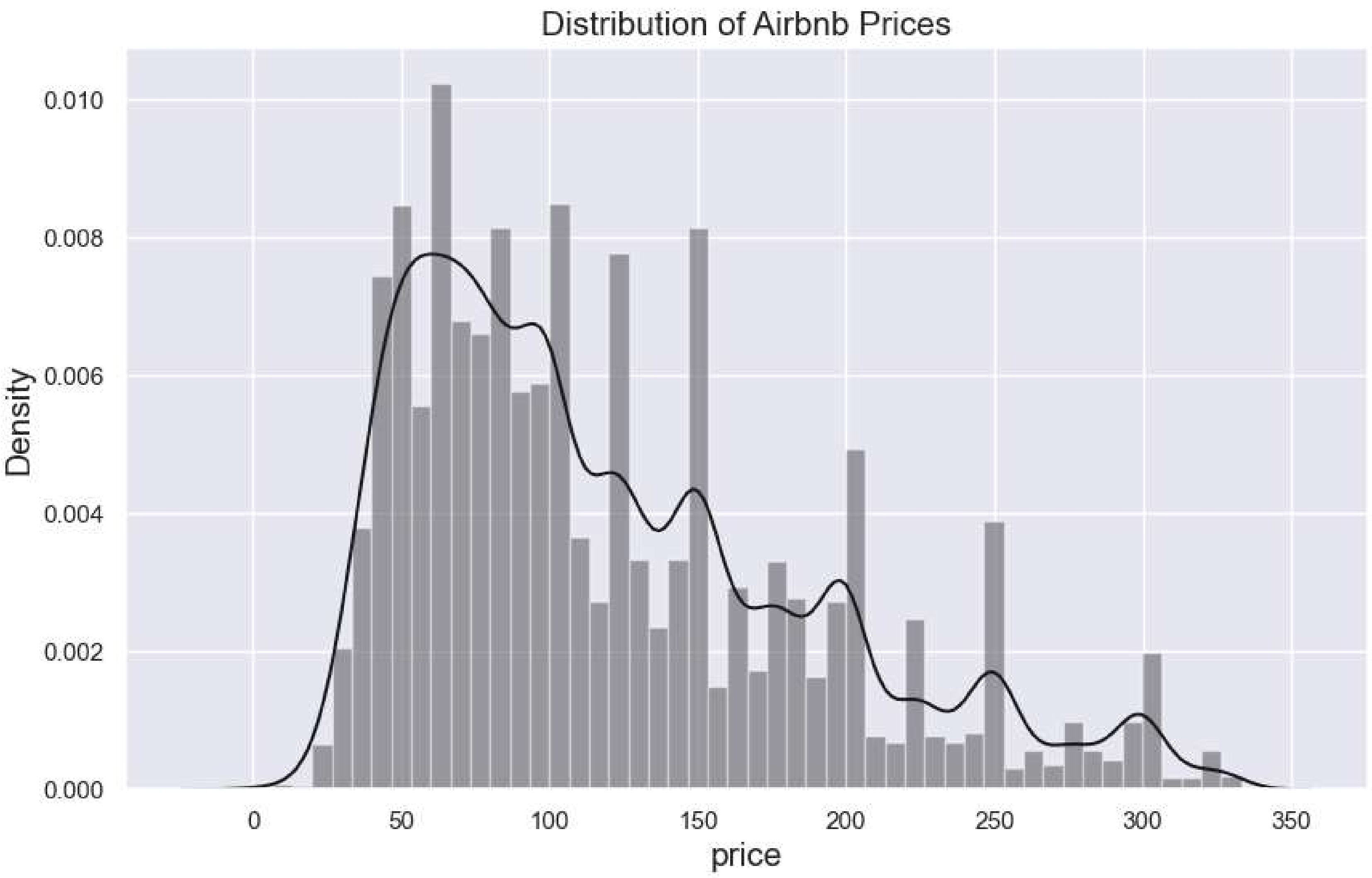
In [24]:

```
plt.figure(figsize=(10,6))
sns.set_theme(style='darkgrid')
sns.distplot(airbnb['price'],color='k')
plt.xlabel("price",fontsize=15)
plt.ylabel("Density",fontsize=15)
plt.title("Distribution of Airbnb Prices",fontsize=15)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_7532\4046996183.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwaskom/de44147ed29
74457ad6372750bbe5751)

  sns.distplot(airbnb['price'],color='k')
```

Out[24]:

```
Text(0.5, 1.0, 'Distribution of Airbnb Prices')
```



Find Total Listing/Property count in Each Neighborhood Group

In [25]:

```
cou=airbnb.groupby(['neighbourhood_group'])
total=cou[['neighbourhood_group','Host_listing_count']].count()
res=total.sort_values(['Host_listing_count'],ascending=True)
res
```

Out[25]:

	neighbourhood_group	Host_listing_count
neighbourhood_group		
Staten Island	365	365
Bronx	1070	1070
Queens	5567	5567
Brooklyn	19415	19415
Manhattan	19501	19501

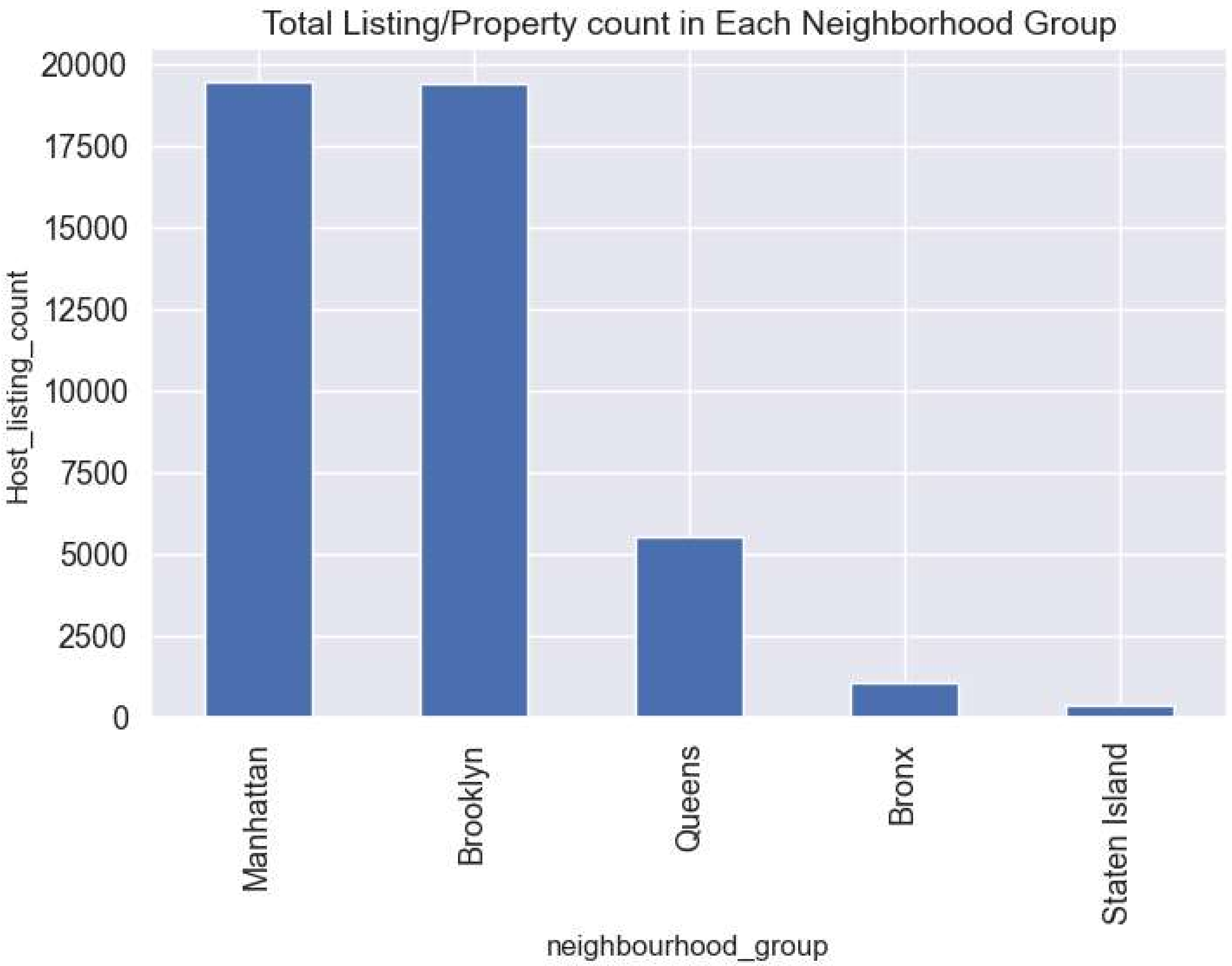
Ploting a "Count_Plot" to show Total Listing/Property count in Each Neighborhood Group

In [26]:

```
plt.figure(figsize=(8,5))
airbnb['neighbourhood_group'].value_counts().plot(kind='bar',fontsize=13)
plt.xlabel("neighbourhood_group",fontsize=12)
plt.ylabel("Host_listing_count",fontsize=12)
plt.title("Total Listing/Property count in Each Neighborhood Group",fontsize=14)
```

Out[26]:

```
Text(0.5, 1.0, 'Total Listing/Property count in Each Neighborhood Group')
```



Average Price Of listing/Property in Each Neighborhood Groups and also Neighborhood.

```
In [27]: avg=airbnb.groupby(["neighbourhood_group","neighbourhood"])
         res=avg[['price']].mean()
         res.head(5)
```

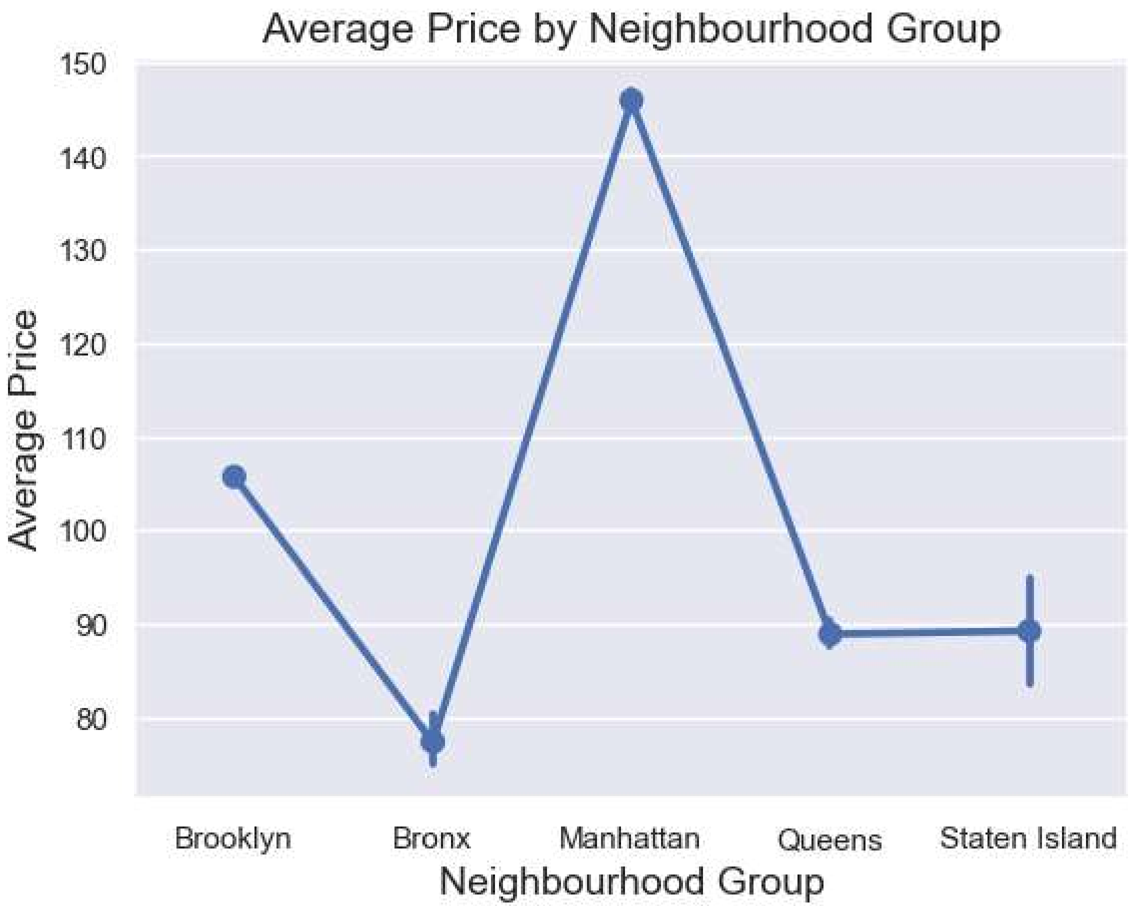
```
Out[27]:
                                                price
         neighbourhood_group  neighbourhood
                                   Allerton  78.756098
                                 Baychester  75.428571
                       Bronx        Belmont  77.125000
                                  Bronxdale  57.105263
                                Castle Hill  63.000000
```

In [28]:

```
from statistics import mean
sns.pointplot(x='neighbourhood_group',y='price',data=airbnb,estimator=np.mean)
plt.xlabel('Neighbourhood Group',fontsize=14)
plt.ylabel('Average Price',fontsize=14)
plt.title('Average Price by Neighbourhood Group',fontsize=15)
```

Out[28]:

```
Text(0.5, 1.0, 'Average Price by Neighbourhood Group')
```



Price Distribution Of Each Neighborhood Group

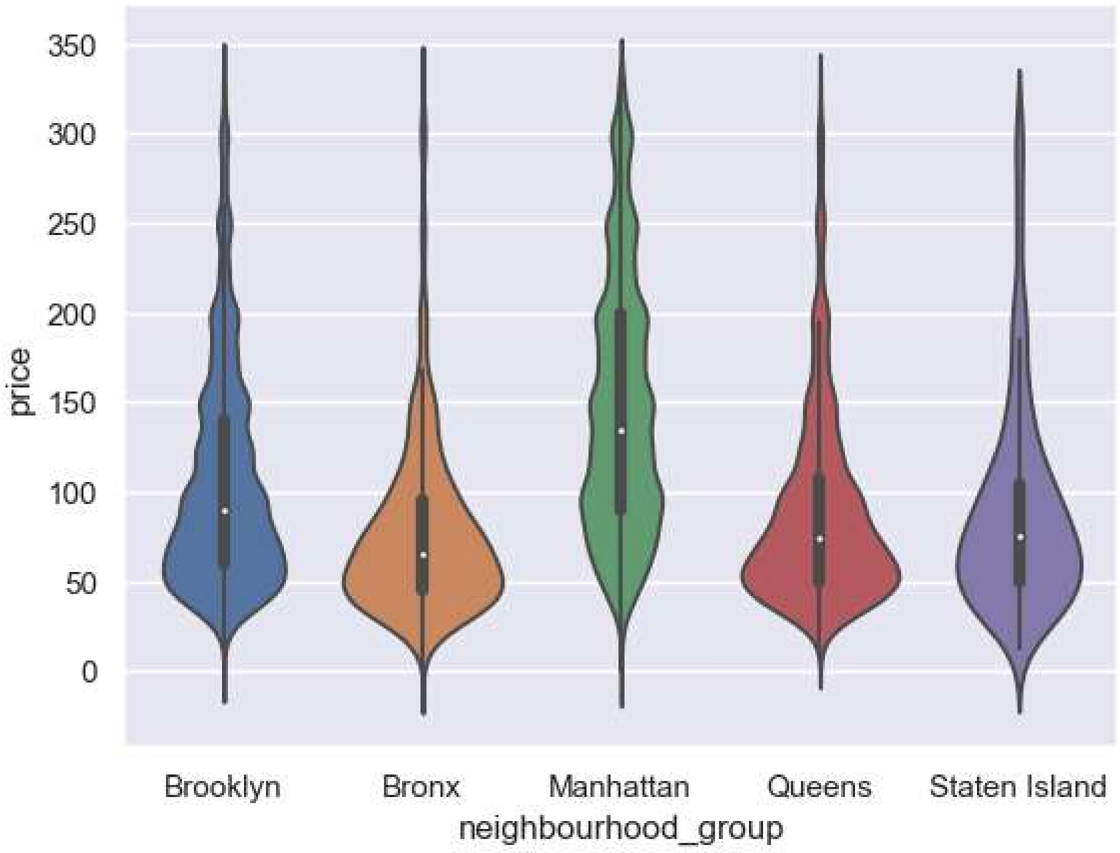
```
In [29]: price=airbnb.groupby(['neighbourhood_group'])
         tot=price['price'].sum()
         tot
```

```
Out[29]: neighbourhood_group
         Bronx              82781
         Brooklyn         2052158
         Manhattan        2845286
         Queens            494931
         Staten Island      32571
         Name: price, dtype: int64
```

plotting a "violineplot" to show price distribution

```
In [30]: sns.violinplot(x='neighbourhood_group',y='price',data=airbnb)
```

```
Out[30]: <Axes: xlabel='neighbourhood_group', ylabel='price'>
```



Find Top neighborhoods and hosts by listing/property in entire NYC.

```
In [31]: #For Neighbourhood
         top=airbnb['neighbourhood'].value_counts()[:10].reset_index()
         top
```

```
Out[31]:
            neighbourhood  count
   0         Williamsburg   3732
   1   Bedford-Stuyvesant   3638
   2               Harlem   2585
   3             Bushwick   2438
   4      Upper West Side   1788
   5        Hell's Kitchen   1731
   6         East Village   1714
   7      Upper East Side   1670
   8        Crown Heights   1519
   9              Midtown   1143
```

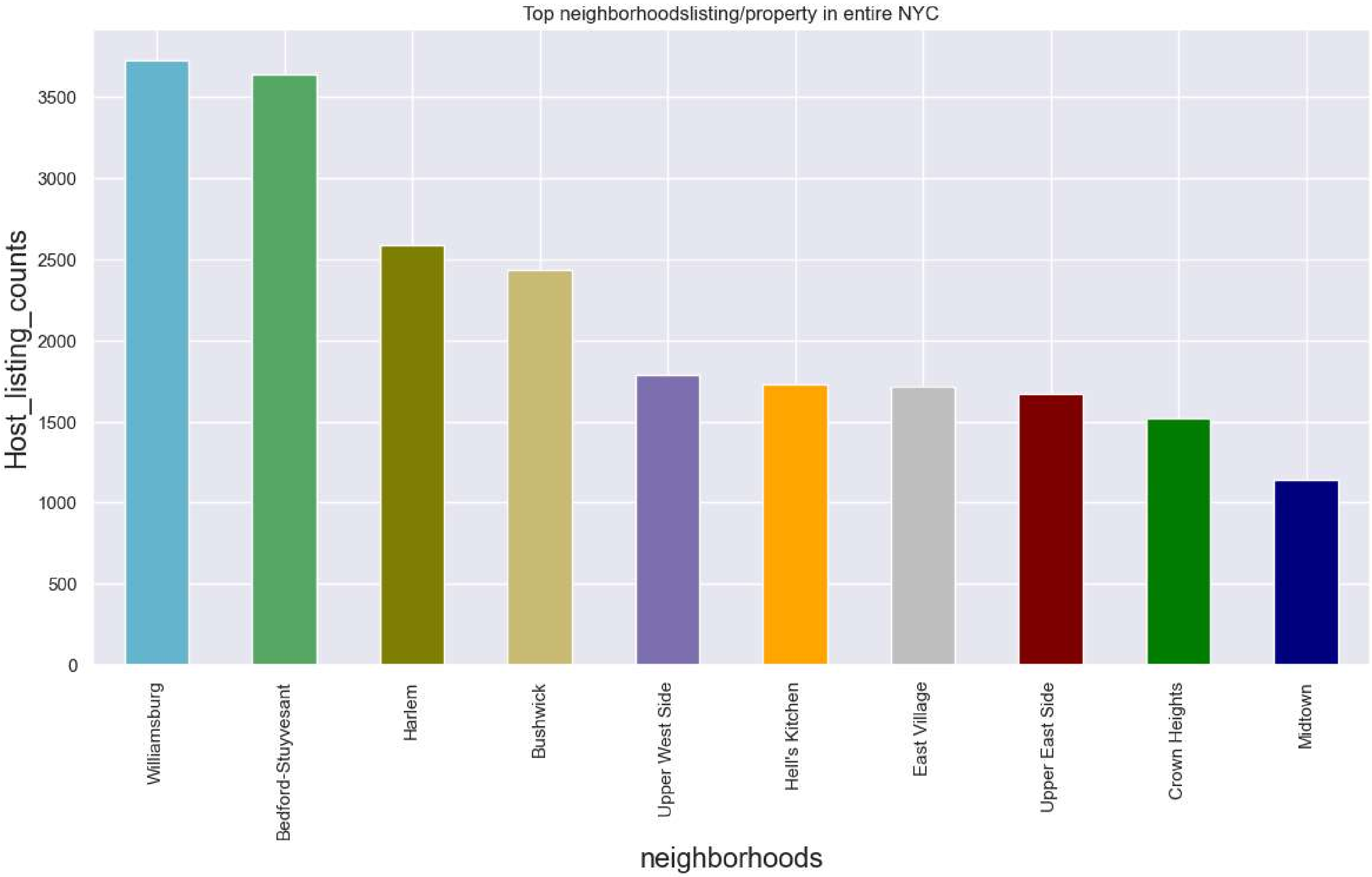
Plotting a "Barplot" to show Top neighborhoods and hosts by listing/property in entire NYC

In [32]:

```
colors = ['c', 'g', 'olive', 'y', 'm', 'orange', '#C0C0C0', '#800000', '#008000', '#000080']
top_10=airbnb['neighbourhood'].value_counts().nlargest(10)
top_10.plot(kind='bar', figsize=(14,7), color=colors)
plt.xlabel('neighborhoods',fontsize=17)
plt.ylabel('Host_listing_counts',fontsize=17)
plt.title('Top neighborhoodslisting/property in entire NYC')
```

Out[32]:

```
Text(0.5, 1.0, 'Top neighborhoodslisting/property in entire NYC')
```



In [33]:

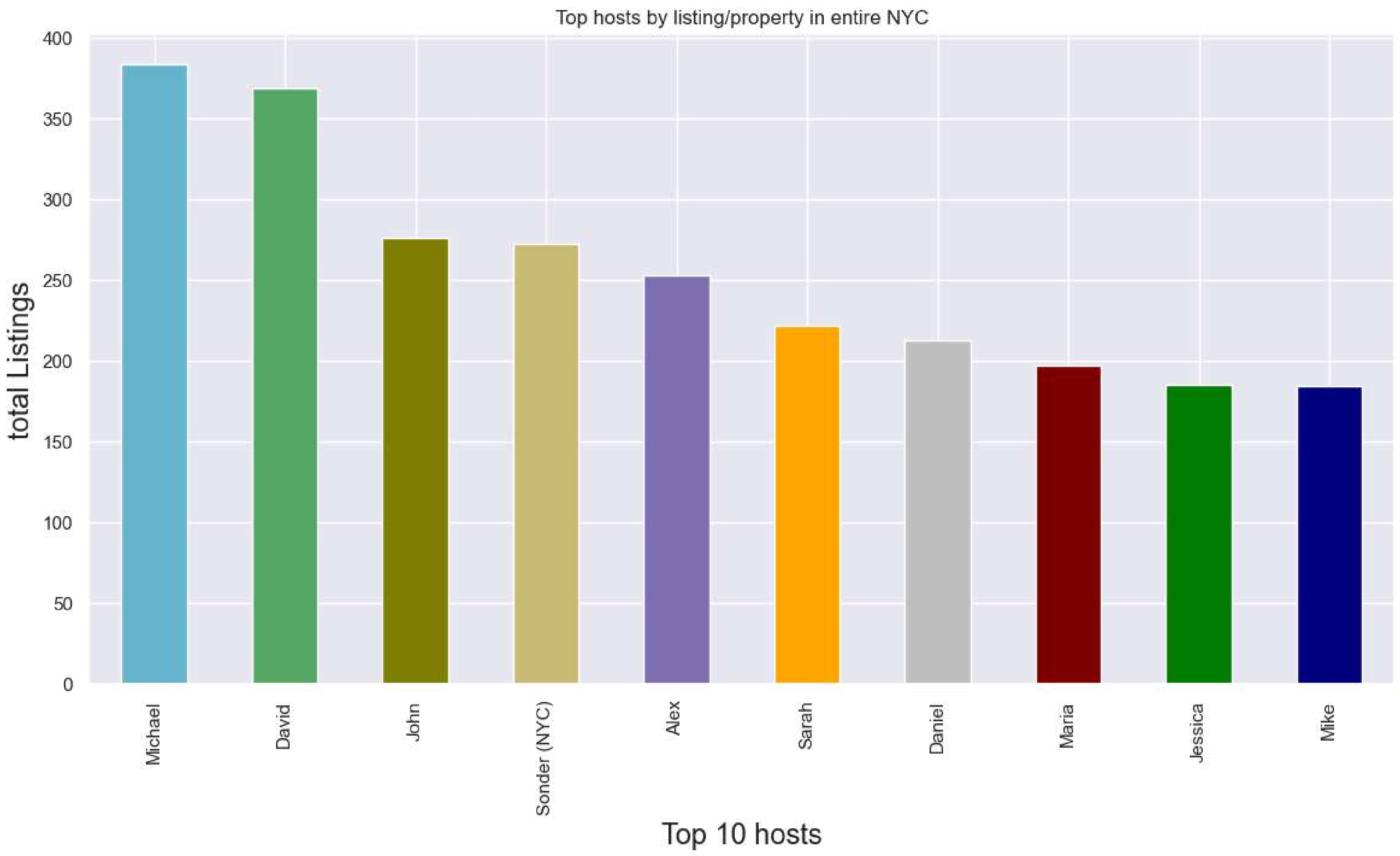
```
#for Hosts
host=airbnb['host_name'].value_counts()[:10].reset_index()
host
```

Out[33]:

	host_name	count
0	Michael	383
1	David	368
2	John	276
3	Sonder (NYC)	272
4	Alex	253
5	Sarah	221
6	Daniel	212
7	Maria	197
8	Jessica	185
9	Mike	184

In [34]:
```
colors = ['c', 'g', 'olive', 'y', 'm', 'orange', '#C0C0C0', '#800000', '#008000', '#000080']
top_a=airbnb['host_name'].value_counts().nlargest(10)
top_a.plot(kind='bar', figsize=(14,7), color=colors)
plt.xlabel('Top 10 hosts',fontsize=17)
plt.ylabel('total Listings',fontsize=17)
plt.title('Top hosts by listing/property in entire NYC')
```

Out[34]:
```
Text(0.5, 1.0, 'Top hosts by listing/property in entire NYC')
```



Find the no. of Active hosts per location by each neighborhood groups.

In [35]:
```
act=airbnb.groupby('neighbourhood_group')
res=act[['neighbourhood_group','Host_listing_count']].count()
res
```

Out[35]:

	neighbourhood_group	Host_listing_count
neighbourhood_group		
Bronx	1070	1070
Brooklyn	19415	19415
Manhattan	19501	19501
Queens	5567	5567
Staten Island	365	365

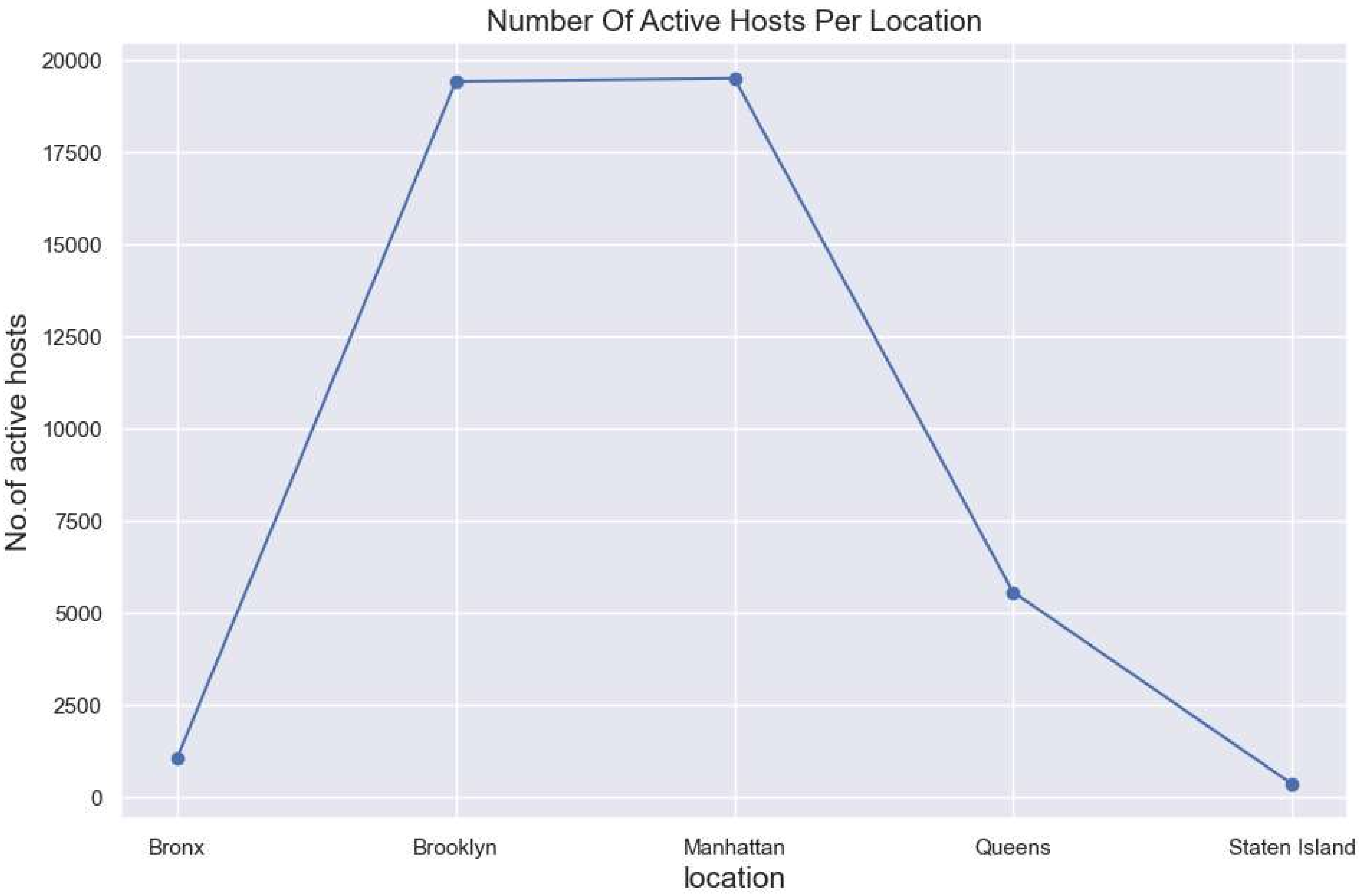
Number Of Active Hosts Per Location Using Line Chart

In [36]:

```
hosts_per_location=airbnb.groupby('neighbourhood_group')['listing_id'].count()
loca=hosts_per_location.index
coun=hosts_per_location.values
plt.figure(figsize=(11,7))
plt.plot(loca,coun, marker='o')
plt.xlabel("location",fontsize=15)
plt.ylabel("No.of active hosts",fontsize=15)
plt.title("Number Of Active Hosts Per Location",fontsize=15)
```

Out[36]:

```
Text(0.5, 1.0, 'Number Of Active Hosts Per Location')
```



Find total count in each Room Types in entire NYC

In [37]:

```
fin=airbnb[["room_type"]].value_counts()
fin
```

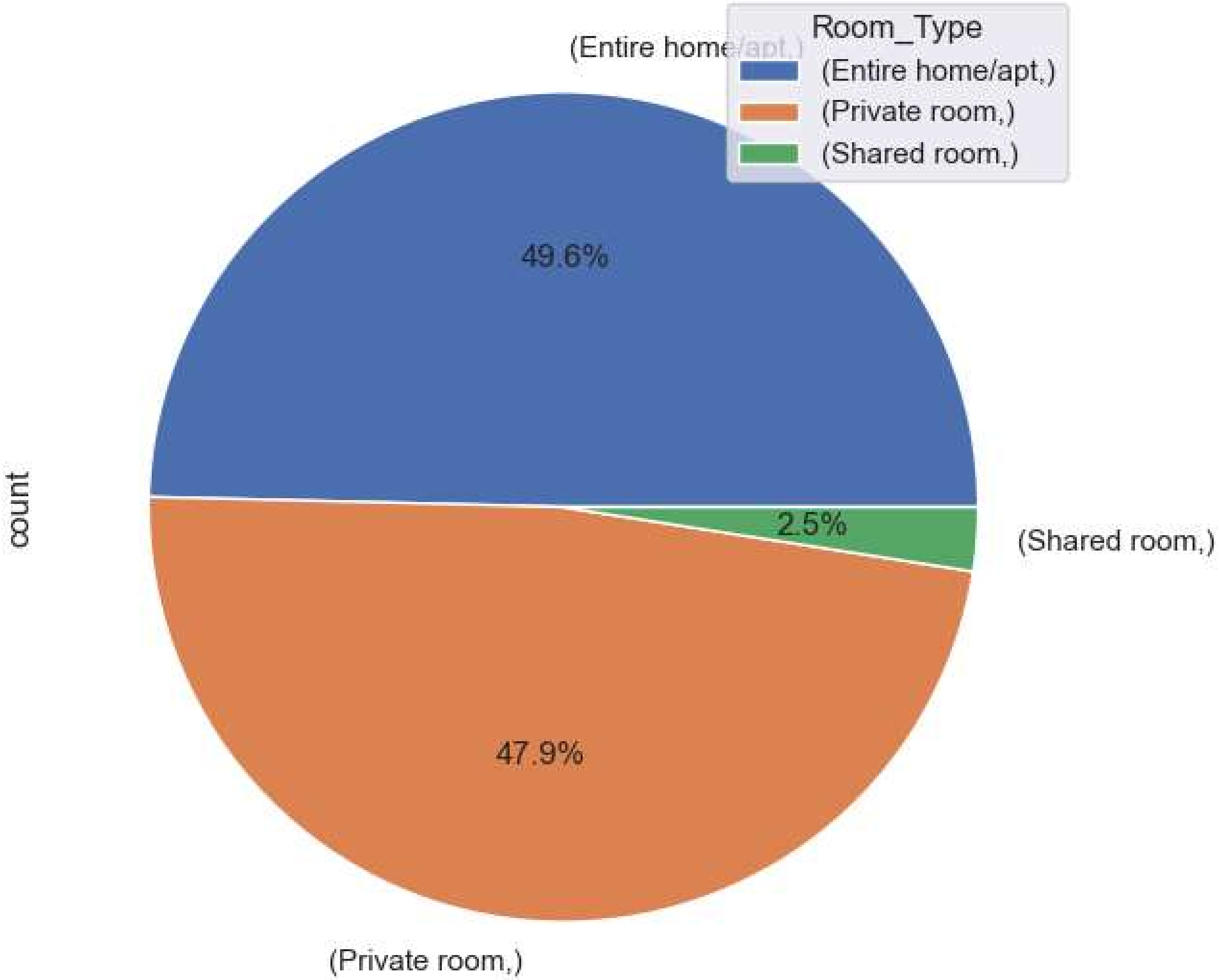
Out[37]:

```
room_type
Entire home/apt    22784
Private room       21996
Shared room         1138
Name: count, dtype: int64
```

plot a pie chart for visualize each romm type for entire nyc.

```
In [38]: plt.figure(figsize=(10,7))
         fin.plot(kind='pie',autopct='%1.1f%%')
         plt.legend(title='Room_Type')
```

```
Out[38]: <matplotlib.legend.Legend at 0x27aaabfd210>
```



Find Stay Requirement Counts By Mininum Nights

In [39]:

```
airbnb.columns
```

Out[39]:

```
Index(['listing_id', 'listing_name', 'host_id', 'host_name',
       'neighbourhood_group', 'neighbourhood', 'latitude', 'longitude',
       'room_type', 'price', 'minimum_nights', 'total_reviews',
       'reviews_per_month', 'Host_listing_count', 'availability_365'],
      dtype='object')
```

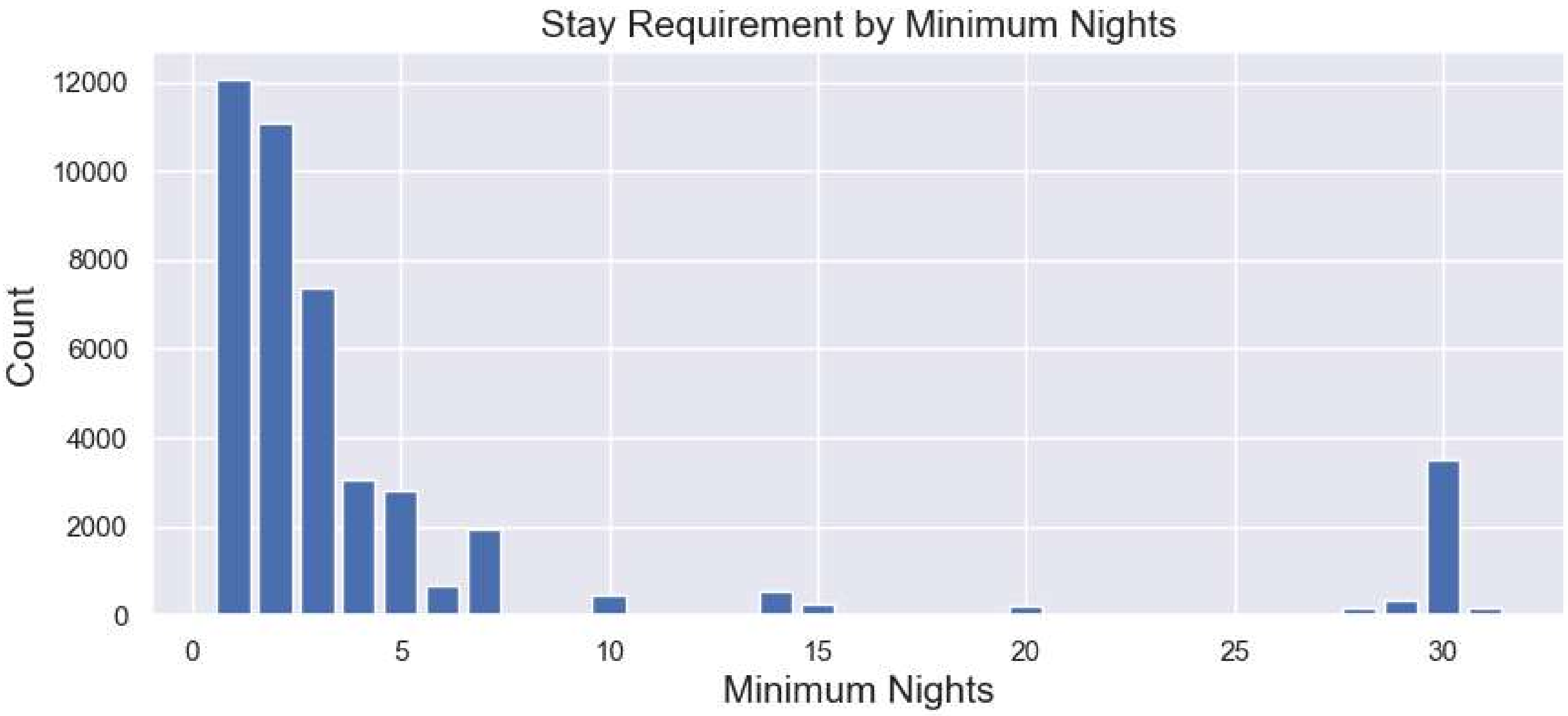
```
In [40]: min_nights_count = airbnb.groupby('minimum_nights').size().reset_index(name = 'count')
         min_nights_count = min_nights_count.sort_values('count', ascending=False)
         min_nights_count = min_nights_count.head(15)
         min_nights_count = min_nights_count.reset_index(drop=True)
         min_nights_count
```

```
Out[40]:
        minimum_nights   count
    0                1   12067
    1                2   11080
    2                3    7375
    3               30    3489
    4                4    3066
    5                5    2821
    6                7    1951
    7                6     679
    8               14     539
    9               10     462
   10               29     327
   11               15     272
   12               20     215
   13               31     189
   14               28     173
```

Stay Requirement Counts By Mininum Nights using bar plot

In [41]:
```
minimum_nights = min_nights_count['minimum_nights']
count = min_nights_count['count']
plt.figure(figsize=(10, 4))
plt.bar(minimum_nights, count)
plt.xlabel('Minimum Nights', fontsize='15')
plt.ylabel('Count', fontsize='15')
plt.title('Stay Requirement by Minimum Nights', fontsize='15')
```

Out[41]:
```
Text(0.5, 1.0, 'Stay Requirement by Minimum Nights')
```



Find the total numbers of Reviews and Maximum Reviews by Each Neighborhood Group.

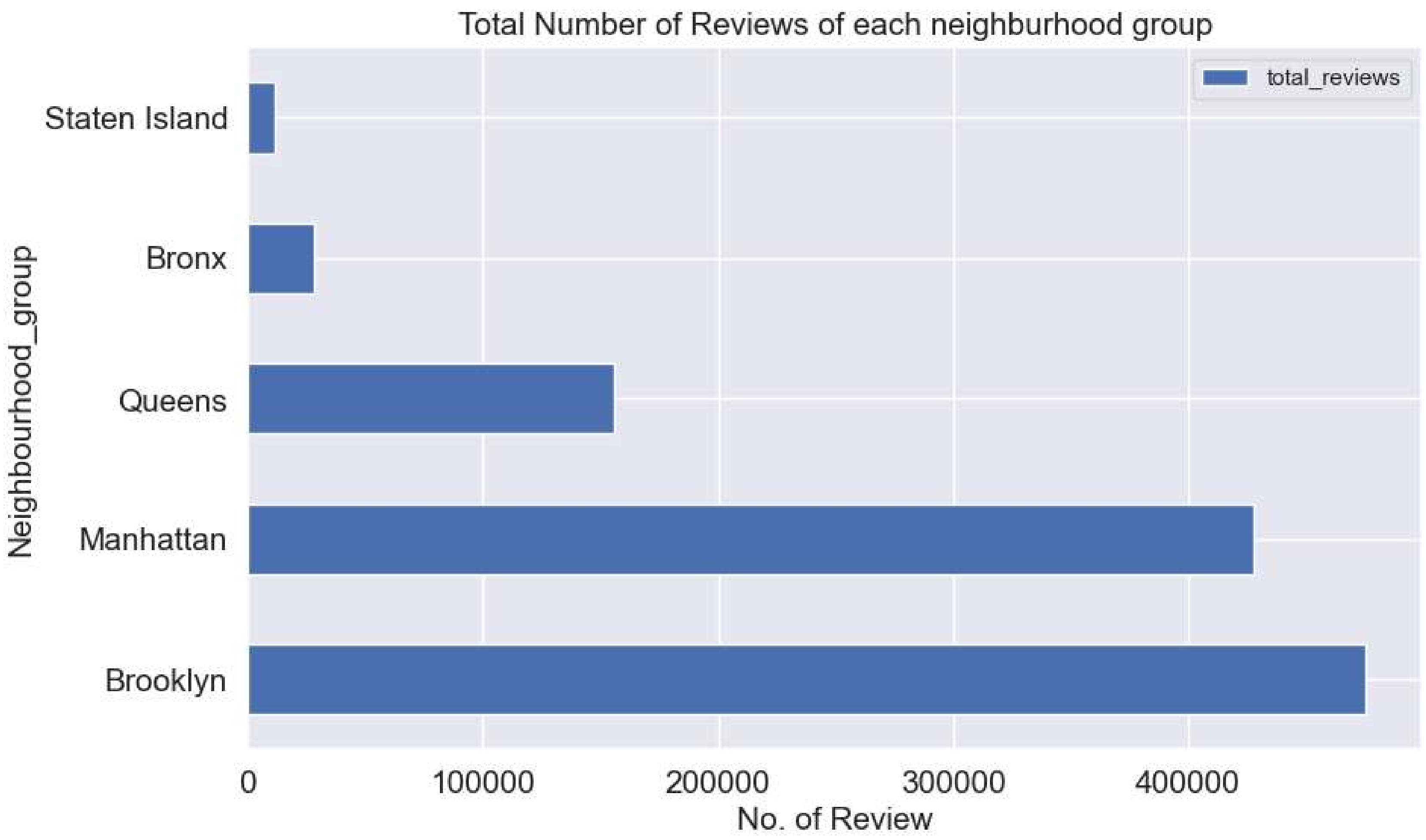
```
In [42]: review=airbnb.groupby(['neighbourhood_group'])
         total=review[['total_reviews']].sum()
         result=total.sort_values(['total_reviews'],ascending=False)
         result=result.head()
         result
```

```
Out[42]:
                         total_reviews
    neighbourhood_group
               Brooklyn         475936
              Manhattan         428128
                 Queens         155719
                  Bronx          28185
          Staten Island          11536
```

Total Number of Reviews of each neighburhood group by using barh plot

In [43]:
```
result.plot(kind='barh',figsize=(10,6),fontsize=15)
plt.title('Total Number of Reviews of each neighburhood group',fontsize=15)
plt.xlabel('No. of Review',fontsize=15)
plt.ylabel('Neighbourhood_group',fontsize=15)
```

Out[43]:
```
Text(0, 0.5, 'Neighbourhood_group')
```



Find Most reviewed room type in Neighborhood groups per month

In [44]:

```
airbnb.columns
```

Out[44]:

```
Index(['listing_id', 'listing_name', 'host_id', 'host_name',
       'neighbourhood_group', 'neighbourhood', 'latitude', 'longitude',
       'room_type', 'price', 'minimum_nights', 'total_reviews',
       'reviews_per_month', 'Host_listing_count', 'availability_365'],
      dtype='object')
```

In [45]:

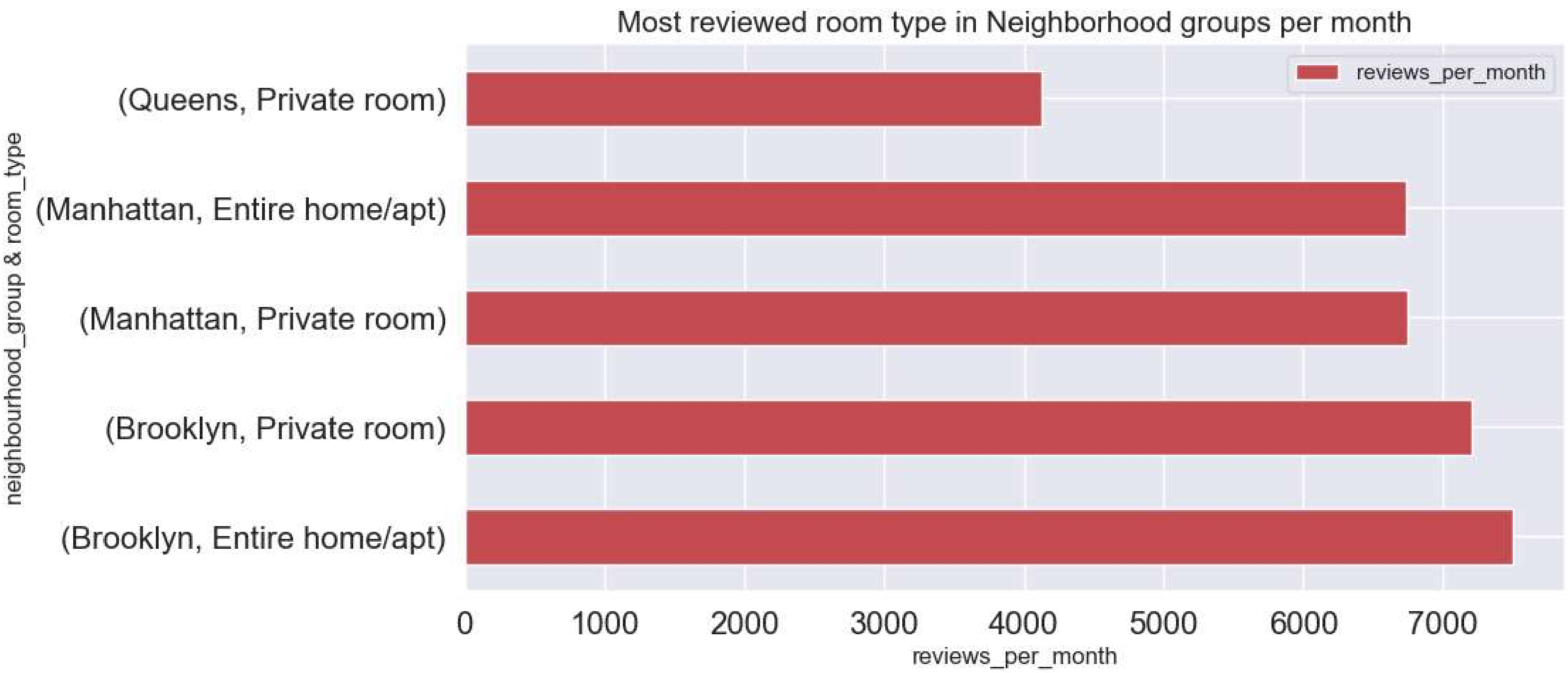
```
room=airbnb.groupby(['neighbourhood_group','room_type'])
review=room[['reviews_per_month']].sum()
most_review=review.sort_values(['reviews_per_month'],ascending=False)
most_review=most_review.head()
most_review
```

Out[45]:

```
                                          reviews_per_month
neighbourhood_group         room_type
          Brooklyn    Entire home/apt                  7498
                         Private room                  7204
         Manhattan       Private room                  6752
                      Entire home/apt                  6742
            Queens       Private room                  4122
```

Most reviewed room type in Neighborhood groups per month using barh plot

```
In [46]: most_review.plot(kind='barh',figsize=(10,5),color='r',fontsize=16)
         plt.title('Most reviewed room type in Neighborhood groups per month',fontsize=15)
         plt.xlabel('reviews_per_month')
         plt.ylabel('neighbourhood_group & room_type')

Out[46]: Text(0, 0.5, 'neighbourhood_group & room_type')
```



Find Best location listing/property location for travelers

```
In [47]: airbnb.columns

Out[47]: Index(['listing_id', 'listing_name', 'host_id', 'host_name',
                'neighbourhood_group', 'neighbourhood', 'latitude', 'longitude',
                'room_type', 'price', 'minimum_nights', 'total_reviews',
                'reviews_per_month', 'Host_listing_count', 'availability_365'],
               dtype='object')
```

In [48]:

```
group=airbnb.groupby(['neighbourhood'])
location=group[['minimum_nights']].max()
best_location=location.sort_values(['minimum_nights'],ascending=False)
best_location=best_location.head()
best_location
```

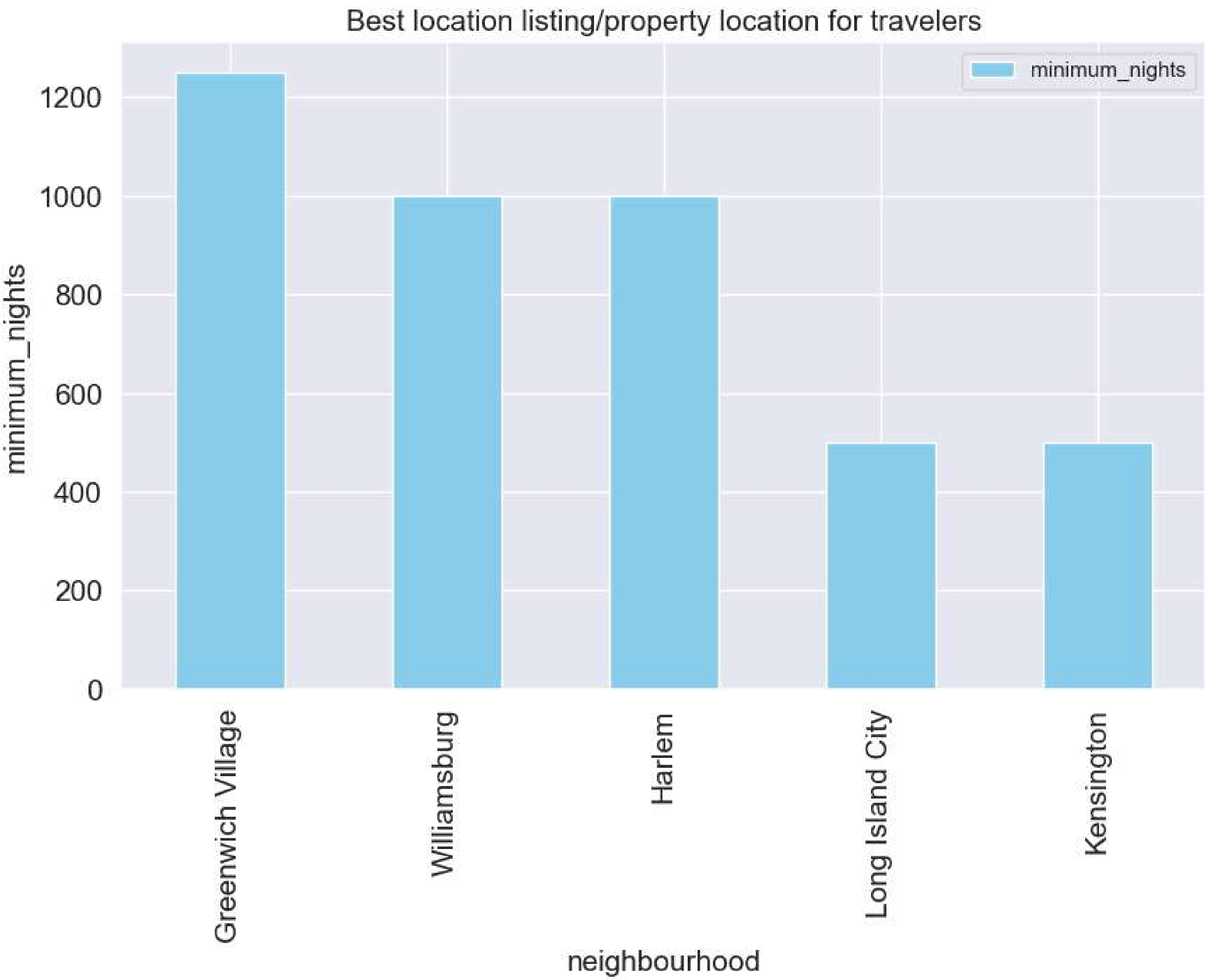
Out[48]:

	minimum_nights
neighbourhood	
Greenwich Village	1250
Williamsburg	999
Harlem	999
Long Island City	500
Kensington	500

Best location listing/property location for travelers using bar plot

```
In [49]: best_location.plot(kind='bar',figsize=(10,6),color='skyblue',fontsize=15)
         plt.title('Best location listing/property location for travelers',fontsize=15)
         plt.xlabel('neighbourhood',fontsize=15)
         plt.ylabel('minimum_nights',fontsize=15)
```

```
Out[49]: Text(0, 0.5, 'minimum_nights')
```


Best location listing/property location for travelers
minimum_nights
1200
1000
800
600
400
200
0
minimum_nights
Greenwich Village
Williamsburg
Harlem
Long Island City
Kensington
neighbourhood

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```