Airbnb Booking in New York (2019)

Understanding The Dataset Variables:

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

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.

number_of_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.

Host_listings_count : This is the total number of listings that the host has on Airbnb.

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

Importing Needed Libraries

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

import warnings
warnings.filterwarnings('ignore')
```

Load Dataset

```
df = pd.read_csv("Airbnb NYC 2019.csv")
df.head()
```

```
     id                                              name  host_id  \
0  2539                Clean & quiet apt home by the park     2787
1  2595                             Skylit Midtown Castle     2845
2  3647               THE VILLAGE OF HARLEM....NEW YORK !     4632
3  3831                   Cozy Entire Floor of Brownstone     4869
4  5022  Entire Apt: Spacious Studio/Loft by central park     7192

     host_name neighbourhood_group neighbourhood  latitude  longitude
\
0         John            Brooklyn    Kensington  40.64749  -73.97237

1     Jennifer           Manhattan       Midtown  40.75362  -73.98377

2    Elisabeth           Manhattan        Harlem  40.80902  -73.94190

3  LisaRoxanne            Brooklyn  Clinton Hill  40.68514  -73.95976

4        Laura           Manhattan   East Harlem  40.79851  -73.94399


         room_type  price  minimum_nights  number_of_reviews
last_review  \
0     Private room    149               1                  9  2018-10-
19
1  Entire home/apt    225               1                 45  2019-05-
21
2     Private room    150               3                  0
NaN
3  Entire home/apt     89               1                270  2019-07-
05
4  Entire home/apt     80              10                  9  2018-11-
19

   reviews_per_month  calculated_host_listings_count  availability_365

0               0.21                               6               365

1               0.38                               2               355

2                NaN                               1               365

3               4.64                               1               194

4               0.10                               1                 0
```

Data cleaning and Exploration

```
df.shape
```

```
(48895, 16)
```

```
df.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
```

```
#Convert from String to date
df['last_review']=df['last_review'].astype('datetime64[ns]')
df['last_review'] = [time.date() for time in df['last_review']]
df['last_review']
```

```
0        2018-10-19
1        2019-05-21
2               NaT
3        2019-07-05
4        2018-11-19
            ...    
48890           NaT
48891           NaT
48892           NaT
48893           NaT
48894           NaT
Name: last_review, Length: 48895, dtype: object
```

```
# fill the nulls
df['name'].fillna('null',inplace= True)
df['host_name'].fillna('null',inplace= True)
df['last_review'].fillna('null',inplace= True)
df['reviews_per_month'].fillna(pd.NA,inplace= True)

df.nunique()
```

```
id                                48895
name                              47906
host_id                           37457
host_name                         11453
neighbourhood_group                   5
neighbourhood                       221
latitude                          19048
longitude                         14718
room_type                             3
price                               674
minimum_nights                      109
number_of_reviews                   394
last_review                        1765
reviews_per_month                   937
calculated_host_listings_count       47
availability_365                    366
dtype: int64
```

```
df.dtypes
```

```
id                                 int64
name                              object
host_id                            int64
host_name                         object
neighbourhood_group               object
neighbourhood                     object
latitude                         float64
longitude                        float64
room_type                         object
price                              int64
minimum_nights                     int64
number_of_reviews                  int64
last_review                       object
reviews_per_month                float64
calculated_host_listings_count     int64
availability_365                   int64
dtype: object
```

```
# Check for the duplicates rows
print('number of duplicates: ', df.duplicated().sum())
df[df.duplicated()]
```

```
number of duplicates:  0
```

```
Empty DataFrame
Columns: [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]
Index: []
```

```
df.describe().T
```

```
                                  count          mean           std  \
id                              48895.0  1.901714e+07  1.098311e+07
host_id                         48895.0  6.762001e+07  7.861097e+07
latitude                        48895.0  4.072895e+01  5.453008e-02
longitude                       48895.0 -7.395217e+01  4.615674e-02
price                           48895.0  1.527207e+02  2.401542e+02
minimum_nights                  48895.0  7.029962e+00  2.051055e+01
number_of_reviews               48895.0  2.327447e+01  4.455058e+01
reviews_per_month               38843.0  1.373221e+00  1.680442e+00
calculated_host_listings_count  48895.0  7.143982e+00  3.295252e+01
availability_365                48895.0  1.127813e+02  1.316223e+02

                                       min           25%           50%
\
id                              2539.00000  9.471945e+06  1.967728e+07

host_id                         2438.00000  7.822033e+06  3.079382e+07

latitude                          40.49979  4.069010e+01  4.072307e+01

longitude                        -74.24442 -7.398307e+01 -7.395568e+01

price                              0.00000  6.900000e+01  1.060000e+02

minimum_nights                     1.00000  1.000000e+00  3.000000e+00

number_of_reviews                  0.00000  1.000000e+00  5.000000e+00

reviews_per_month                  0.01000  1.900000e-01  7.200000e-01

calculated_host_listings_count     1.00000  1.000000e+00  1.000000e+00

availability_365                   0.00000  0.000000e+00  4.500000e+01


                                       75%           max
id                              2.915218e+07  3.648724e+07
host_id                         1.074344e+08  2.743213e+08
latitude                        4.076311e+01  4.091306e+01
longitude                      -7.393627e+01 -7.371299e+01
price                           1.750000e+02  1.000000e+04
minimum_nights                  5.000000e+00  1.250000e+03
```

```
number_of_reviews                2.400000e+01  6.290000e+02
reviews_per_month                2.020000e+00  5.850000e+01
calculated_host_listings_count   2.000000e+00  3.270000e+02
availability_365                 2.270000e+02  3.650000e+02
```

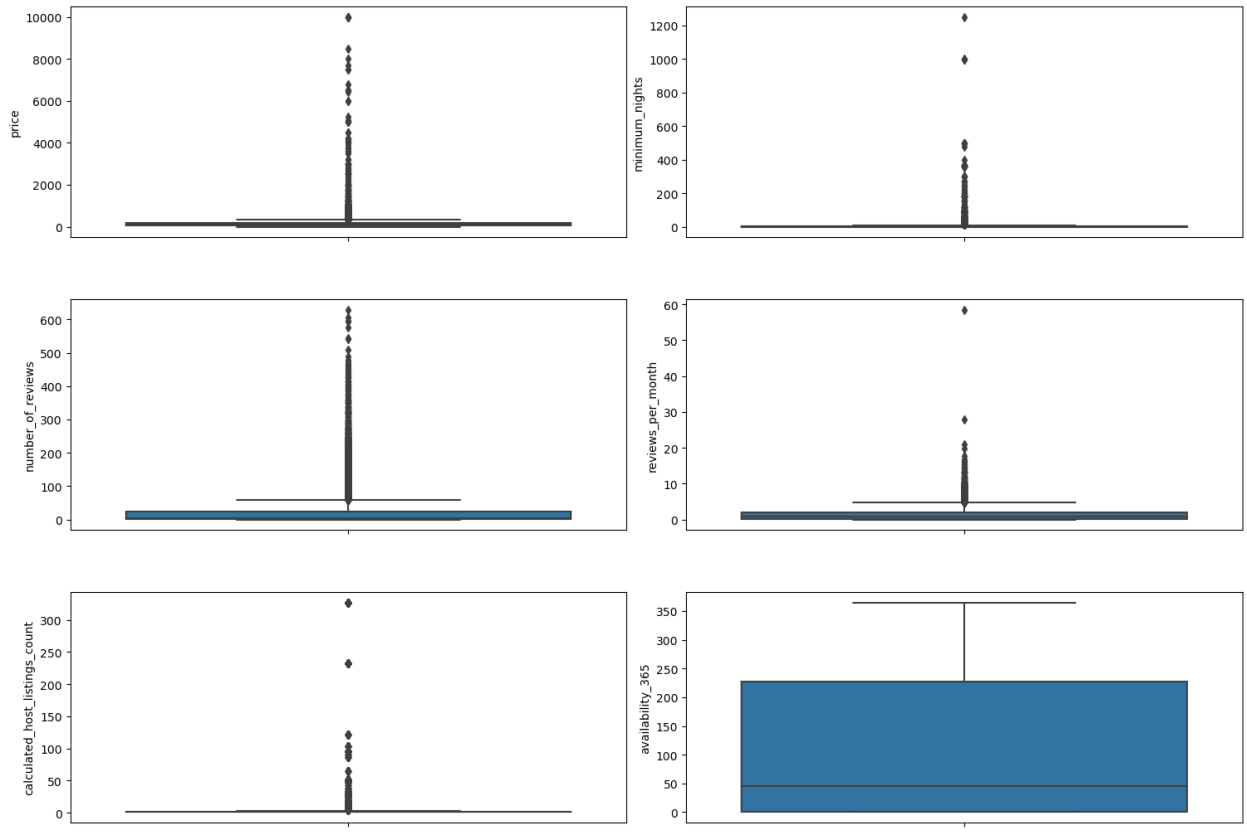
```
df.describe(include = 'object').T
```

```
                     count unique              top   freq
name                 48895  47906   Hillside Hotel     18
host_name            48895  11453          Michael    417
neighbourhood_group  48895      5        Manhattan  21661
neighbourhood        48895    221     Williamsburg   3920
room_type            48895      3  Entire home/apt  25409
last_review          48895   1765             null  10052
```

```
df.columns
```

```
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')
```

```
# Check for the outliers by using box plot

fig , axs = plt.subplots(ncols=2 , nrows=3 , figsize=(15,10))
index=0
axs=axs.flatten()
data
=df[['price','minimum_nights','number_of_reviews','reviews_per_month',
'calculated_host_listings_count','availability_365']]
for k,v in data.items():
    sns.boxplot(y=k, data=df , ax=axs[index])
    index+=1
plt.tight_layout(pad=0.4 , w_pad=0.5, h_pad=5)
```



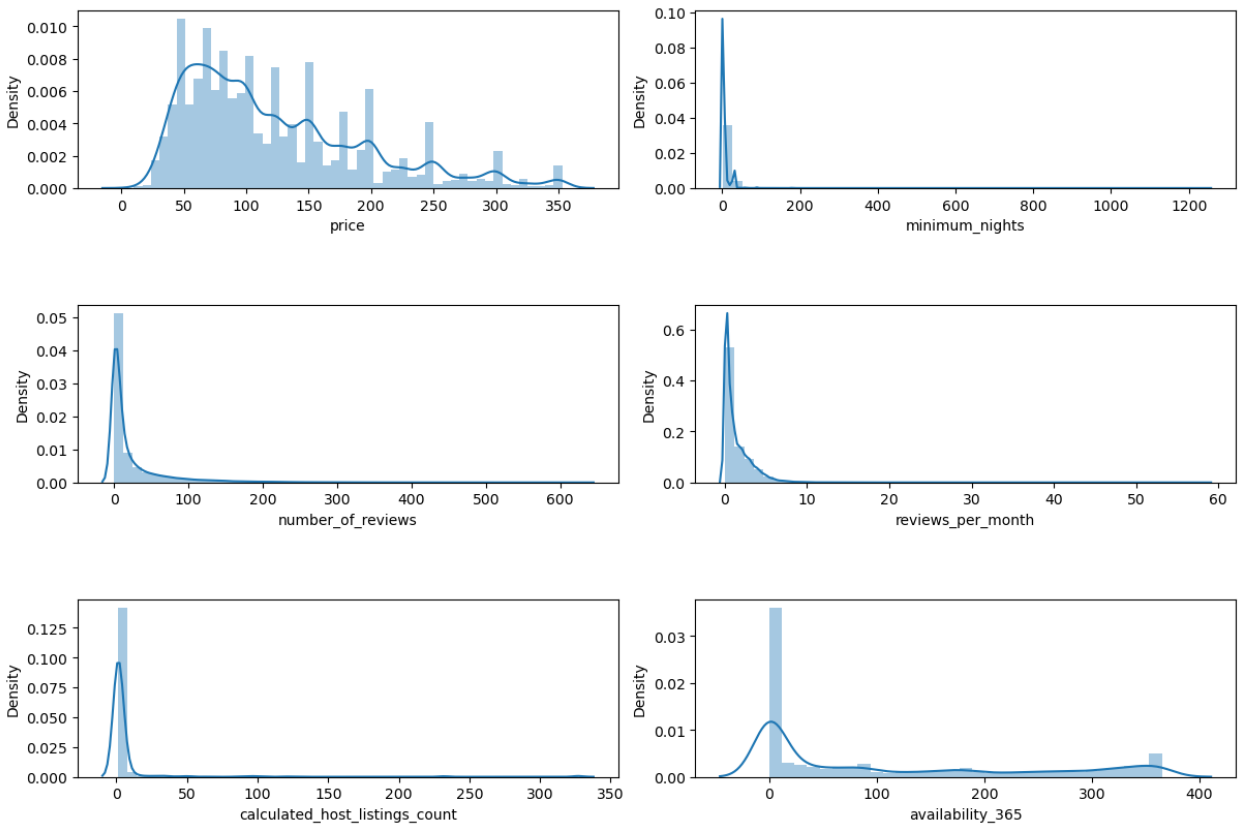

```
# Number of outliers per columns
for k, v in data.items():
    q1 = v.quantile(0.25)
    q3 = v.quantile(0.75)
    irq = q3 - q1
    v_col = v[(v <= q1 - 1.5 * irq) | (v >= q3 + 1.5 * irq)]
    perc = np.shape(v_col)[0] * 100.0 / np.shape(data)[0]
    print("Column %s outliers = %.2f%%" % (k, perc))
```

```
Column price outliers = 6.09%
Column minimum_nights outliers = 13.58%
Column number_of_reviews outliers = 12.31%
Column reviews_per_month outliers = 3.67%
Column calculated_host_listings_count outliers = 14.48%
Column availability_365 outliers = 0.00%
```

```
# Remove the outlier of price column within 95% percentile

q_low = df["price"].quantile(0)
q_hi  = df["price"].quantile(0.95)

df = df[(df["price"] < q_hi) & (df["price"] > q_low)]
df.shape
```

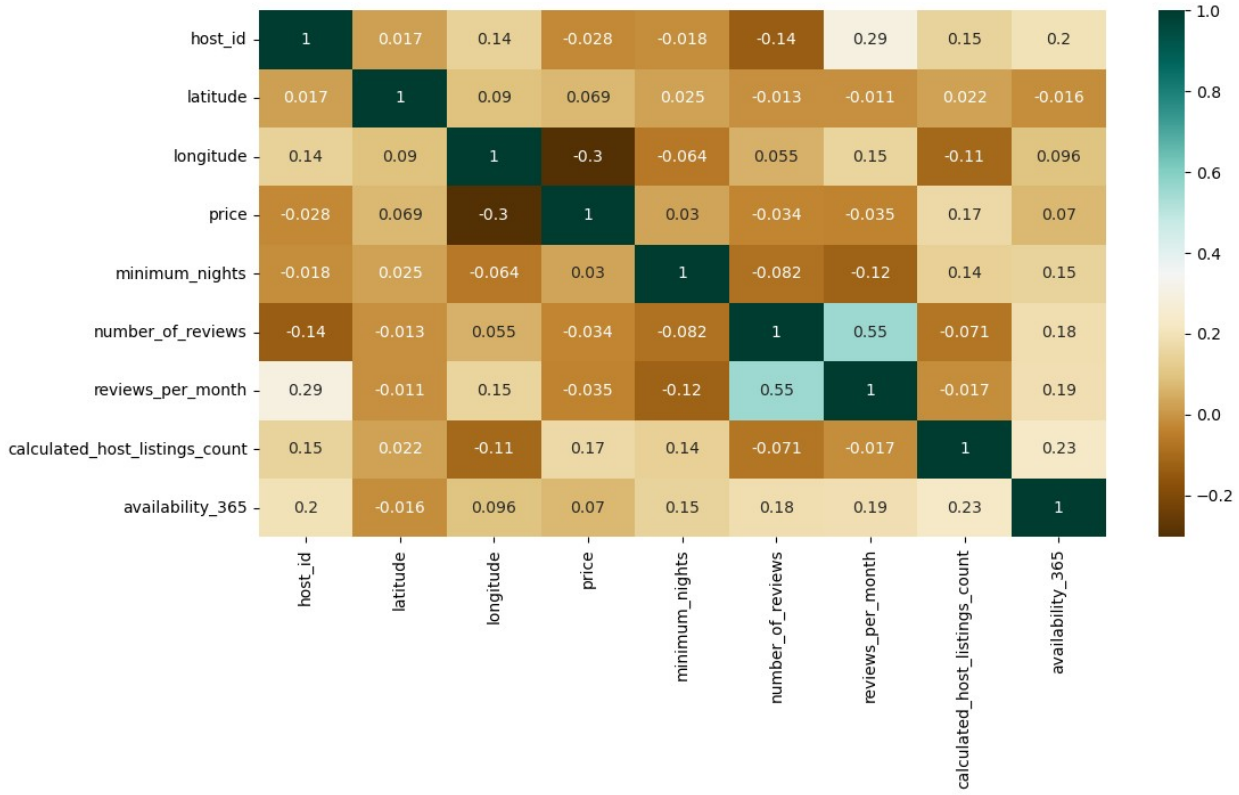
```
(46433, 16)
```

```
# The distribution of the Data

fig , axs = plt.subplots(ncols=2 , nrows=3 , figsize=(12,8))
index=0
axs=axs.flatten()
data
=df[['price','minimum_nights','number_of_reviews','reviews_per_month',
'calculated_host_listings_count','availability_365']]
for k,v in data.items():
    sns.distplot(v , ax=axs[index])
    index+=1
plt.tight_layout(pad=0.4 , w_pad=0.5, h_pad=5)
```




```
# Correlation Heatmap

data
=df[['host_id','latitude','longitude','price','minimum_nights','number
_of_reviews',

'reviews_per_month','calculated_host_listings_count','availability_365
']]
corr = data.corr()
plt.figure(figsize=(12,6))
```

```
sns.heatmap(corr, cmap='BrBG',annot=True)

plt.show()
```



Exploratory Data Analysis

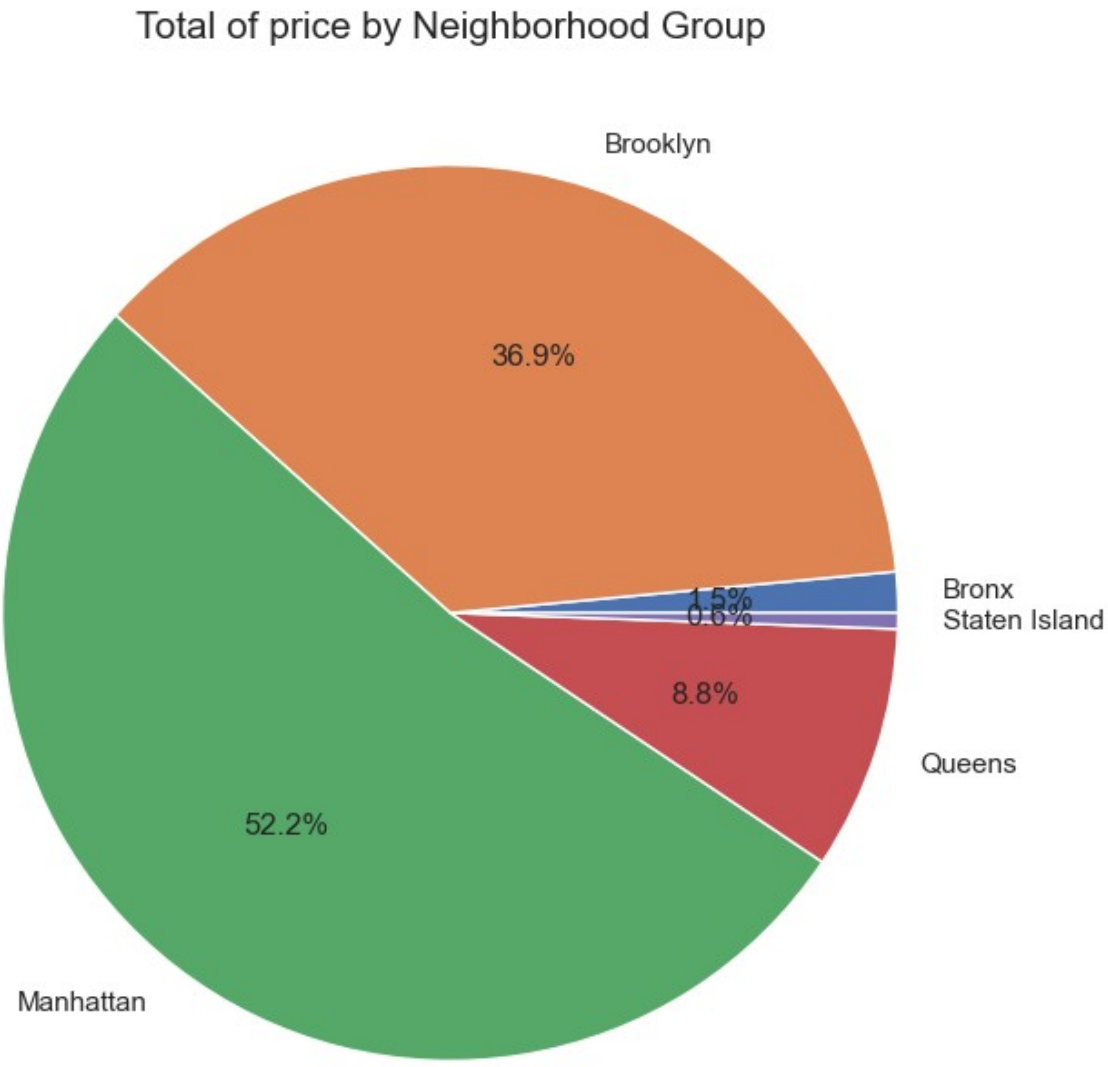
Total price for each neighbourhood group

```
Result=df.groupby(['neighbourhood_group'])['price'].sum()
Result.reset_index()
```

	neighbourhood_group	price
0	Bronx	83831
1	Brooklyn	2102762
2	Manhattan	2970234
3	Queens	501558
4	Staten Island	32571

```
Price_by_neighbourhood_group = df.groupby("neighbourhood_group")
["price"].sum()


plt.pie(Price_by_neighbourhood_group,
labels=Price_by_neighbourhood_group.index, autopct='%1.1f%%')
plt.title("Total of price by Neighborhood Group", fontsize='15')
```

```
plt.show()
```

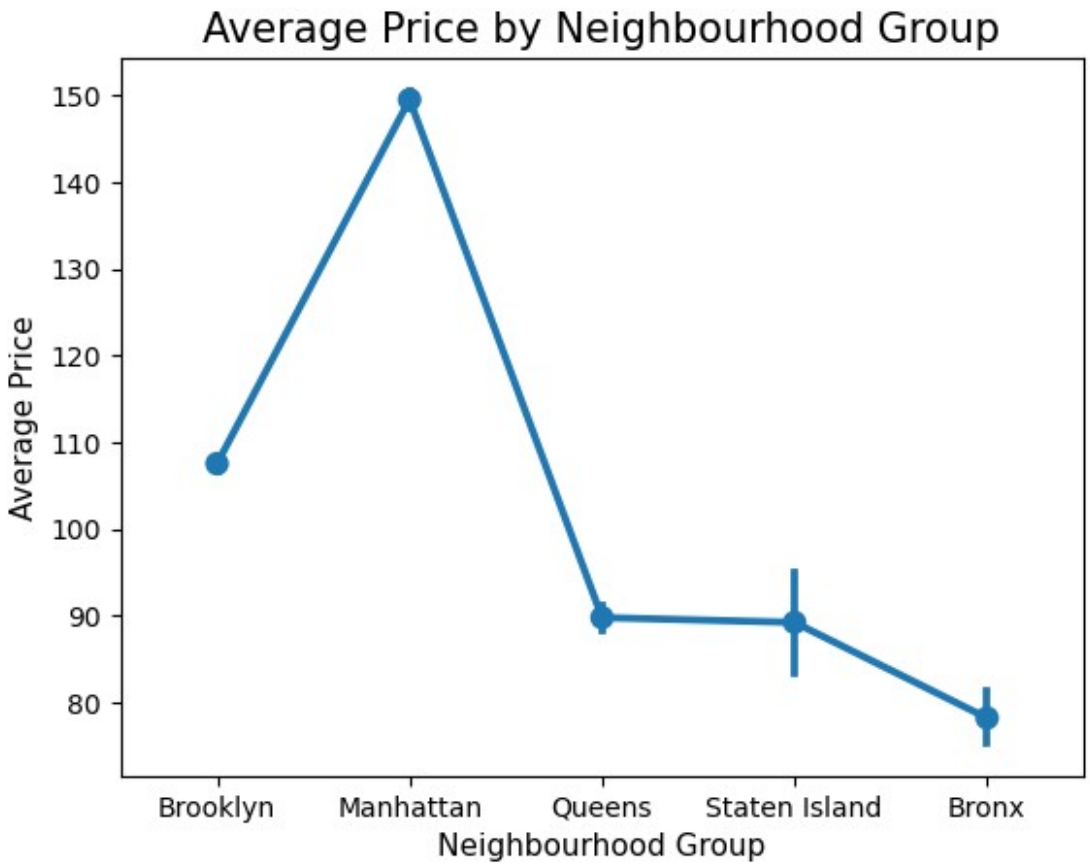


Average price for neighbourhood group

```
df.groupby(['neighbourhood_group'])['price'].mean().reset_index()
```

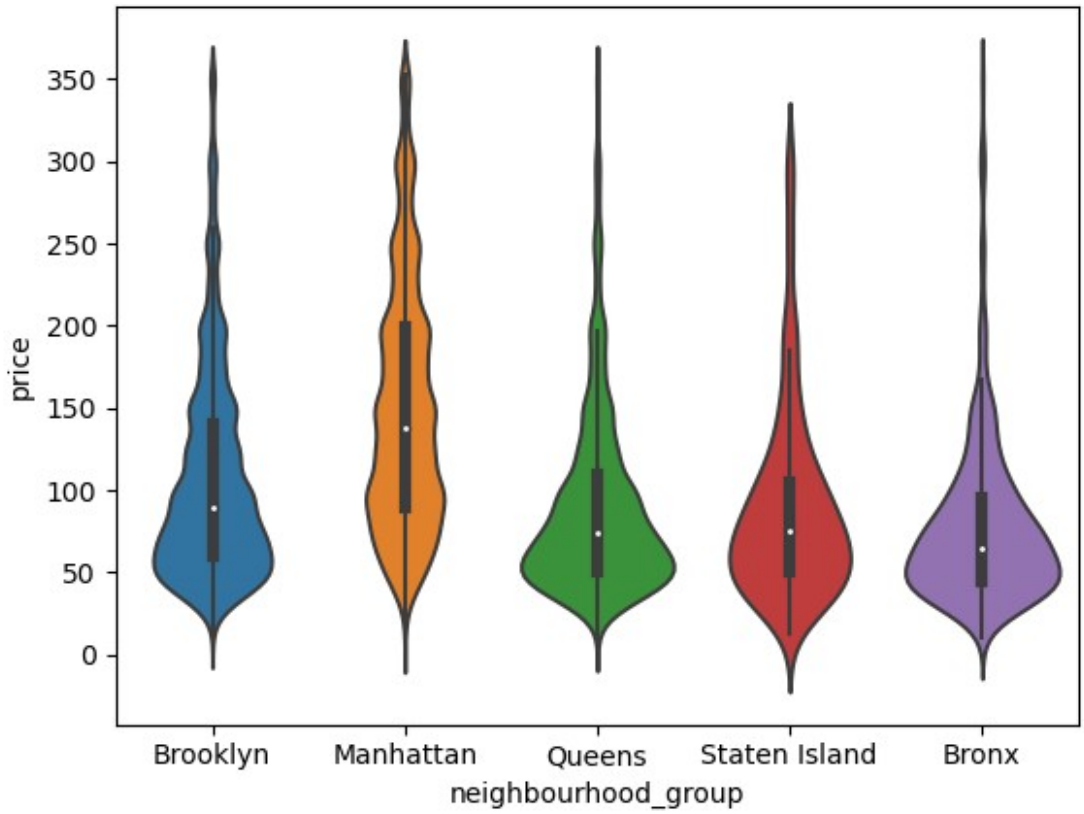
	neighbourhood_group	price
0	Bronx	78.200560
1	Brooklyn	107.552657
2	Manhattan	149.566141
3	Queens	89.788400
4	Staten Island	89.235616

```
sns.pointplot(x = 'neighbourhood_group', y='price', data=df, estimator
= np.mean)

plt.xlabel('Neighbourhood Group',fontsize=11)
plt.ylabel('Average Price',fontsize=11)
plt.title('Average Price by Neighbourhood Group',fontsize=15)

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



Price Distribution for Each Neighborhood Group

```
ax= sns.violinplot(x='neighbourhood_group',y='price',data= df)
```

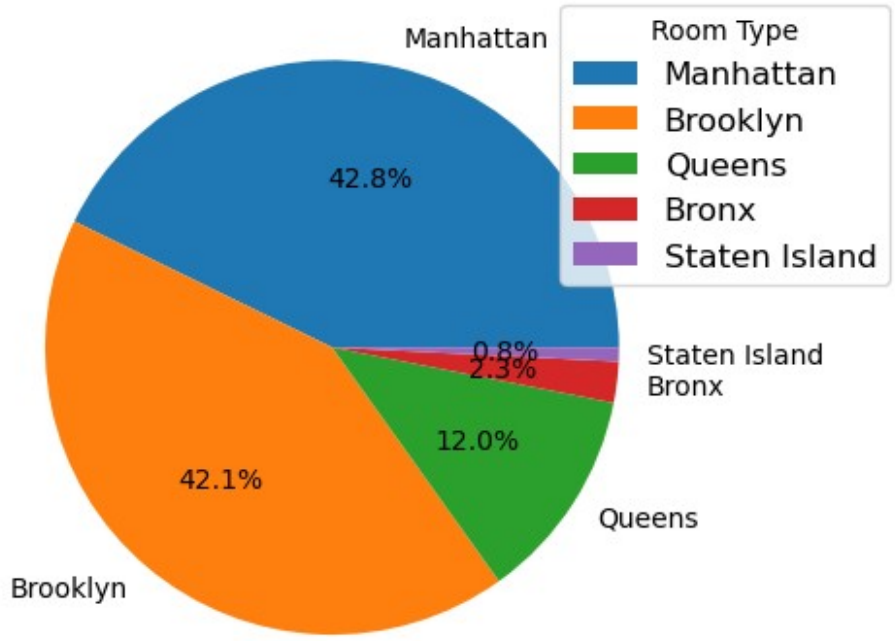


Number of Listings for each neighbourhood group

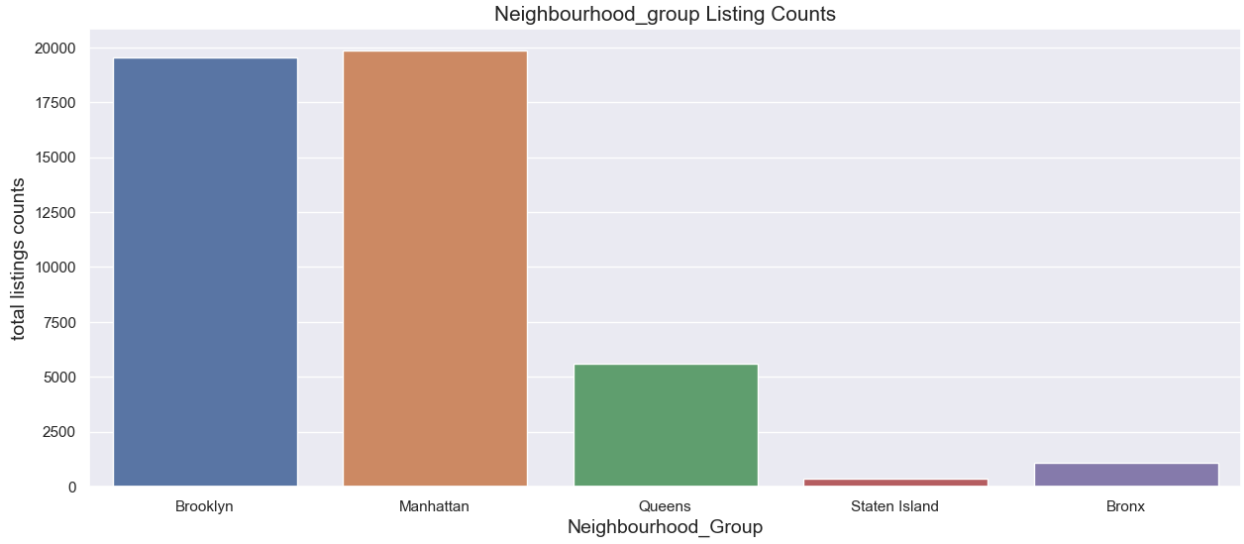
```
Result=df.groupby(['neighbourhood_group'])
['calculated_host_listings_count'].count()
Result.reset_index()
```

	neighbourhood_group	calculated_host_listings_count
0	Bronx	1072
1	Brooklyn	19551
2	Manhattan	19859
3	Queens	5586
4	Staten Island	365

```
room_type_counts = df['neighbourhood_group'].value_counts()
labels = room_type_counts.index
sizes = room_type_counts.values


plt.pie(sizes, labels=labels, autopct='%1.1f%%')

plt.legend(title='Room Type', bbox_to_anchor=(0.8, 0, 0.5, 1),
fontsize='12')

plt.show()
```




```
plt.figure(figsize=(15,6))
sns.countplot(df,x='neighbourhood_group')
plt.title('Neighbourhood_group Listing Counts',fontsize=15)
plt.xlabel('Neighbourhood_Group', fontsize=14)
plt.ylabel('total listings counts', fontsize=14)

Text(0, 0.5, 'total listings counts')
```

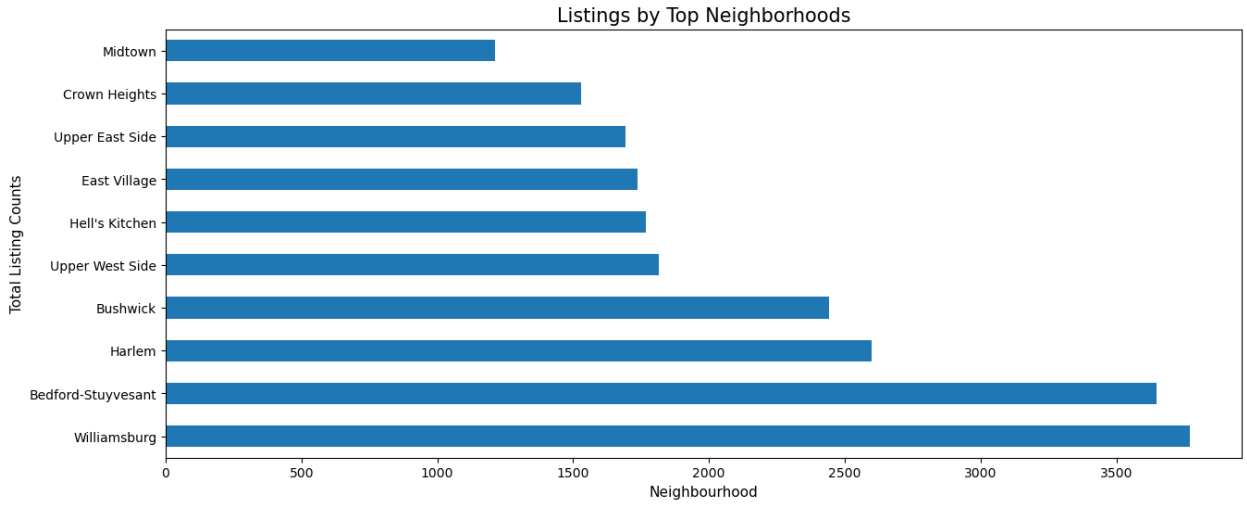


Number of Listings for Top 10 neighbourhoods

```
df['neighbourhood'].value_counts()[:10].reset_index()
```

```
        neighbourhood  count
0        Williamsburg   3771
1  Bedford-Stuyvesant   3647
2             Harlem   2599
3           Bushwick   2442
4     Upper West Side   1815
5      Hell's Kitchen   1769
6        East Village   1737
7     Upper East Side   1692
8       Crown Heights   1528
9             Midtown   1211
```

```
Top_10_Neighbourhoods= df['neighbourhood'].value_counts().nlargest(10)
Top_10_Neighbourhoods.plot(kind='barh', figsize=(15, 6))

plt.xlabel('Neighbourhood', fontsize=11)
plt.ylabel('Total Listing Counts', fontsize=11)
plt.title('Listings by Top Neighborhoods', fontsize=15)

Text(0.5, 1.0, 'Listings by Top Neighborhoods')
```



Number of Listings for Top 10 Hosts

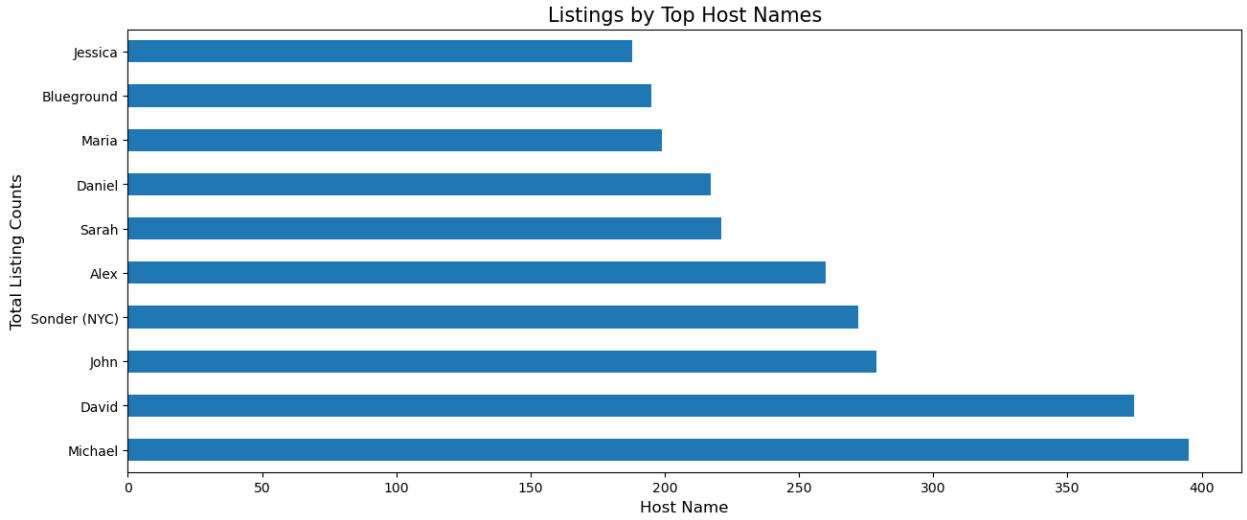
```
df['host_name'].value_counts()[:10].reset_index()
```

```
      host_name  count
0       Michael    395
1         David    375
2          John    279
3  Sonder (NYC)    272
4          Alex    260
5         Sarah    221
```

```
6         Daniel    217
7          Maria    199
8     Blueground    195
9        Jessica    188
```

```
Top_10_Neighbourhoods= df['host_name'].value_counts().nlargest(10)
Top_10_Neighbourhoods.plot(kind='barh', figsize=(15, 6))

plt.xlabel('Host Name', fontsize=12)
plt.ylabel('Total Listing Counts', fontsize=12)
plt.title('Listings by Top Host Names', fontsize=15)
```

```
Text(0.5, 1.0, 'Listings by Top Host Names')
```

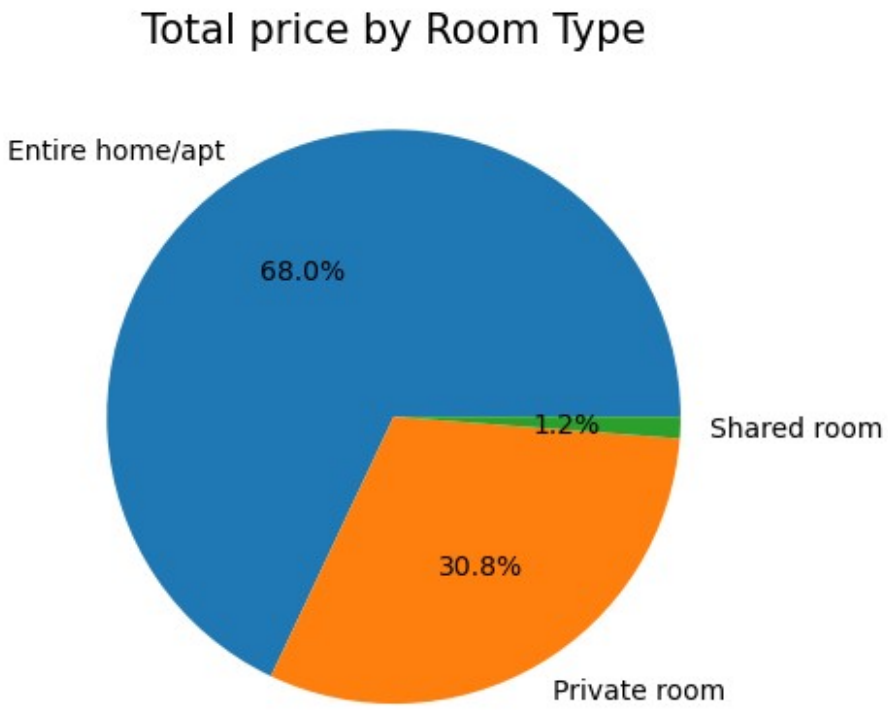


Total price by room type

```
df.groupby(['room_type'])['price'].sum().reset_index()
```

```
          room_type    price
0  Entire home/apt  3867615
1     Private room  1754478
2      Shared room    68863
```

```
Price_by_room_type = df.groupby("room_type")["price"].sum()


plt.pie(Price_by_room_type, labels=Price_by_room_type.index,
autopct='%1.1f%%')
plt.title("Total price by Room Type", fontsize='15')


plt.show()
```



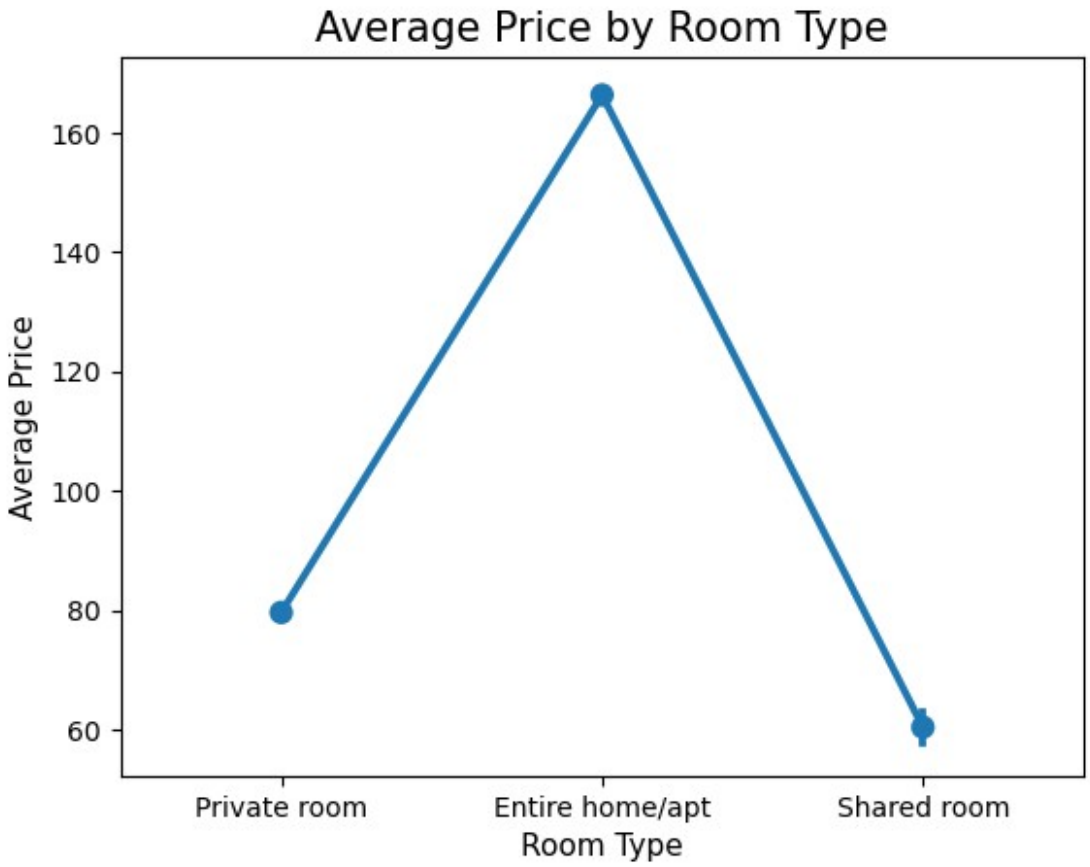
Average price for each room_type

```
df.groupby(['room_type'])['price'].mean().reset_index()
```

```
         room_type       price
0  Entire home/apt  166.298964
1     Private room   79.618715
2      Shared room   60.406140
```

```
sns.pointplot(x = 'room_type', y='price', data=df, estimator =
np.mean)

plt.xlabel('Room Type',fontsize=11)
plt.ylabel('Average Price',fontsize=11)
plt.title('Average Price by Room Type',fontsize=15)
```

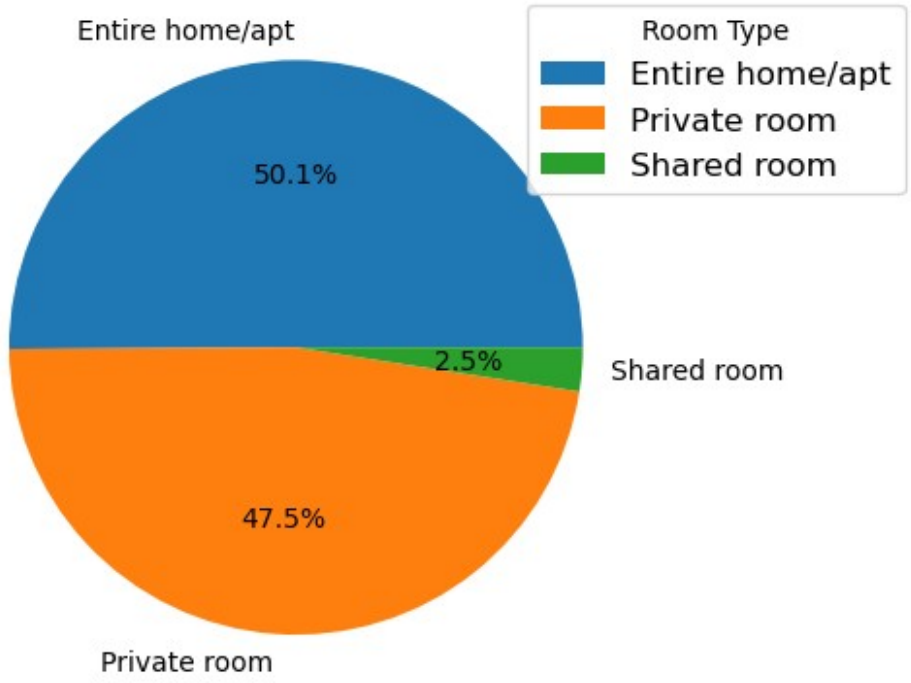
```
Text(0.5, 1.0, 'Average Price by Room Type')
```




```
#### **Number of Listings for each room type**

df.groupby(['room_type'])['id'].count().reset_index()
```

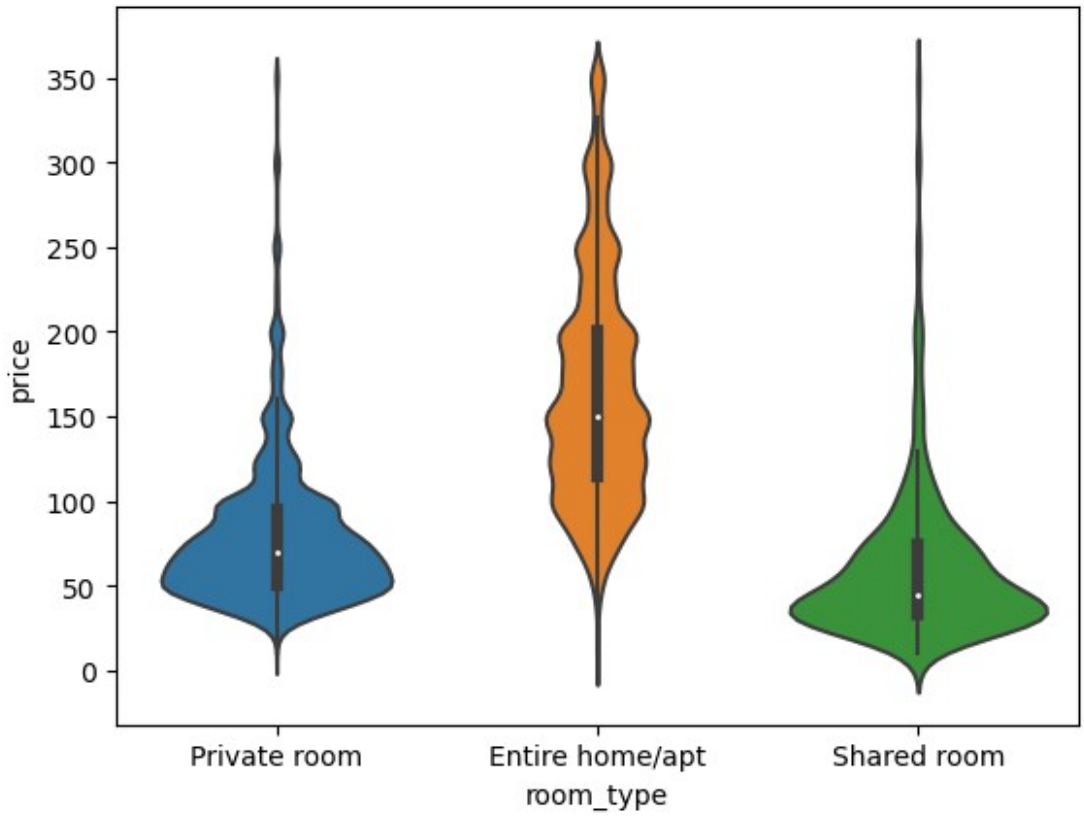
	room_type	id
0	Entire home/apt	23257
1	Private room	22036
2	Shared room	1140

```
room_type_counts = df['room_type'].value_counts()
labels = room_type_counts.index
sizes = room_type_counts.values


plt.pie(sizes, labels=labels, autopct='%1.1f%%')

plt.legend(title='Room Type', bbox_to_anchor=(0.8, 0, 0.5, 1),
fontsize='12')

plt.show()
```



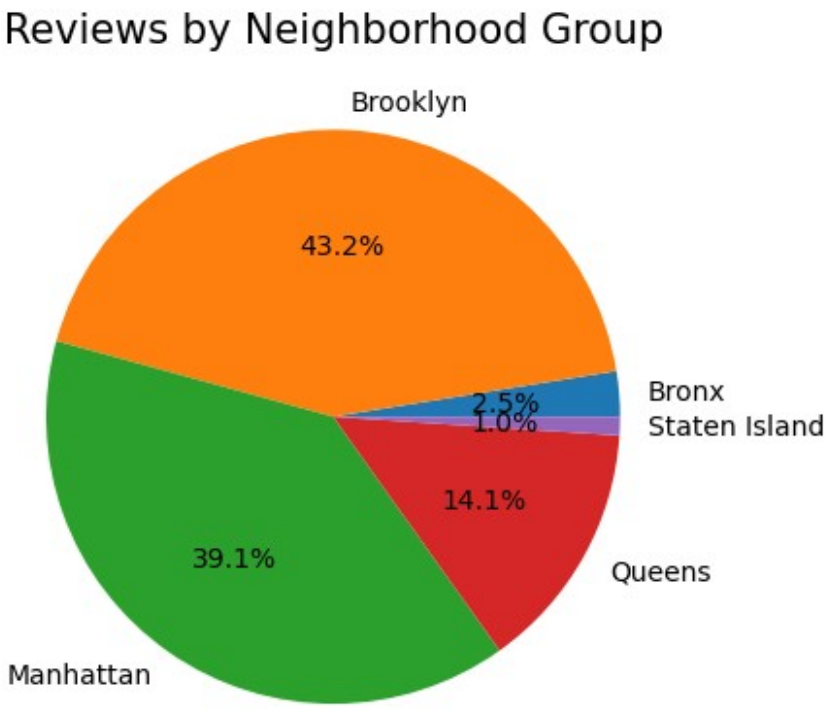
Price Distribution for Each room type

```
ax= sns.violinplot(x='room_type',y='price',data= df)
```



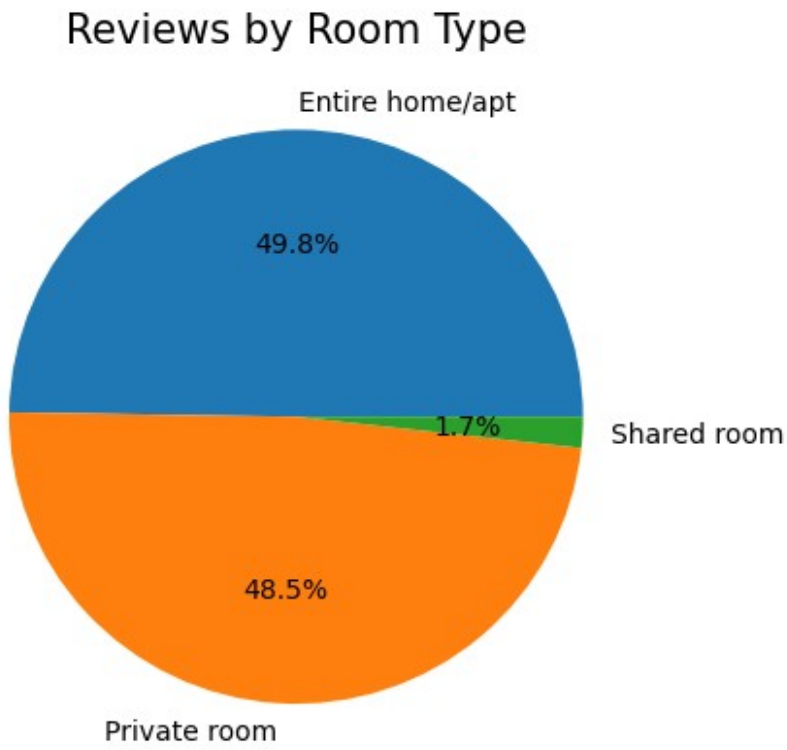
% of Reviews by Neighborhood Group

```
reviews_by_neighbourhood_group = df.groupby("neighbourhood_group")
["number_of_reviews"].sum()

plt.pie(reviews_by_neighbourhood_group,
labels=reviews_by_neighbourhood_group.index, autopct='%1.1f%%')
plt.title("Reviews by Neighborhood Group", fontsize='15')

plt.show()
```



% of Reviews by room type

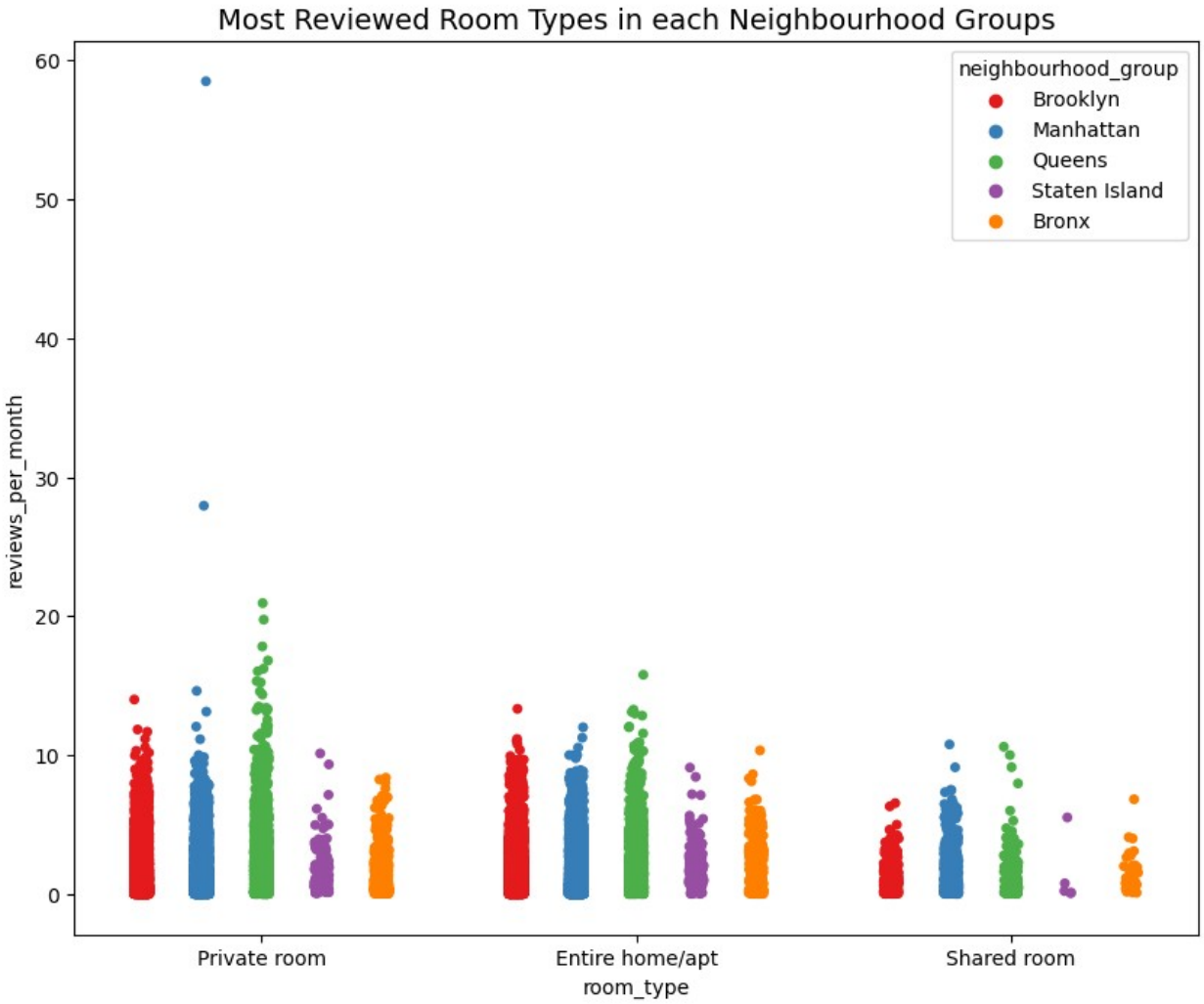
```
reviews_by_room_type = df.groupby("room_type")
["number_of_reviews"].sum()

plt.pie(reviews_by_room_type, labels=reviews_by_room_type.index,
autopct='%1.1f%%')
plt.title("Reviews by Room Type", fontsize='15')

plt.show()
```



Most reviewed room type per month in neighbourhood groups

```
f, ax = plt.subplots(figsize=(10, 8))

ax = sns.stripplot(x='room_type', y='reviews_per_month',
hue='neighbourhood_group', dodge=True, data=df, palette='Set1')

ax.set_title('Most Reviewed Room Types in each Neighbourhood Groups',
fontsize='14')
```

```
Text(0.5, 1.0, 'Most Reviewed Room Types in each Neighbourhood
Groups')
```



Pivot table for neighbourhood group per each room type

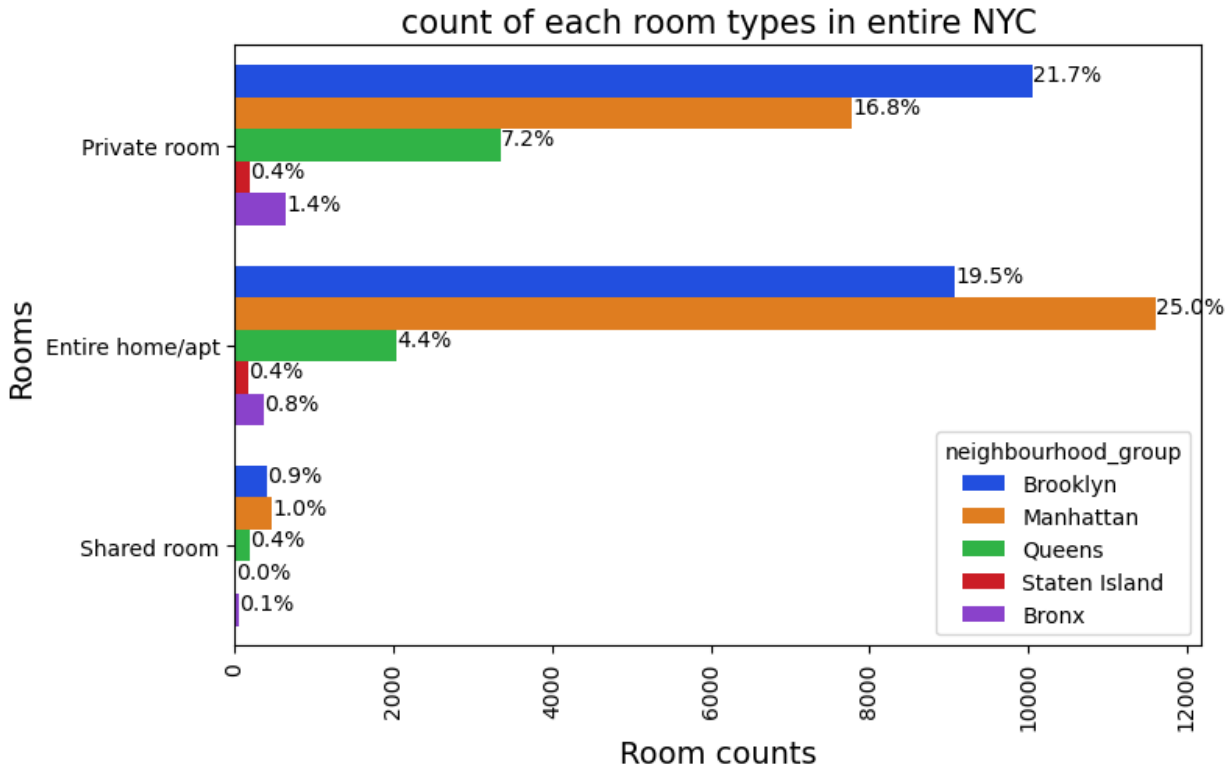
```
GroupBy= df.groupby(['neighbourhood_group','room_type'])
Data = pd.DataFrame({
    'Mean_Price': GroupBy['price'].mean(),
    'Sum_Price': GroupBy['price'].sum(),
    'Count': GroupBy['price'].count(),
})
Data
```

		Mean_Price	Sum_Price	Count
neighbourhood_group	room_type			
Bronx	Entire home/apt	114.144809	41777	366
	Private room	60.689335	39266	647
	Shared room	47.254237	2788	59
Brooklyn	Entire home/apt	151.227388	1372691	9077
	Private room	70.574069	710328	10065
	Shared room	48.271394	19743	409
Manhattan	Entire home/apt	186.236454	2162019	11609

```
                     Private room      99.153836     771516   7781
                     Shared room       78.249467      36699    469
Queens               Entire home/apt  132.933235     270785   2037
                     Private room      66.067660     221657   3355
                     Shared room       46.989691       9116    194
Staten Island        Entire home/apt  121.089286      20343    168
                     Private room      62.292553      11711    188
                     Shared room       57.444444        517      9
```

Number Of listings for each room types by thier neighbourhood groups

```
plt.rcParams['figure.figsize'] = (8, 5)

ax = sns.countplot(y='room_type', hue='neighbourhood_group', data=df,
palette='bright')

total = len(df['room_type'])


for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_width()/total)
        x = p.get_x() + p.get_width() + 0.02
        y = p.get_y() + p.get_height()/2
        ax.annotate(percentage, (x, y))


plt.title('count of each room types in entire NYC', fontsize='15')
plt.xlabel('Room counts', fontsize='14')
plt.xticks(rotation=90)
plt.ylabel('Rooms', fontsize='14')
plt.show()
```



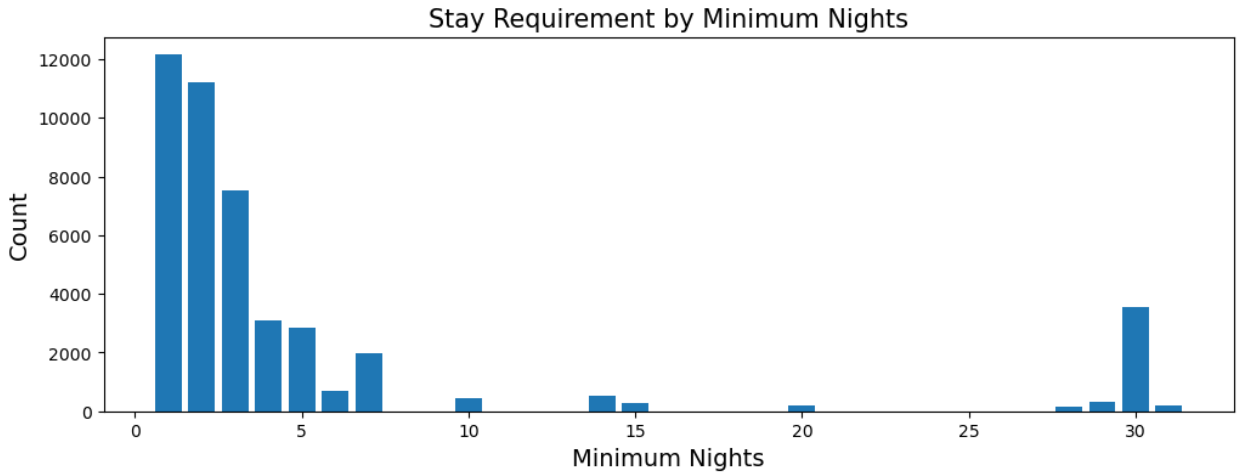
Stay Requirement counts by Minimum Nights

```
min_nights_count =
df.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
```

	minimum_nights	count
0	1	12148
1	2	11199
2	3	7507
3	30	3537
4	4	3107
5	5	2854
6	7	1975
7	6	694
8	14	543
9	10	464
10	29	328
11	15	272

```
12                20    216
13                31    191
14                28    174
```

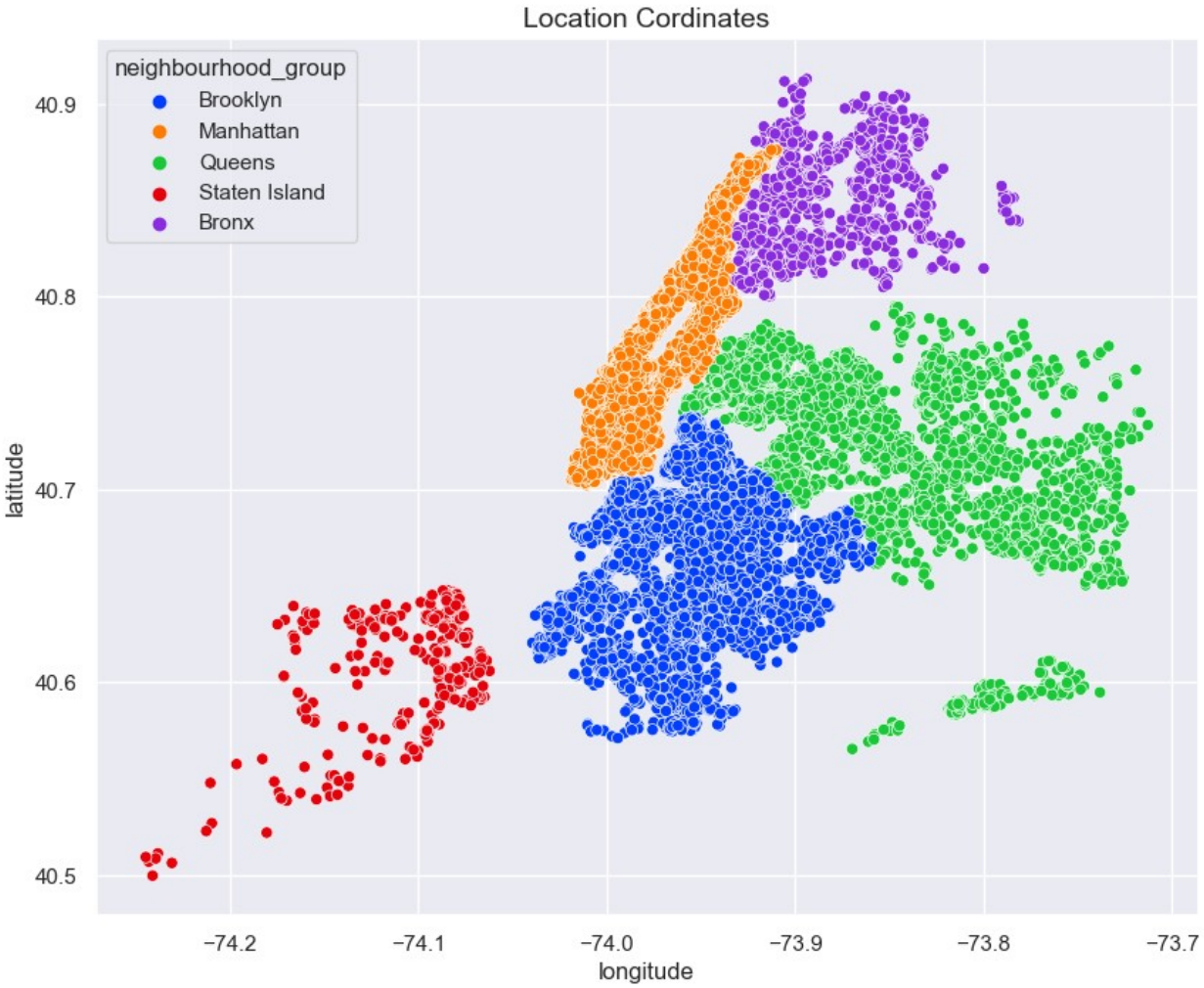
```
minimum_nights = min_nights_count['minimum_nights']
count = min_nights_count['count']


plt.figure(figsize=(12, 4))


plt.bar(minimum_nights, count)


plt.xlabel('Minimum Nights', fontsize='14')
plt.ylabel('Count', fontsize='14')
plt.title('Stay Requirement by Minimum Nights', fontsize='15')


plt.show()
```



neighbourhood groups and room types by using latitude and longitude as a map

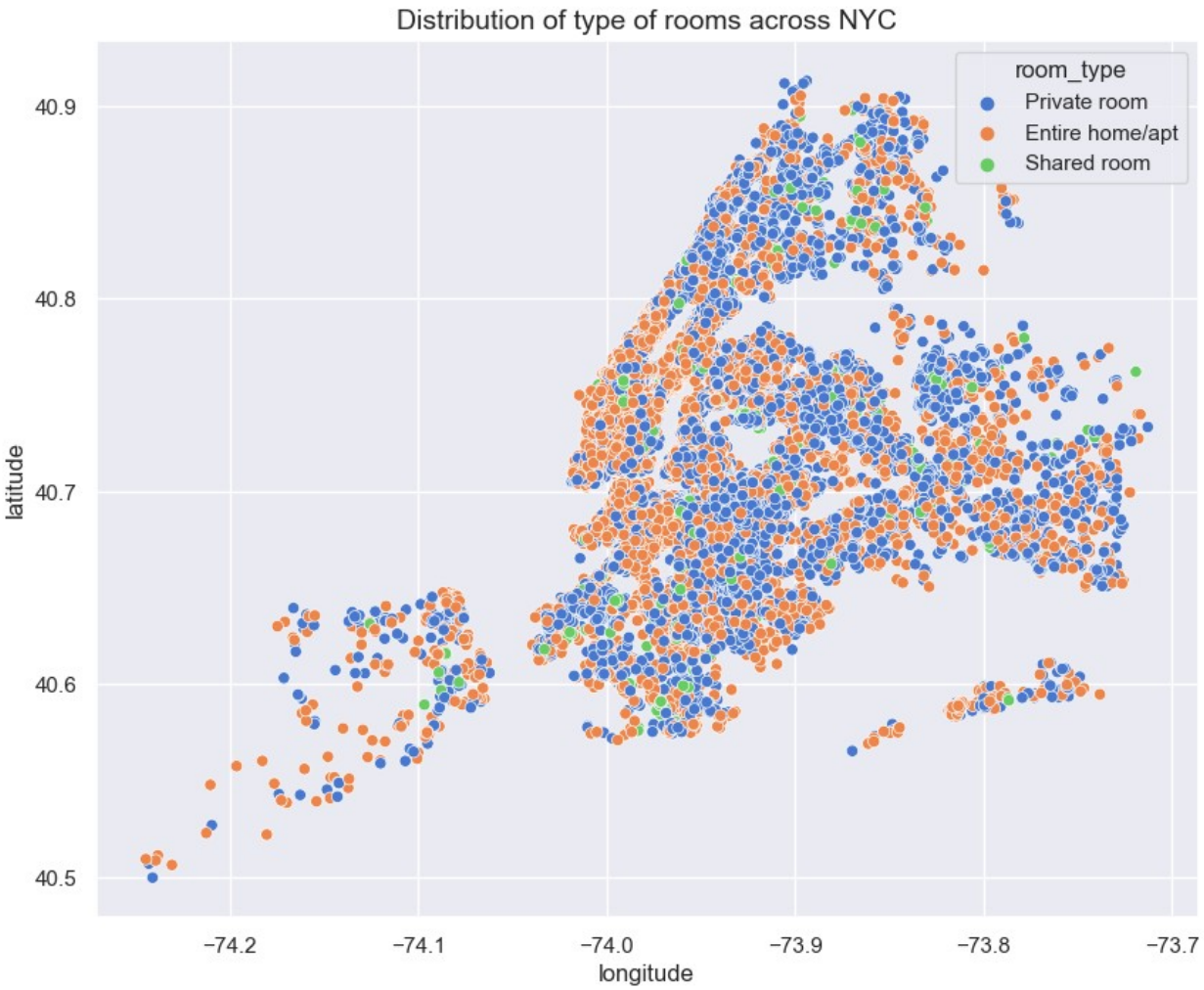
```
sns.set(rc={"figure.figsize": (10, 8)})

ax = sns.scatterplot(data=df, x="longitude", y="latitude",
hue='neighbourhood_group', palette='bright')

ax.set_title('Location Cordinates', fontsize='14')
```

```
Text(0.5, 1.0, 'Location Cordinates')
```




```
sns.set(rc={"figure.figsize": (10, 8)})

# create a scatter plot that displays the longitude and latitude of
the listings in the Airbnb NYC dataset with room_types.
ax = sns.scatterplot(x=df.longitude, y=df.latitude, hue=df.room_type,
palette='muted')

# set the title of the plot
ax.set_title('Distribution of type of rooms across NYC',
fontsize='14')
```

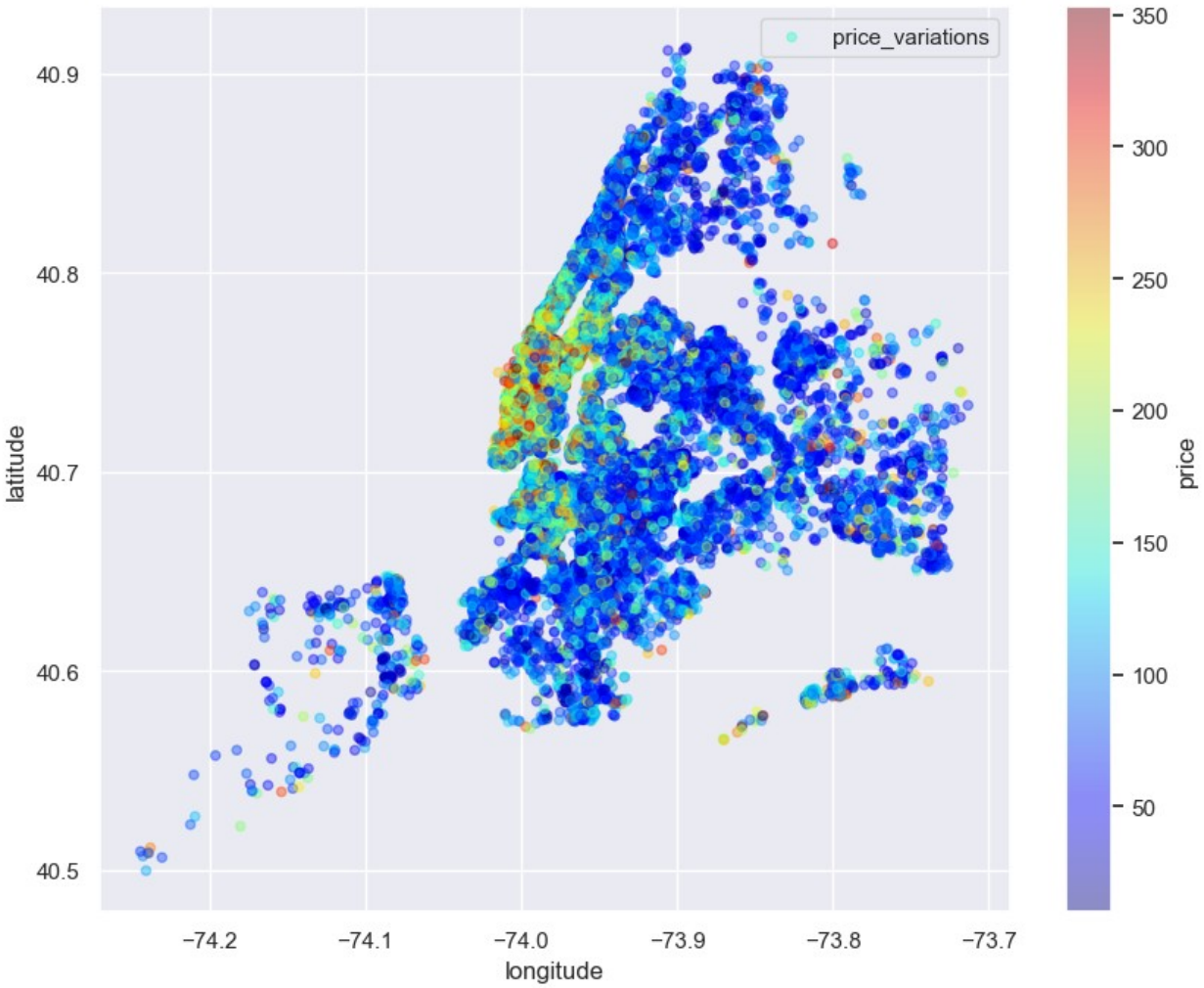
```
Text(0.5, 1.0, 'Distribution of type of rooms across NYC')
```



Price variations in NYC Neighbourhood groups

```
lat_long = df.plot(kind='scatter', x='longitude', y='latitude',
label='price_variations',c='price',
                   cmap=plt.get_cmap('jet'), colorbar=True, alpha=0.4,
figsize=(10, 8))

lat_long.legend()
```

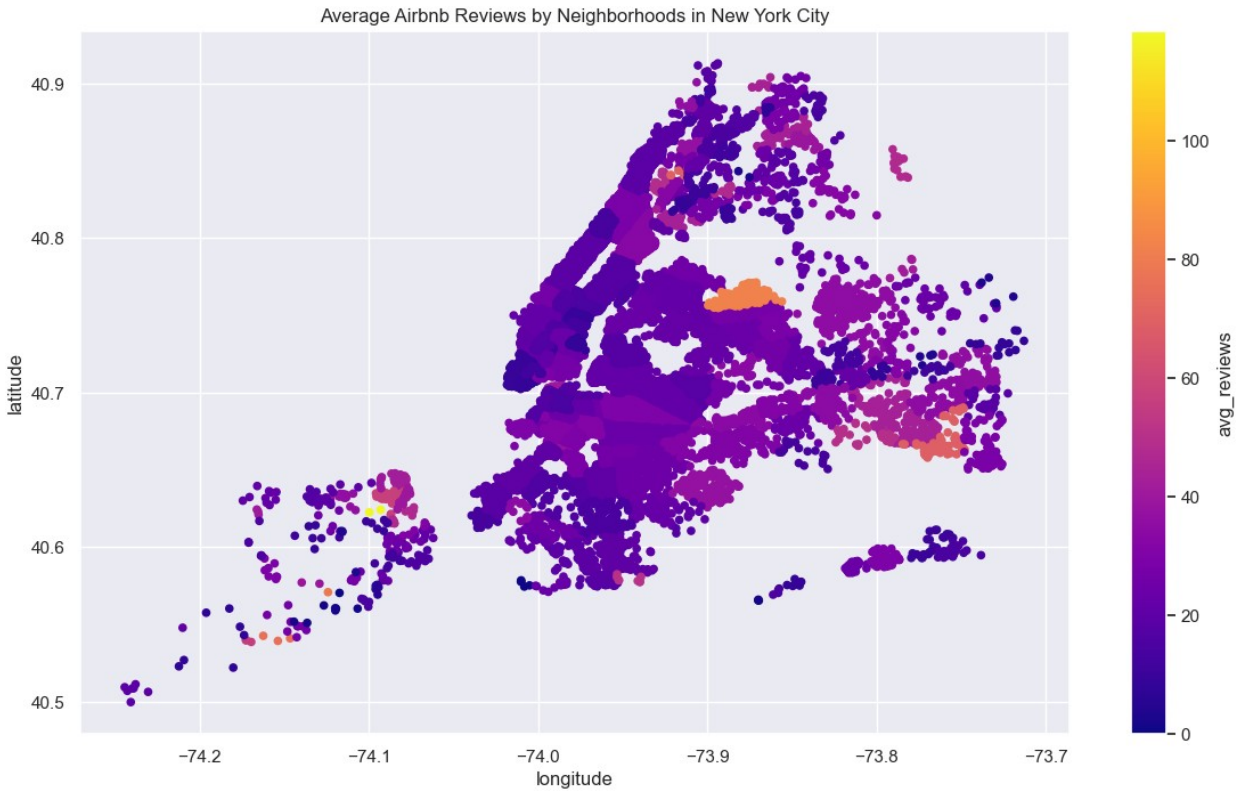
```
<matplotlib.legend.Legend at 0x18ae976f150>
```



Best Location Listing/Property Location by the reviews

```
neighbourhood_avg_reviews = df.groupby("neighbourhood")
["number_of_reviews"].mean()

neighbourhood_reviews = pd.DataFrame({"neighbourhood":
neighbourhood_avg_reviews.index, "avg_reviews":
neighbourhood_avg_reviews.values})

data = df.merge(neighbourhood_reviews, on="neighbourhood")

fig = data.plot.scatter(x="longitude", y="latitude", c="avg_reviews",
title="Average Airbnb Reviews by Neighborhoods in New York City",
figsize=(14,8), cmap="plasma")
fig
```

```
<Axes: title={'center': 'Average Airbnb Reviews by Neighborhoods in
New York City'}, xlabel='longitude', ylabel='latitude'>
```




```
df.to_csv('Airbnb NYC 2019.csv')
```

BUSINESS CONCLUSION :-

- Manhattan and Brooklyn have the highest demand for Airbnb rentals, as evidenced by the large number of listings in these neighborhoods. This could make them attractive areas for hosts to invest in property.
- Manhattan is world-famous for its parks, museums, buildings, town, liberty, gardens, markets, island and also its substantial number of tourists throughout the year ,it makes sense that demand and price both high.
- Brooklyn comes in second with significant number of listings and cheaper prices as compared to the Manhattan: With most listings located in Williamsburg and Bedford Stuyvesant two neighborhoods strategically close to Manhattan tourists get the chance to enjoy both boroughs equally while spending less.
- Williamsburg, Bedford-Stuyvesant, Harlem, Bushwick, and the Upper West Side are the top neighborhoods in terms of listing counts, indicating strong demand for Airbnb rentals in these areas.
- The average price of a listing in New York City is higher in the center of the city (Manhattan) compared to the outer boroughs. This could indicate that investing in property in Manhattan may be more lucrative for Airbnb rentals. But Manhattan and

Brooklyn have the largest number of hosts, indicating a high level of competition in these boroughs.

- The data suggests that Airbnb rentals are primarily used for short-term stays, with relatively few listings requiring a minimum stay of 30 nights or more. Hosts may want to consider investing in property that can accommodate shorter stays in order to maximize their occupancy rate.
- The majority of listings on Airbnb are for entire homes or apartments and also Private Rooms with relatively fewer listings for shared rooms. This suggests that travelers using Airbnb have a wide range of accommodation options to choose from, and hosts may want to consider investing in property that can accommodate multiple guests.
- The data indicates that the availability of Airbnb rentals varies significantly across neighborhoods, with some neighborhoods having a high concentration of listings and others having relatively few.
- The data indicates that there is a high level of competition among Airbnb hosts, with a small number of hosts dominating a large portion of the market. Hosts may want to consider investing in property in areas with relatively fewer listings in order to differentiate themselves from the competition.
- The neighborhoods near the airport in Queens would have a higher average number of reviews, as they are likely to attract a lot of tourists or visitors who are passing through the area. The proximity to the airport could make these neighborhoods a convenient and appealing place to stay for travelers for short-term stay with spending less money because The price distribution is high in Manhattan and Brooklyn.