Problem Description :

Airbnb is an online marketplace that connects people who want to rent out their properties with travelers seeking \ accommodations.As a popular platform for short-term rentals, Airbnb generates vast amounts of data related to property listings, host information, guest reviews, and pricing. \ This project aims to perform a comprehensive analysis of Airbnb data to gain insights into the rental market and understand \ factors that influence pricing and availability in different neighborhoods and room types

Conclusion:

The Airbnb Data Analysis project aims to provide valuable insights into the rental market by exploring and visualizing \ various aspects of the dataset. Through exploratory data analysis and geospatial visualization, this project will uncover \ patterns and trends related to property listings, pricing, and availability across differentneighborhoods and room types.

Import Libraries and Load Dataset

In [1]:

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

In [2]:

```
air =pd.read_csv(r'D:\DatSets\Airbnb_Data_Analysis\airbnb.csv')
air.head()
```

Out[2]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longit
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94

Data Exploration and Cleaning

In [3]:

```
air.shape
```

Out[3]:

```
(48895, 16)
```

In [4]:

```
air.dtypes
```

Out[4]:

```
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
```

In [5]:

```
air.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

In [6]:

```python
air.duplicated().sum() # to check any duplicatde values in rows in dataset
```

Out[6]:

```
0
```

In [7]:

```python
air.isnull().sum()
```

Out[7]:

```
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

We Dont Required Columns like, name,host_name, id,last_review, so we can drop these columns

In [8]:

```python
air.drop('id', inplace =True, axis=1)
```

In [9]:

```python
air.drop(['name','host_name','last_review'], axis=1, inplace =True)
```

In [10]:

```python
air.head(2)
```

Out[10]:

	host_id	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_r
0	2787	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	
1	2845	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	

In [11]:

```
air.isnull().sum()
```

Out[11]:

```
host_id                               0
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

Rreplace the 'reviews per month' by zero

In [12]:

```
air.fillna({'reviews_per_month':0}, inplace=True)
```

In [13]:

```
air.isnull().sum()
```

Out[13]:

```
host_id                           0
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

Remove the NaN values from the dataset

In [14]:

```
air.dropna(how='any', inplace=True)
```

In [15]:

```
air.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   host_id                         48895 non-null  int64  
 1   neighbourhood_group             48895 non-null  object 
 2   neighbourhood                   48895 non-null  object 
 3   latitude                        48895 non-null  float64
 4   longitude                       48895 non-null  float64
 5   room_type                       48895 non-null  object 
 6   price                           48895 non-null  int64  
 7   minimum_nights                  48895 non-null  int64  
 8   number_of_reviews               48895 non-null  int64  
 9   reviews_per_month               48895 non-null  float64
 10  calculated_host_listings_count  48895 non-null  int64  
 11  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(6), object(3)
memory usage: 4.5+ MB
```

Exploratory Data Analysis (EDA)

In [16]:
```
air.describe()
```

Out[16]:

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

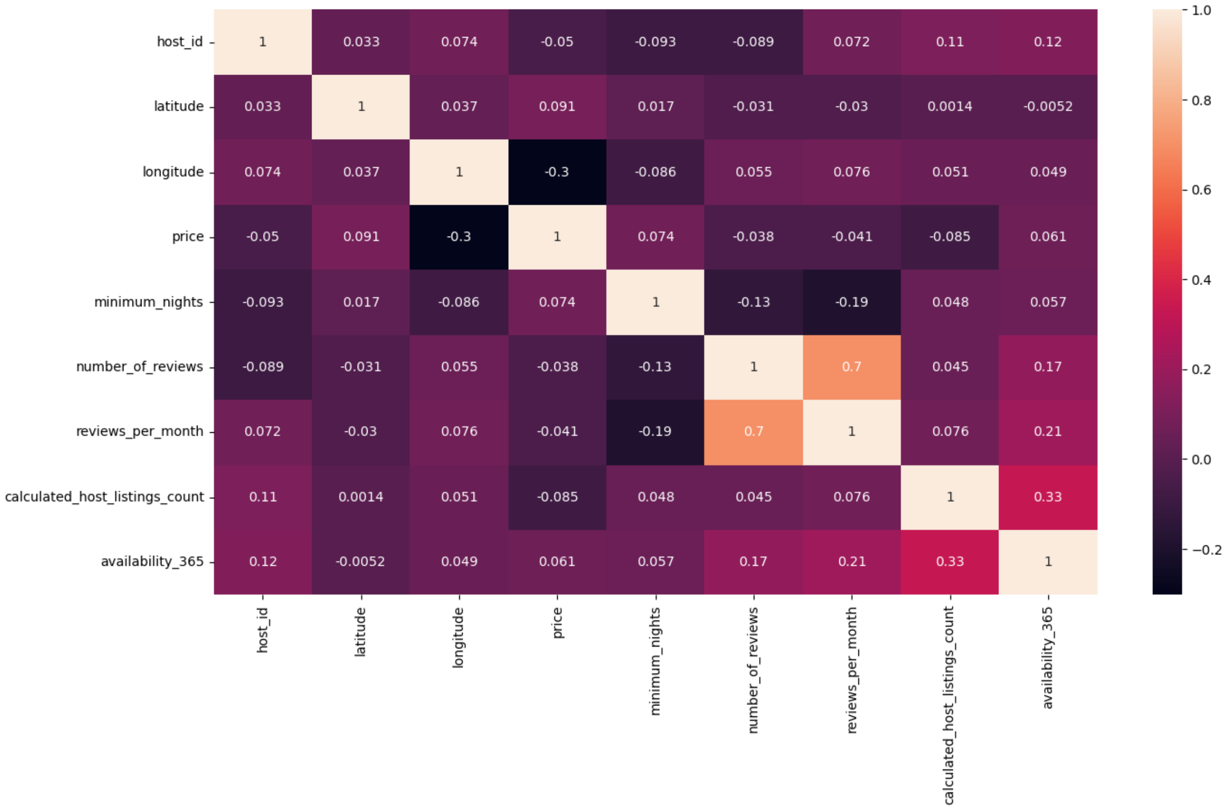
In [17]:
```
air.columns
```

Out[17]:
```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

Visualize the correlation between numerical attributes using a heatmap

In [18]:
```
plt.figure(figsize=(15,8))
corr = air.corr(method='kendall')
#The Kendall correlation coefficient (Kendall's Tau) measures the strength and directi
sns.heatmap(corr,annot=True)
```

Out[18]:
```
<Axes: >
```



Plot all Neighbourhood Group

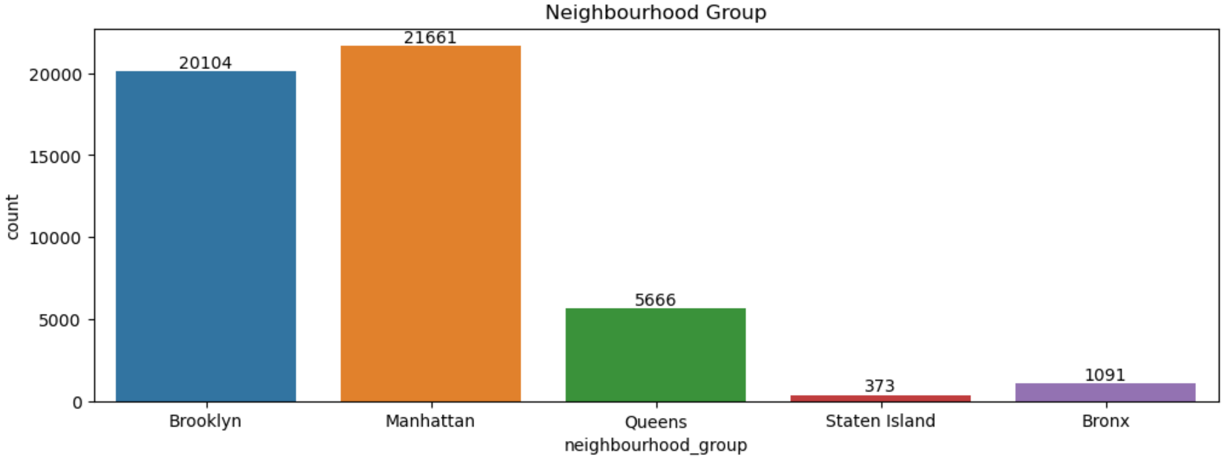
In [19]:
```
air.columns
```

Out[19]:
```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [20]:
```
air.neighbourhood_group.unique()
```

Out[20]:
```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

In [21]:
```
plt.figure(figsize=(12,4))
aa=sns.countplot(x=air['neighbourhood_group'])
for bars in aa.containers:
    aa.bar_label(bars)
plt.title('Neighbourhood Group')
plt.show()
```



Plot all Neighbourhood

In [22]:
```
air.columns
```

Out[22]:
```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [23]:
```
air.neighbourhood.unique()
```

Out[23]:

```
array(['Kensington', 'Midtown', 'Harlem', 'Clinton Hill', 'East Harlem',
       'Murray Hill', 'Bedford-Stuyvesant', "Hell's Kitchen",
       'Upper West Side', 'Chinatown', 'South Slope', 'West Village',
       'Williamsburg', 'Fort Greene', 'Chelsea', 'Crown Heights',
       'Park Slope', 'Windsor Terrace', 'Inwood', 'East Village',
       'Greenpoint', 'Bushwick', 'Flatbush', 'Lower East Side',
       'Prospect-Lefferts Gardens', 'Long Island City', 'Kips Bay',
       'SoHo', 'Upper East Side', 'Prospect Heights',
       'Washington Heights', 'Woodside', 'Brooklyn Heights',
       'Carroll Gardens', 'Gowanus', 'Flatlands', 'Cobble Hill',
       'Flushing', 'Boerum Hill', 'Sunnyside', 'DUMBO', 'St. George',
       'Highbridge', 'Financial District', 'Ridgewood',
       'Morningside Heights', 'Jamaica', 'Middle Village', 'NoHo',
       'Ditmars Steinway', 'Flatiron District', 'Roosevelt Island',
       'Greenwich Village', 'Little Italy', 'East Flatbush',
       'Tompkinsville', 'Astoria', 'Clason Point', 'Eastchester',
       'Kingsbridge', 'Two Bridges', 'Queens Village', 'Rockaway Beach',
       'Forest Hills', 'Nolita', 'Woodlawn', 'University Heights',
       'Gravesend', 'Gramercy', 'Allerton', 'East New York',
       'Theater District', 'Concourse Village', 'Sheepshead Bay',
       'Emerson Hill', 'Fort Hamilton', 'Bensonhurst', 'Tribeca',
       'Shore Acres', 'Sunset Park', 'Concourse', 'Elmhurst',
       'Brighton Beach', 'Jackson Heights', 'Cypress Hills', 'St. Albans',
       'Arrochar', 'Rego Park', 'Wakefield', 'Clifton', 'Bay Ridge',
       'Graniteville', 'Spuyten Duyvil', 'Stapleton', 'Briarwood',
       'Ozone Park', 'Columbia St', 'Vinegar Hill', 'Mott Haven',
       'Longwood', 'Canarsie', 'Battery Park City', 'Civic Center',
       'East Elmhurst', 'New Springville', 'Morris Heights', 'Arverne',
       'Cambria Heights', 'Tottenville', 'Mariners Harbor', 'Concord',
       'Borough Park', 'Bayside', 'Downtown Brooklyn', 'Port Morris',
       'Fieldston', 'Kew Gardens', 'Midwood', 'College Point',
       'Mount Eden', 'City Island', 'Glendale', 'Port Richmond',
       'Red Hook', 'Richmond Hill', 'Bellerose', 'Maspeth',
       'Williamsbridge', 'Soundview', 'Woodhaven', 'Woodrow',
       'Co-op City', 'Stuyvesant Town', 'Parkchester', 'North Riverdale',
       'Dyker Heights', 'Bronxdale', 'Sea Gate', 'Riverdale',
       'Kew Gardens Hills', 'Bay Terrace', 'Norwood', 'Claremont Village',
       'Whitestone', 'Fordham', 'Bayswater', 'Navy Yard', 'Brownsville',
       'Eltingville', 'Fresh Meadows', 'Mount Hope', 'Lighthouse Hill',
       'Springfield Gardens', 'Howard Beach', 'Belle Harbor',
       'Jamaica Estates', 'Van Nest', 'Morris Park', 'West Brighton',
       'Far Rockaway', 'South Ozone Park', 'Tremont', 'Corona',
       'Great Kills', 'Manhattan Beach', 'Marble Hill', 'Dongan Hills',
       'Castleton Corners', 'East Morrisania', 'Hunts Point', 'Neponsit',
       'Pelham Bay', 'Randall Manor', 'Throgs Neck', 'Todt Hill',
       'West Farms', 'Silver Lake', 'Morrisania', 'Laurelton',
       'Grymes Hill', 'Holliswood', 'Pelham Gardens', 'Belmont',
       'Rosedale', 'Edgemere', 'New Brighton', 'Midland Beach',
       'Baychester', 'Melrose', 'Bergen Beach', 'Richmondtown',
       'Howland Hook', 'Schuylerville', 'Coney Island', 'New Dorp Beach',
       "Prince's Bay", 'South Beach', 'Bath Beach', 'Jamaica Hills',
       'Oakwood', 'Castle Hill', 'Hollis', 'Douglaston', 'Huguenot',
       'Olinville', 'Edenwald', 'Grant City', 'Westerleigh',
       'Bay Terrace, Staten Island', 'Westchester Square', 'Little Neck',
       'Fort Wadsworth', 'Rosebank', 'Unionport', 'Mill Basin',
       'Arden Heights', "Bull's Head", 'New Dorp', 'Rossville',
       'Breezy Point', 'Willowbrook'], dtype=object)
```

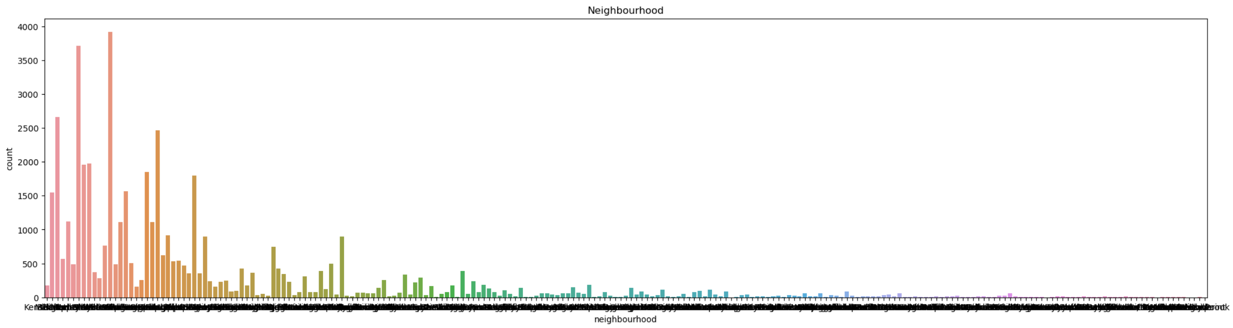
In [24]:

```python
plt.figure(figsize=(25,6))
sns.countplot(x=air['neighbourhood'])
```

```
plt.title('Neighbourhood')
```

Out[24]:

```
Text(0.5, 1.0, 'Neighbourhood')
```



Visualize the distribution of different room types using countplot().

In [25]:

```
air.columns
```

Out[25]:

```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [26]:

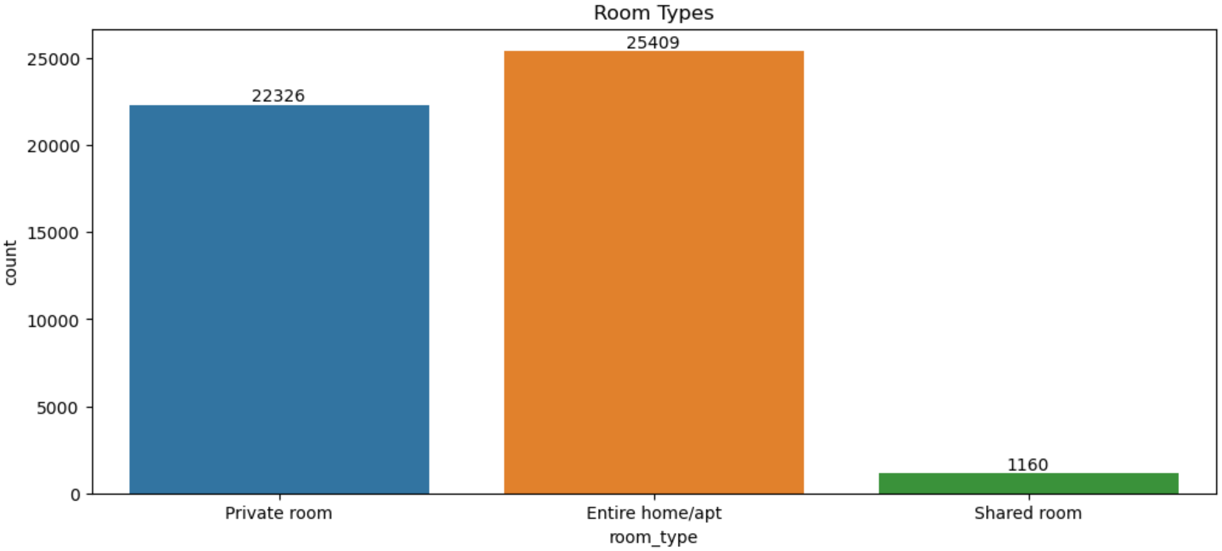
```
air.room_type.unique()
```

Out[26]:

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

In [27]:

```
plt.figure(figsize=(12,5))
ab =sns.countplot(x=air['room_type'])
for bars in ab.containers :
    ab.bar_label(bars)
plt.title('Room Types')
plt.show()
```



Relation between neighbourgroup and Availability of Room

In [28]:

```
air.columns
```

Out[28]:

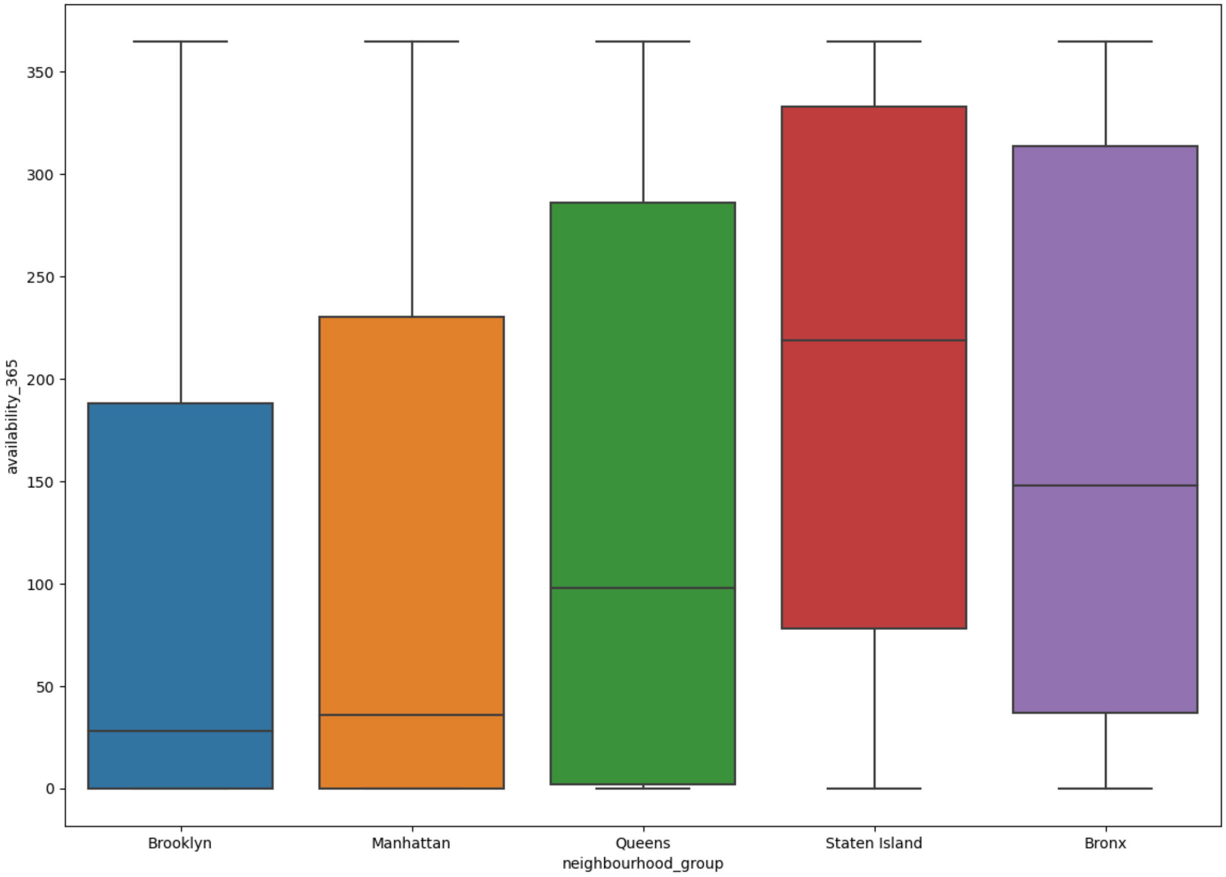
```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [29]:

```
plt.figure(figsize=(14,10))
sns.boxplot(data=air, x='neighbourhood_group', y='availability_365')
```

Out[29]:

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



Geospatial Analysis

In [30]:

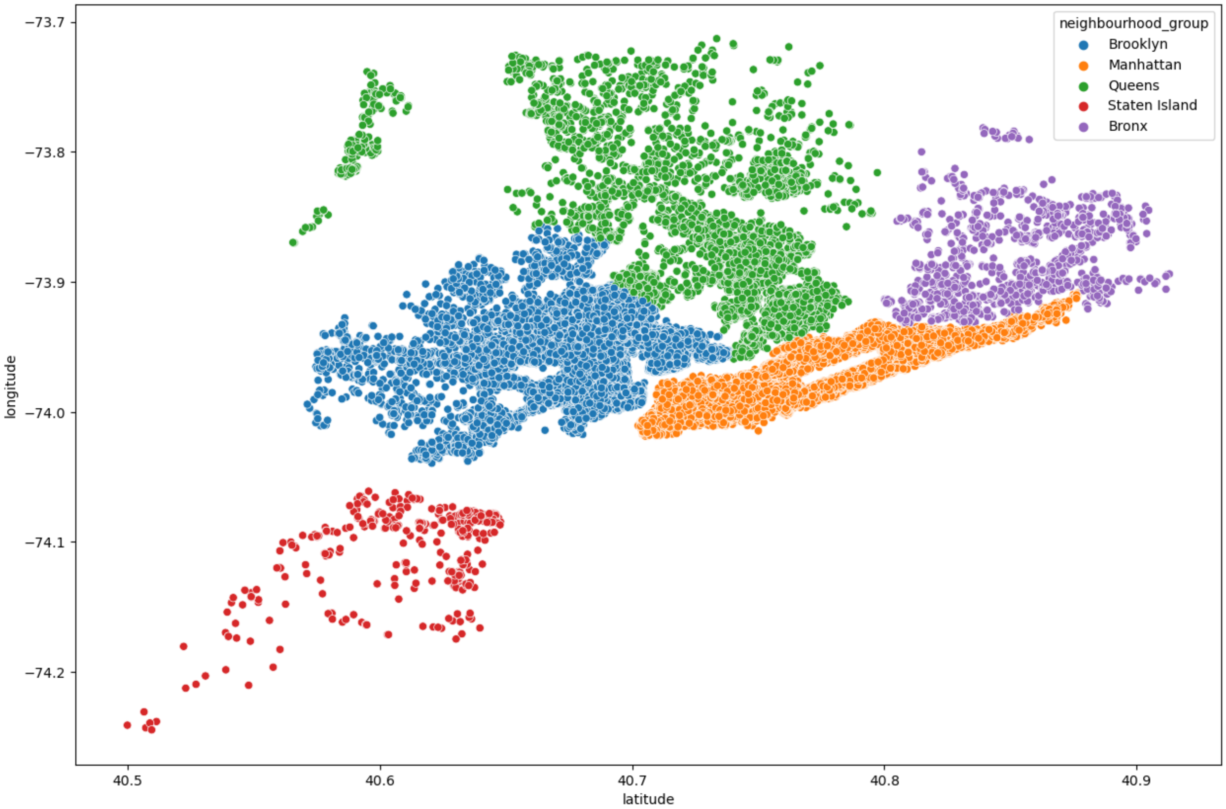
```
air.columns
```

Out[30]:

```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [31]:

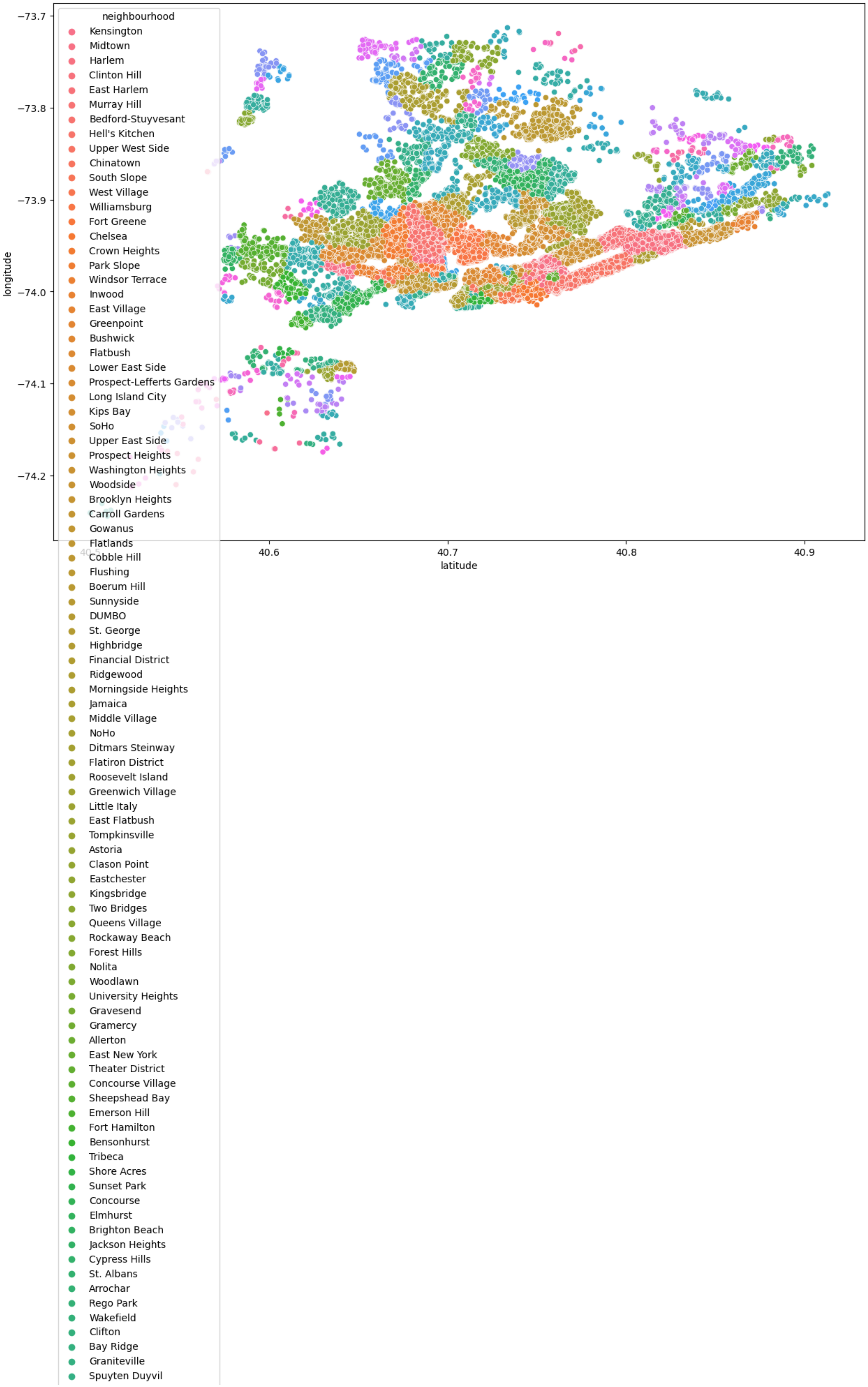
```
plt.figure(figsize=(15,10))
sns.scatterplot(data=air, x='latitude', y='longitude', hue='neighbourhood_group')
plt.ioff()
plt.show()
```

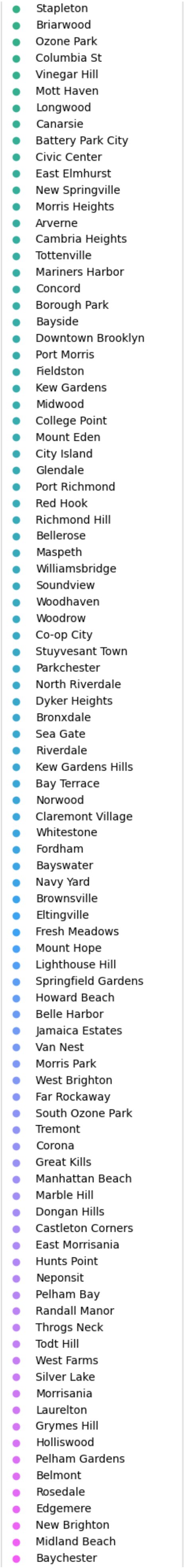


In [32]:
```
air.columns
```

Out[32]:
```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

In [33]:
```
plt.figure(figsize=(15,10))
sns.scatterplot(data=air, x='latitude', y='longitude', hue='neighbourhood')
plt.ioff()
plt.show()
```

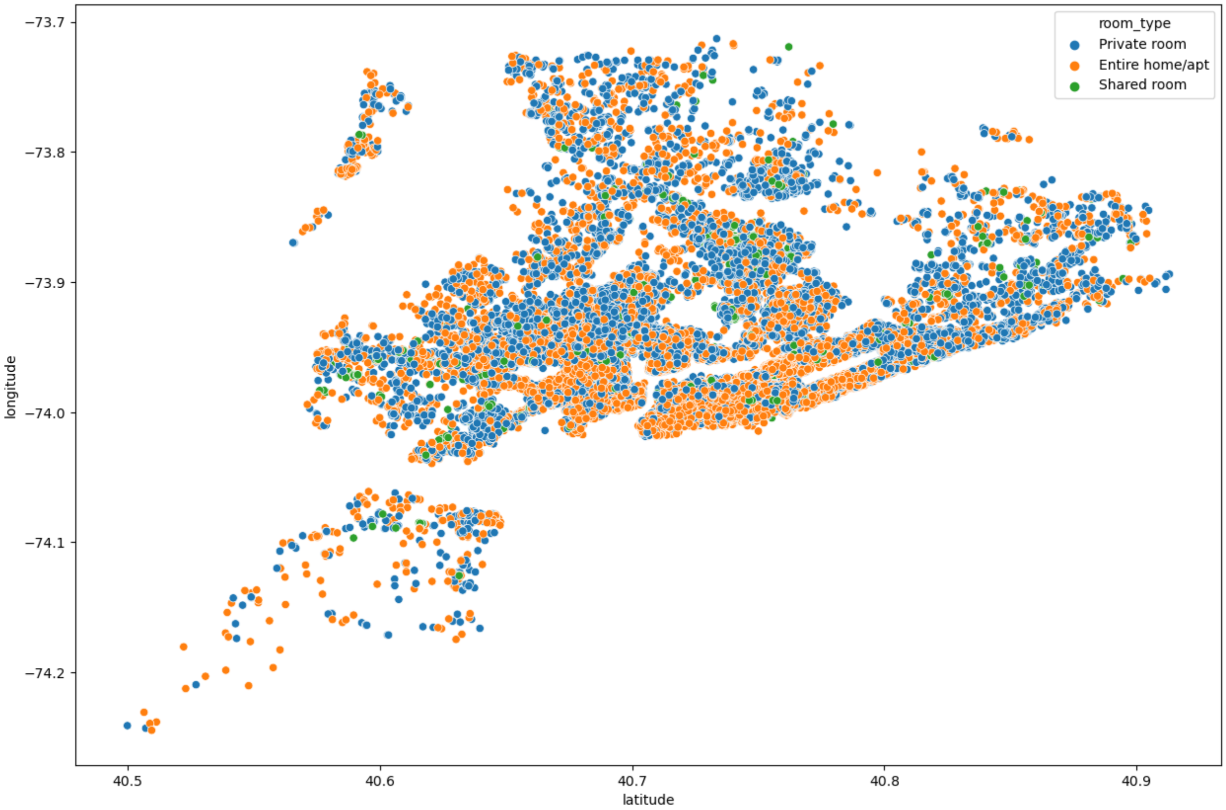

neighbourhood
Kensington
Midtown
Harlem
Clinton Hill
East Harlem
Murray Hill
Bedford-Stuyvesant
Hell's Kitchen
Upper West Side
Chinatown
South Slope
West Village
Williamsburg
Fort Greene
Chelsea
Crown Heights
Park Slope
Windsor Terrace
Inwood
East Village
Greenpoint
Bushwick
Flatbush
Lower East Side
Prospect-Lefferts Gardens
Long Island City
Kips Bay
SoHo
Upper East Side
Prospect Heights
Washington Heights
Woodside
Brooklyn Heights
Carroll Gardens
Gowanus
Flatlands
Cobble Hill
Flushing
Boerum Hill
Sunnyside
DUMBO
St. George
Highbridge
Financial District
Ridgewood
Morningside Heights
Jamaica
Middle Village
NoHo
Ditmars Steinway
Flatiron District
Roosevelt Island
Greenwich Village
Little Italy
East Flatbush
Tompkinsville
Astoria
Clason Point
Eastchester
Kingsbridge
Two Bridges
Queens Village
Rockaway Beach
Forest Hills
Nolita
Woodlawn
University Heights
Gravesend
Gramercy
Allerton
East New York
Theater District
Concourse Village
Sheepshead Bay
Emerson Hill
Fort Hamilton
Bensonhurst
Tribeca
Shore Acres
Sunset Park
Concourse
Elmhurst
Brighton Beach
Jackson Heights
Cypress Hills
St. Albans
Arrochar
Rego Park
Wakefield
Clifton
Bay Ridge
Graniteville
Spuyten Duyvil
−73.7
−73.8
−73.9
−74.0
−74.1
−74.2
longitude
40.5
40.6
40.7
40.8
40.9
latitude


Stapleton
Briarwood
Ozone Park
Columbia St
Vinegar Hill
Mott Haven
Longwood
Canarsie
Battery Park City
Civic Center
East Elmhurst
New Springville
Morris Heights
Arverne
Cambria Heights
Tottenville
Mariners Harbor
Concord
Borough Park
Bayside
Downtown Brooklyn
Port Morris
Fieldston
Kew Gardens
Midwood
College Point
Mount Eden
City Island
Glendale
Port Richmond
Red Hook
Richmond Hill
Bellerose
Maspeth
Williamsbridge
Soundview
Woodhaven
Woodrow
Co-op City
Stuyvesant Town
Parkchester
North Riverdale
Dyker Heights
Bronxdale
Sea Gate
Riverdale
Kew Gardens Hills
Bay Terrace
Norwood
Claremont Village
Whitestone
Fordham
Bayswater
Navy Yard
Brownsville
Eltingville
Fresh Meadows
Mount Hope
Lighthouse Hill
Springfield Gardens
Howard Beach
Belle Harbor
Jamaica Estates
Van Nest
Morris Park
West Brighton
Far Rockaway
South Ozone Park
Tremont
Corona
Great Kills
Manhattan Beach
Marble Hill
Dongan Hills
Castleton Corners
East Morrisania
Hunts Point
Neponsit
Pelham Bay
Randall Manor
Throgs Neck
Todt Hill
West Farms
Silver Lake
Morrisania
Laurelton
Grymes Hill
Holliswood
Pelham Gardens
Belmont
Rosedale
Edgemere
New Brighton
Midland Beach
Baychester

● Melrose
● Bergen Beach

In [34]:
```
plt.figure(figsize=(15,10))
sns.scatterplot(data=air, x='latitude', y='longitude', hue='room_type')
plt.ioff()
plt.show()
```

● Bath Beach


room_type
● Private room
● Entire home/apt
● Shared room
longitude
latitude