Python Project - AirBnB Listing 2024(New York)

Objective

The goal of this project is to:

1. Analyze **room types, prices, and availability** across different neighborhoods.
2. Understand **host behavior** and listing patterns.
3. Detect potential **outliers** in prices.
4. Provide recommendations for guests and hosts based on insights.

Project Overview

This project performs **Exploratory Data Analysis (EDA)** on New York Airbnb data to uncover trends and patterns in rental listings. We use libraries like **Pandas, Numpy, Matplotlib, Seaborn**for cleaning, visualization, and analysis.

In [2]:
```
# Import all the required library

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

%matplotlib inline
```

In [4]:
```
# Load the dataset

df = pd.read_csv("airbnb_datasets.csv")
```

In [5]:
```
df.head()  # Top 5 data
```

Out[5]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	l
0	1.312228e+06	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	55.0	...	
1	4.527754e+07	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100	Entire home/apt	144.0	...	
2	9.710000e+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	187.0	...	
3	3.857863e+06	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500	Private room	120.0	...	
4	4.089661e+07	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	Entire home/apt	85.0	...	

5 rows × 22 columns

In [6]:
```
df.tail()  # bottom 5 data
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	.
20765	2.473690e+07	Rental unit in New York · ★4.75 · 1 bedroom · ...	186680487	Henry D	Manhattan	Lower East Side	40.711380	-73.991560	Private room	45.0	
20766	2.835711e+06	Rental unit in New York · ★4.46 · 1 bedroom · ...	3237504	Aspen	Manhattan	Greenwich Village	40.730580	-74.000700	Entire home/apt	105.0	
20767	5.182527e+07	Rental unit in New York · ★4.93 · 1 bedroom · ...	304317395	Jeff	Manhattan	Hell's Kitchen	40.757350	-73.993430	Entire home/apt	299.0	
20768	7.830000e+17	Rental unit in New York · ★5.0 · 1 bedroom · 1...	163083101	Marissa	Manhattan	Chinatown	40.713750	-73.991470	Entire home/apt	115.0	
20769	5.660000e+17	Rental unit in Queens · ★4.89 · 1 bedroom · 1 ...	93827372	Glenroy	Queens	Rosedale	40.658874	-73.728651	Private room	102.0	

5 rows × 22 columns

In [7]:
```
df.shape  # total 20770 rows and 22 cloumns
```

Out[7]:
```
(20770, 22)
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20770 entries, 0 to 20769
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              20770 non-null  float64
 1   name                            20770 non-null  object 
 2   host_id                         20770 non-null  int64  
 3   host_name                       20770 non-null  object 
 4   neighbourhood_group             20770 non-null  object 
 5   neighbourhood                   20763 non-null  object 
 6   latitude                        20763 non-null  float64
 7   longitude                       20763 non-null  float64
 8   room_type                       20763 non-null  object 
 9   price                           20736 non-null  float64
 10  minimum_nights                  20763 non-null  float64
 11  number_of_reviews               20763 non-null  float64
 12  last_review                     20763 non-null  object 
 13  reviews_per_month               20763 non-null  float64
 14  calculated_host_listings_count  20763 non-null  float64
 15  availability_365                20763 non-null  float64
 16  number_of_reviews_ltm           20763 non-null  float64
 17  license                         20770 non-null  object 
 18  rating                          20770 non-null  object 
 19  bedrooms                        20770 non-null  object 
 20  beds                            20770 non-null  int64  
 21  baths                           20770 non-null  object 
dtypes: float64(10), int64(2), object(10)
memory usage: 3.5+ MB
```

In [9]:
```
# get the statistical summary

df.describe()
```

Out[9]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.000000	20763.000000	20763.000000
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.558493	42.610605	1.257589
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.532697	73.523401	1.904472
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	1.000000	0.010000
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.000000	4.000000	0.210000
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.000000	14.000000	0.650000
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.000000	49.000000	1.800000
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	75.490000

In [10]:
```
# check the null value

df.isnull()
```

Out[10]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	re
0	False	False	False	False	False	False	False	False	False	False	...	False	
1	False	False	False	False	False	False	False	False	False	False	...	False	
2	False	False	False	False	False	False	False	False	False	False	...	False	
3	False	False	False	False	False	False	False	False	False	False	...	False	
4	False	False	False	False	False	False	False	False	False	False	...	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	
20765	False	False	False	False	False	False	False	False	False	False	...	False	
20766	False	False	False	False	False	False	False	False	False	False	...	False	
20767	False	False	False	False	False	False	False	False	False	False	...	False	
20768	False	False	False	False	False	False	False	False	False	False	...	False	
20769	False	False	False	False	False	False	False	False	False	False	...	False	

20770 rows × 22 columns

In [11]:
```
# total null values

df.isnull().sum()
```

Out[11]:
```
id                                0
name                              0
host_id                           0
host_name                         0
neighbourhood_group               0
neighbourhood                     7
latitude                          7
longitude                         7
room_type                         7
price                            34
minimum_nights                    7
number_of_reviews                 7
last_review                       7
reviews_per_month                 7
calculated_host_listings_count    7
availability_365                  7
number_of_reviews_ltm             7
license                           0
rating                            0
bedrooms                          0
beds                              0
baths                             0
dtype: int64
```

In [12]:
```
# Now drop the null values as we have very less null data

df.dropna(inplace = True)
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
id                                0
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
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
number_of_reviews_ltm             0
license                           0
rating                            0
bedrooms                          0
beds                              0
baths                             0
dtype: int64
```

We can use fillna() function to fill the data by replacing the null value with mean, mode, median

In [14]:
```
# Check for the duplicate data
df.duplicated()
```

Out[14]:
```
0        False
1        False
2        False
3        False
4        False
         ...
20765    False
20766    False
20767    False
20768    False
20769    False
Length: 20736, dtype: bool
```

In [15]:
```
# total duplicate data
df.duplicated().sum()
```

Out[15]:
```
12
```

In [17]:
```
# get the duplicate data (rows)
df[df.duplicated()]
```

Out[17]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
6	4.527754e+07	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100	Entire home/apt	144.0
7	9.710000e+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	187.0
8	3.857863e+06	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500	Private room	120.0
9	4.089661e+07	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	Entire home/apt	85.0
10	4.958498e+07	Rental unit in New York · ★5.0 · 1 bedroom · 1...	51501835	Jeniffer	Manhattan	Hell's Kitchen	40.759950	-73.992960	Entire home/apt	115.0

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
20736	7.990000e+17	Rental unit in New York · 2 bedrooms · 2 beds ...	224733902	CozySuites Copake	Manhattan	Upper East Side	40.768970	-73.957592	Entire home/apt	153.0
20737	5.930000e+17	Rental unit in New York · ★4.79 · 2 bedrooms ·...	23219783	Rob	Manhattan	West Village	40.730220	-74.002910	Entire home/apt	175.0
20738	9.230000e+17	Loft in New York · ★4.33 · 1 bedroom · 2 beds ...	520265731	Rodrigo	Manhattan	Greenwich Village	40.728390	-73.999540	Entire home/apt	156.0
20739	1.336161e+07	Rental unit in New York · ★4.89 · 2 bedrooms ·...	8961407	Jamie	Manhattan	Harlem	40.805700	-73.946250	Entire home/apt	397.0
20740	5.119566e+07	Rental unit in New York · Studio · 1 bed · 1 bath	51501835	Jeniffer	Manhattan	Chinatown	40.718360	-73.995850	Entire home/apt	100.0

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
20741	2.523473e+07	Rental unit in New York · ★4.41 · 1 bedroom · ...	1497427	Mara	Manhattan	Upper East Side	40.774030	-73.950580	Entire home/apt	120.0
20742	3.339399e+06	Rental unit in New York · ★4.73 · 1 bedroom · ...	2119276	Urban Furnished	Manhattan	West Village	40.732030	-74.006760	Entire home/apt	143.0

12 rows × 22 columns

In [18]:
```
#  Now drop the duplicate data

df.drop_duplicates(inplace = True)
```

In [19]:
```
# Again check for duplicate data

df.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
# Check the data type for every columns

df.dtypes
```

Out[20]:
```
id                                float64
name                               object
host_id                             int64
host_name                          object
neighbourhood_group                object
neighbourhood                      object
latitude                          float64
longitude                         float64
room_type                          object
price                             float64
minimum_nights                    float64
number_of_reviews                 float64
last_review                        object
reviews_per_month                 float64
calculated_host_listings_count    float64
availability_365                  float64
number_of_reviews_ltm             float64
license                            object
rating                             object
bedrooms                           object
beds                                int64
baths                              object
dtype: object
```

In [21]:
```
# Convert the id and host_id column to object type

df['id'] = df ['id'].astype(object)
```

In [22]:
```
df['id'].dtypes
```

Out[22]:
```
dtype('O')
```

In [23]:
```
df['host_id'] = df ['host_id'].astype(object)
```

In [26]:
```
df['host_id'].dtypes
```

Out[26]:
```
dtype('O')
```

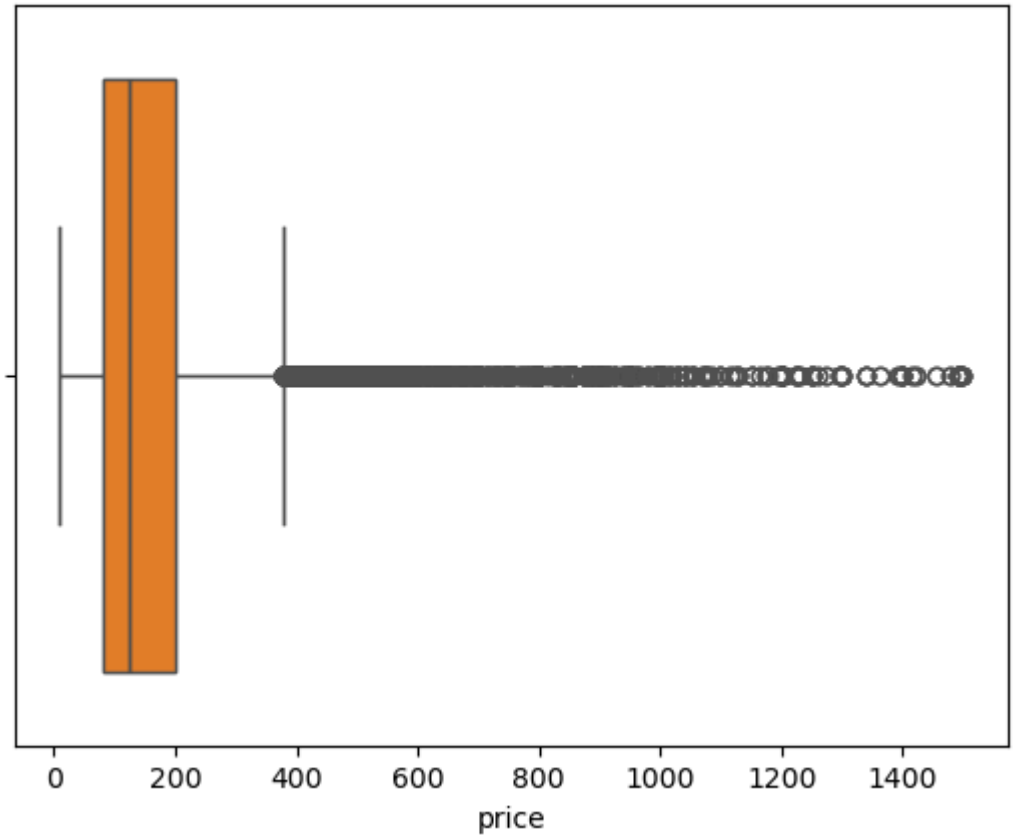
In [27]:
```
df.dtypes
```

Out[27]:
```
id                                 object
name                               object
host_id                            object
host_name                          object
neighbourhood_group                object
neighbourhood                      object
latitude                          float64
longitude                         float64
room_type                          object
price                             float64
minimum_nights                    float64
number_of_reviews                 float64
last_review                        object
reviews_per_month                 float64
calculated_host_listings_count    float64
availability_365                  float64
number_of_reviews_ltm             float64
license                            object
rating                             object
bedrooms                           object
beds                                int64
baths                              object
dtype: object
```

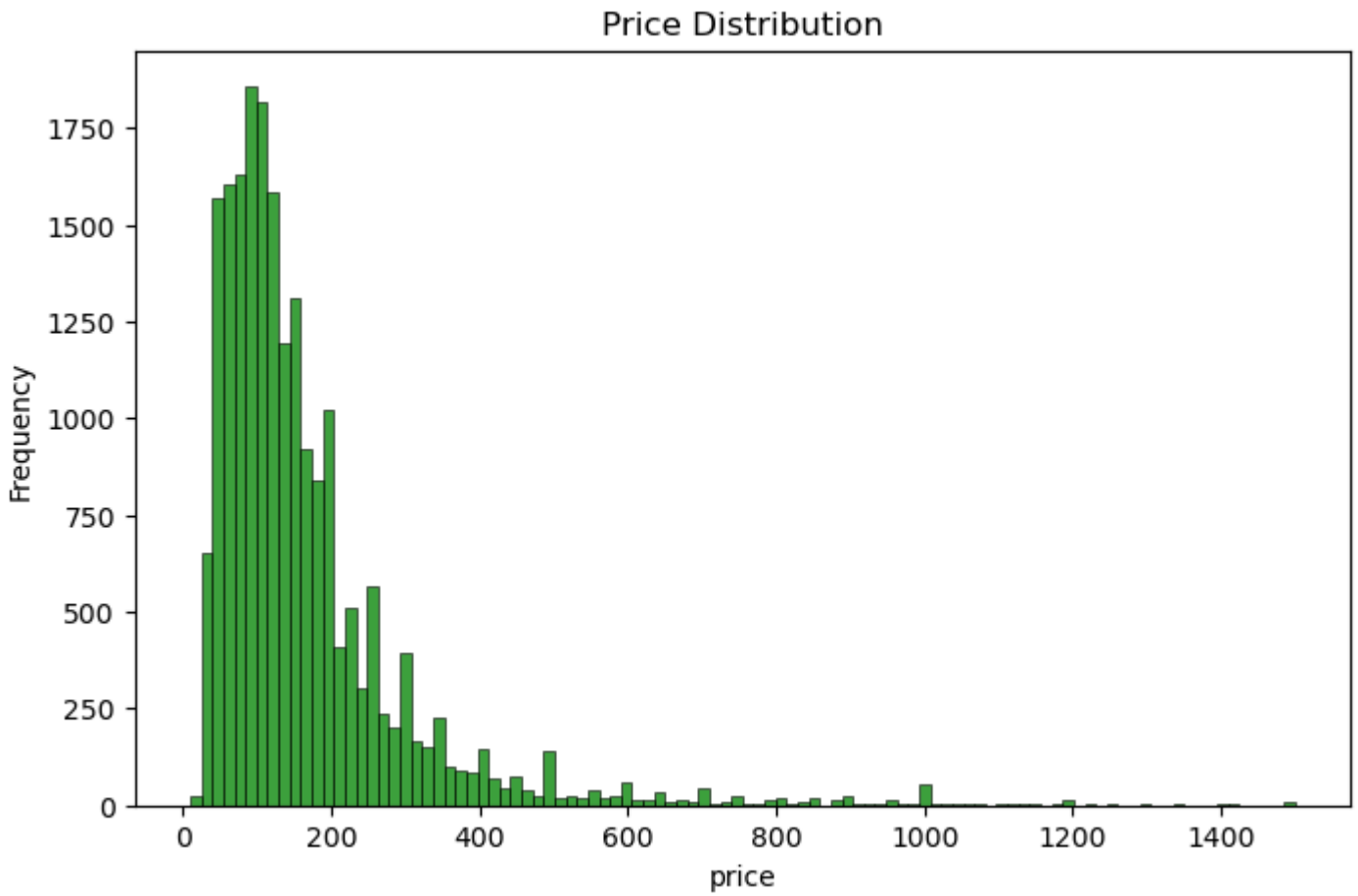
Perform the EDA

Univariate Analysis

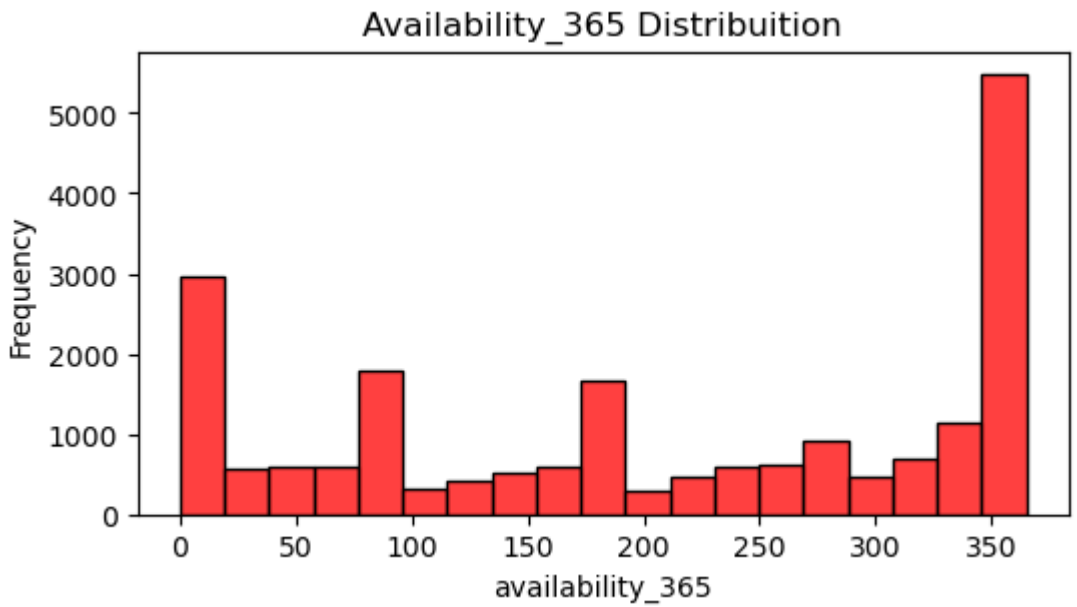
In [30]:
```
# Identify the outlier in price column

df = df[df['price']<1500]

sns.boxplot(data = df, x='price')

plt.show()
```



In [32]:
```
# Price Distribution

plt.figure(figsize = (8,5))
sns.histplot(data=df, x='price', bins = 100, color = 'green')
plt.title("Price Distribution")
plt.ylabel("Frequency")
plt.show()
```



In [34]:
```
#Price distribuion

plt.figure(figsize=(6, 3))
sns.histplot(data=df, x='availability_365', color = 'red')
plt.title('Availability_365 Distribuition')
plt.ylabel("Frequency")
plt.show()
```



In [36]:
```
# Get the group wise data of price with respect to the neighbourhood

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

Out[36]:
```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x000001BF090701D0>
```

In [37]:
```
df.groupby(by='neighbourhood_group')['price'].mean()
```

Out[37]:
```
neighbourhood_group
Bronx            107.990506
Brooklyn         155.138317
Manhattan        204.146014
Queens           121.681939
Staten Island    118.780069
Name: price, dtype: float64
```

In [39]:
```
# price per bed

df['price per bed']= df['price']/df['beds']
df.head()
```

Out[39]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	pri
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	5
1	45277537.0	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100	Entire home/apt	14
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	18
3	3857863.0	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500	Private room	12
4	40896611.0	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	Entire home/apt	8

5 rows × 23 columns

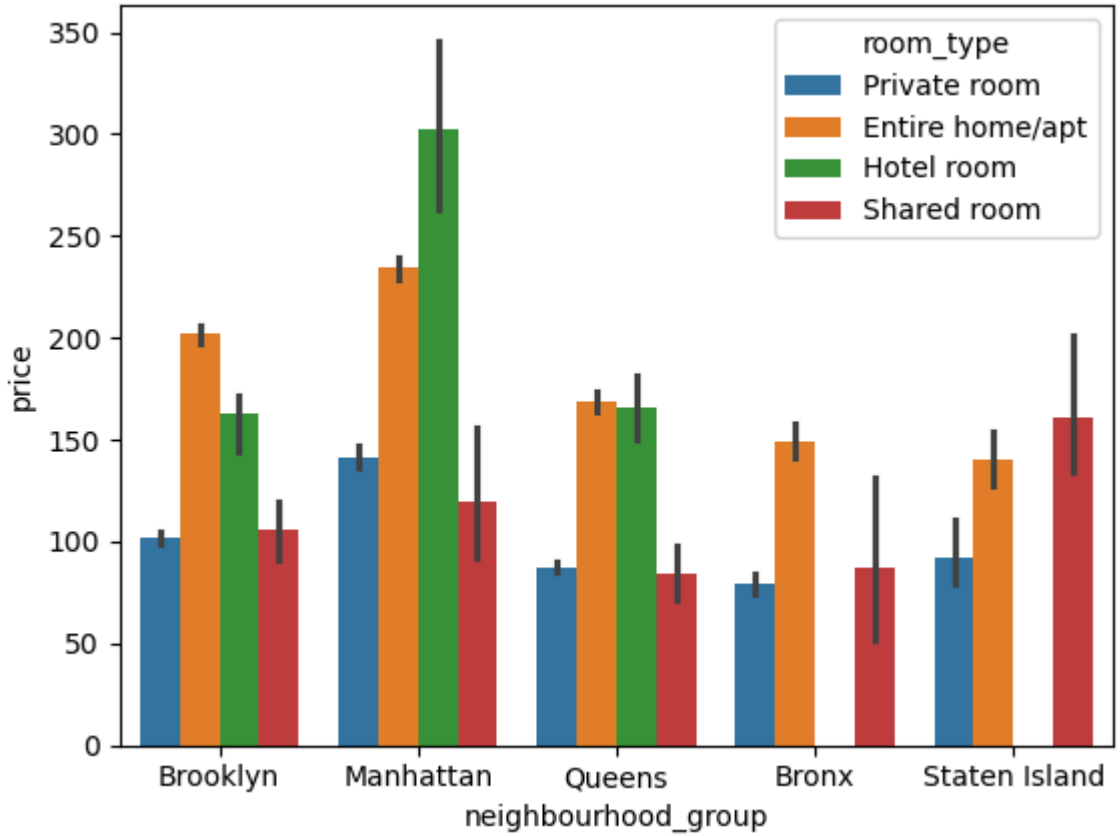
In [40]:
```
# average price per bed
df.groupby(by='neighbourhood_group')['price per bed'].mean()
```

Out[40]:
```
neighbourhood_group
Bronx             74.713639
Brooklyn          99.788493
Manhattan        138.708057
Queens            76.336210
Staten Island     67.728101
Name: price per bed, dtype: float64
```

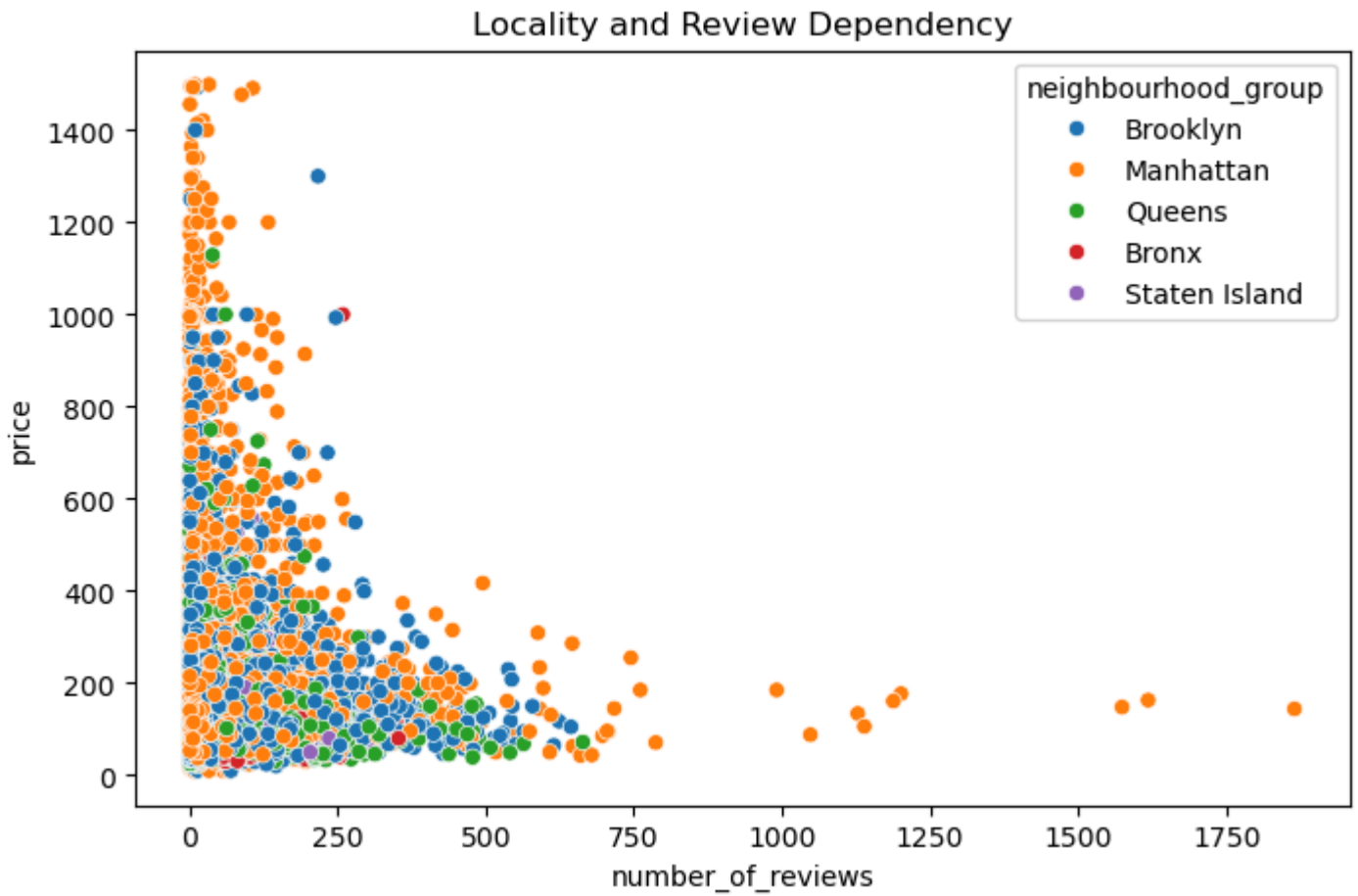
Bi Variable Analysis

One variable depenency in another variable

In [44]:
```
# price dependency on neighbourhood

sns.barplot(data = df, x = 'neighbourhood_group', y = 'price', hue = 'room_type')
plt.show()
```

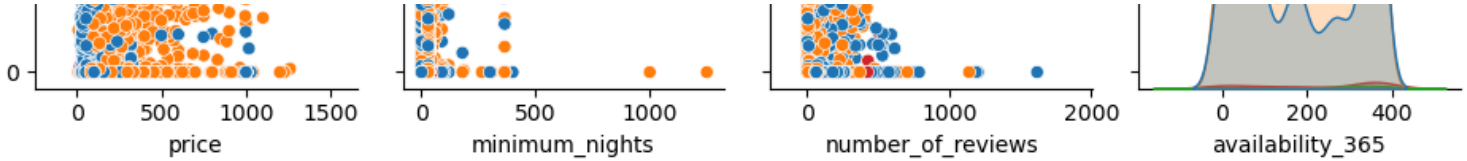


In [45]:
```
# Number of reviews and price relation

plt.figure(figsize=(8, 5))
plt.title("Locality and Review Dependency")
sns.scatterplot(data=df, x='number_of_reviews', y='price', hue='neighbourhood_group')
plt.show()
```



In [46]:
```
# Get the sub plot with respect to price, minimum_night, reviews, availability_365

sns.pairplot(data=df, vars=['price', 'minimum_nights',
                            'number_of_reviews', 'availability_365'],
             hue='room_type')
plt.show()
```


price
minimum_nights
number_of_reviews
availability_365
room_type
Private room
Entire home/apt
Hotel room
Shared room

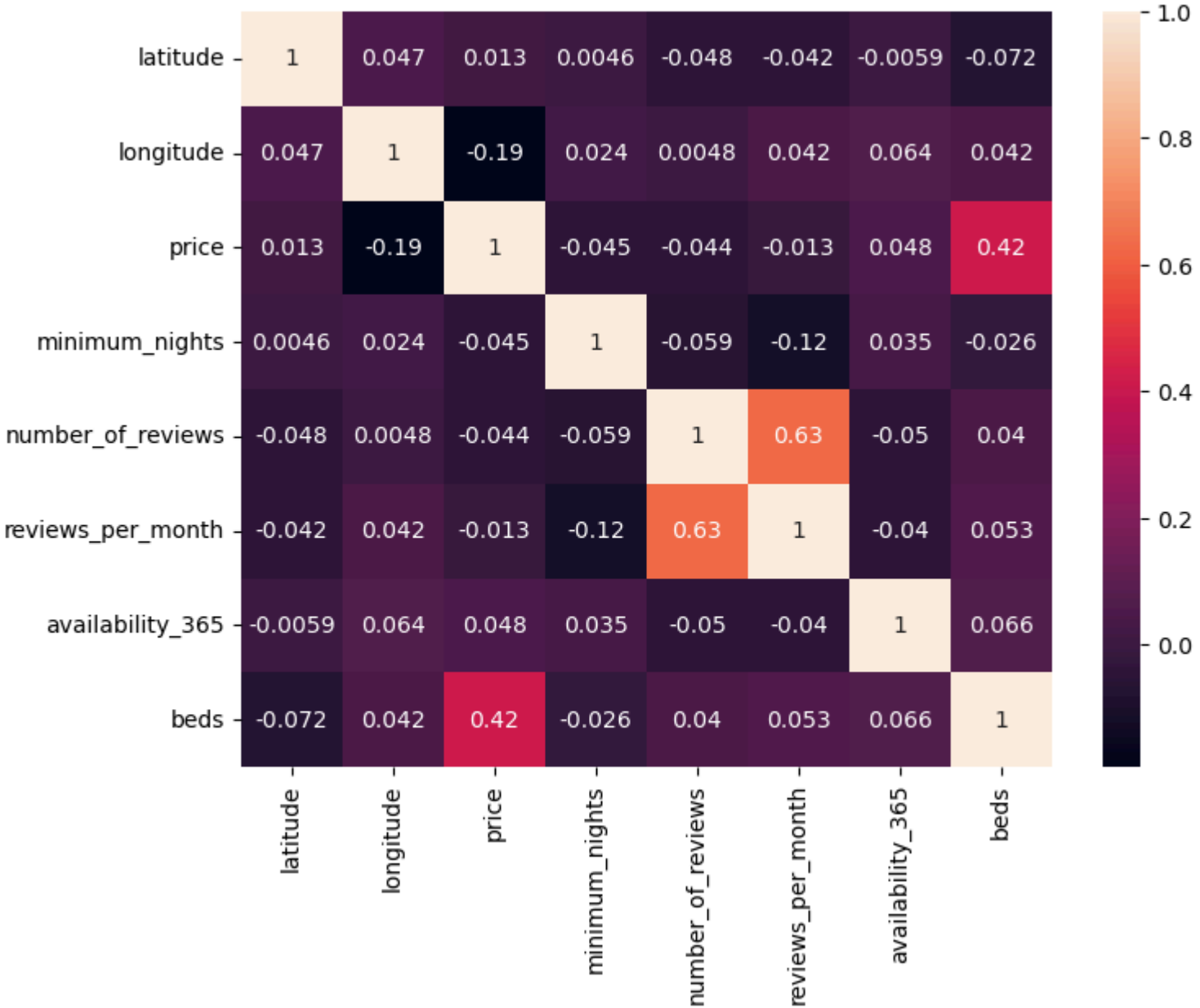


In [47]:
```
# Get the correlation of one variable with others for numerical column

corr = df[['latitude', 'longitude', 'price', 'minimum_nights',
           'number_of_reviews', 'reviews_per_month', 'availability_365', 'beds']].corr()
corr
```

Out[47]:

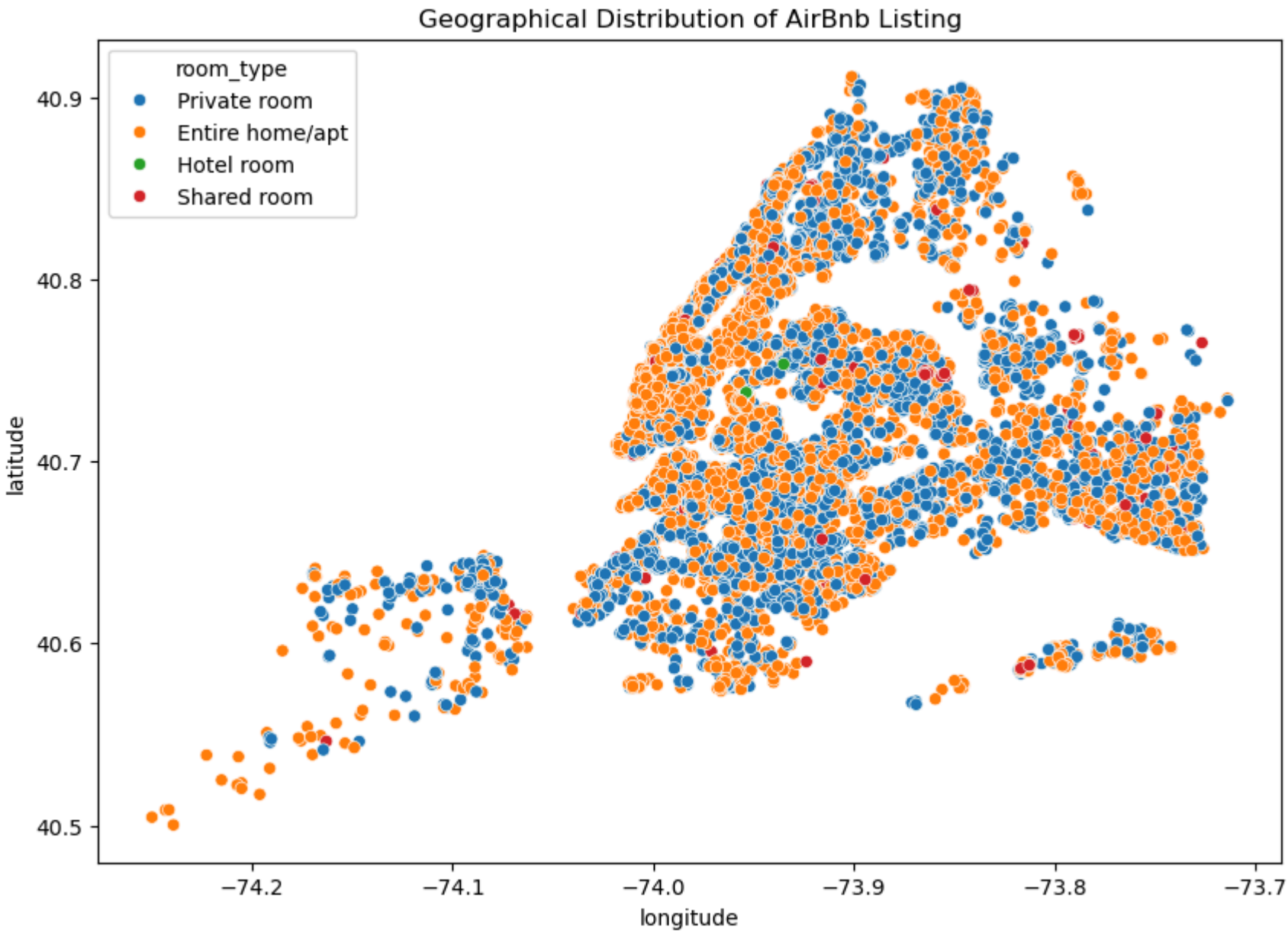
	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	availability_365	beds
latitude	1.000000	0.047369	0.012686	0.004590	-0.047953	-0.041673	-0.005941	-0.071753
longitude	0.047369	1.000000	-0.193728	0.023890	0.004820	0.041720	0.063523	0.041832
price	0.012686	-0.193728	1.000000	-0.044635	-0.043533	-0.012775	0.048036	0.415278
minimum_nights	0.004590	0.023890	-0.044635	1.000000	-0.059049	-0.122509	0.035466	-0.025852
number_of_reviews	-0.047953	0.004820	-0.043533	-0.059049	1.000000	0.631005	-0.049656	0.040071
reviews_per_month	-0.041673	0.041720	-0.012775	-0.122509	0.631005	1.000000	-0.040116	0.053496
availability_365	-0.005941	0.063523	0.048036	0.035466	-0.049656	-0.040116	1.000000	0.065985
beds	-0.071753	0.041832	0.415278	-0.025852	0.040071	0.053496	0.065985	1.000000

In [48]:
```
# present the correlation using the heatmap
plt.figure(figsize = (8,6))
sns.heatmap(data = corr, annot = True)
plt.show()
```



In [49]:
```
# Represent the Geographical distribution of AirBnB properties
```

```
plt.figure(figsize=(10, 7))
sns.scatterplot(data=df, x='longitude', y='latitude', hue='room_type')
plt.title("Geographical Distribution of AirBnb Listing")
plt.show()
```



Key Findings and Insights

1. **Price Trends**:
    - **Manhattan** has the most expensive listings, followed by Brooklyn.
    - **Entire homes/apartments** cost significantly more than private or shared rooms.
2. **Room Type Distribution**:
    - **Entire homes/apartments** are the most common, but **private rooms** offer budget-friendly options.
3. **Outliers in Price**:
    - Few listings priced at **$10,000+** were detected, indicating the need to filter such extreme values.
4. **Availability Patterns**:
    - Listings with **high availability** tend to have lower prices and more reviews, likely due to better guest experience.
5. **Host Behavior**:
    - Some hosts manage **multiple listings**, indicating a trend toward professional hosting.

Recommendations

- **For Guests**:
    - Look for listings with high availability and good reviews for a better experience.
    - **Private rooms** in Brooklyn offer affordable stays compared to Manhattan.
- **For Hosts**:
    - Improve **availability** and **review response rates** to attract more bookings.
    - Manage pricing effectively to compete within the borough's market.

Conclusion

This project offers valuable insights into the New York Airbnb market, helping both guests and hosts make informed decisions. By using **EDA techniques**, we identified key trends and developed actionable recommendations. Future improvements can involve advanced analytics and predictive modeling to further enhance the findings.

In [ ]: