AirBnb Python Project - AirBnB Listing 2024(New York)

steps

1. importing all dependenices (lib)
2. loading datasets
3. initial exploration
4. Data cleaning
5. Data Analysis

Task 1. Importing All Dependencies

In [1]:

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

%matplotlib inline
```

Task 2: Loading Datsets

In [3]:

```
data = pd.read_csv('new_york_listings_2024.csv', encoding_errors='ignore')
```

Task 3: Initial Exploration

In [6]:

```
data.head()
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_ltm	license	rating	bedrooms	beds	baths
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	...	20/12/15	0.03	1.0	0.0	0.0	No License	5	1	1	Not specified
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	...	01/05/23	0.24	139.0	364.0	2.0	No License	4.67	2	1	1
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	...	18/12/23	1.67	1.0	343.0	6.0	Exempt	4.17	1	2	1
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	...	17/09/23	1.38	2.0	363.0	12.0	No License	4.64	1	1	1
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	...	03/12/23	0.24	133.0	335.0	3.0	No License	4.91	Studio	1	1

5 rows × 22 columns

In [7]:

```
data.tail()
```

Out[7]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_ltm	license	rating	bedrooms	beds	baths
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	...	29/09/23	1.81	1.0	157.0	12.0	No License	4.75	1	1	1
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	...	01/07/23	0.48	1.0	0.0	1.0	No License	4.46	1	2	1
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	...	08/12/23	2.09	1.0	0.0	27.0	No License	4.93	1	1	1
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	...	17/09/23	0.91	1.0	363.0	7.0	No License	5	1	1	1
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	...	10/12/23	4.50	1.0	0.0	62.0	OSE-STRREG-0000513	4.89	1	1	1

5 rows × 22 columns

In [9]:

```
data.shape
```

Out[9]:

```
(20770, 22)
```

In [11]:

```
data.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 [12]:
```
# Statistical Summary
data.describe()
```

Out[12]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_ltm	beds
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.000000	20763.000000	20763.000000	20763.000000	20763.000000	20763.000000	20770.000000
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.558493	42.610605	1.257589	18.866686	206.067957	10.848962	1.723592
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.532697	73.523401	1.904472	70.921443	135.077259	21.354876	1.211993
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	1.000000	0.010000	1.000000	0.000000	0.000000	1.000000
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.000000	4.000000	0.210000	1.000000	87.000000	1.000000	1.000000
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.000000	14.000000	0.650000	2.000000	215.000000	3.000000	1.000000
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.000000	49.000000	1.800000	5.000000	353.000000	15.000000	2.000000
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	75.490000	713.000000	365.000000	1075.000000	42.000000

Task 4: Data Cleaning

In [14]:
```
data.isnull().sum()

# dropping all missing values rows
data.dropna(inplace=True)

# data.fillna()
data.isnull().sum()
```

Out[14]:
```
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
```

In [20]:
```
# dealing with duplicates rows
data.duplicated().sum()
```

```
# deleting all duplicated rows
# data[data.duplicated()]

data.drop_duplicates(inplace=True)
data.duplicated().sum()
```

Out[20]:
```
np.int64(0)
```

In [26]:
```
# type casting
# changing data types

data.dtypes

data['id'] = data['id'].astype(object)
data.dtypes

data['host_id'] = data['host_id'].astype(object)
data.dtypes
```

Out[26]:
```
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
```

EDA

Task 5: Data Analysis

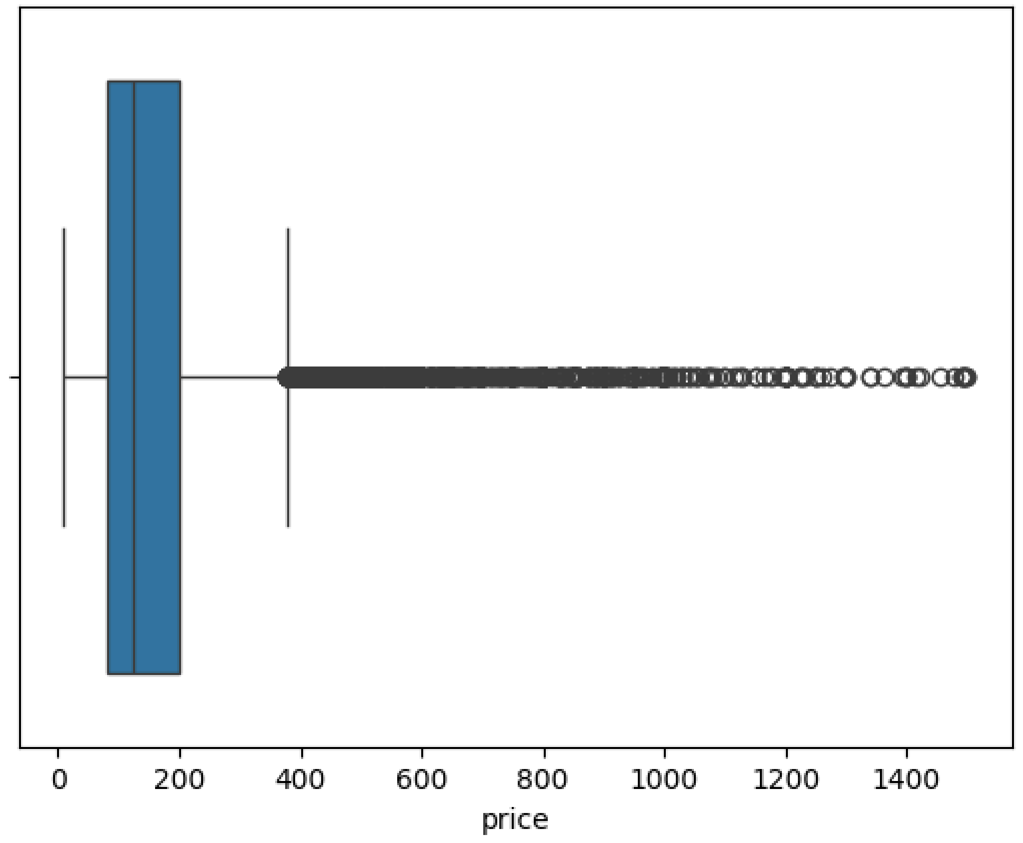
Univariate Analysis

In [32]:
```
# idenfying outliers in price

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

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

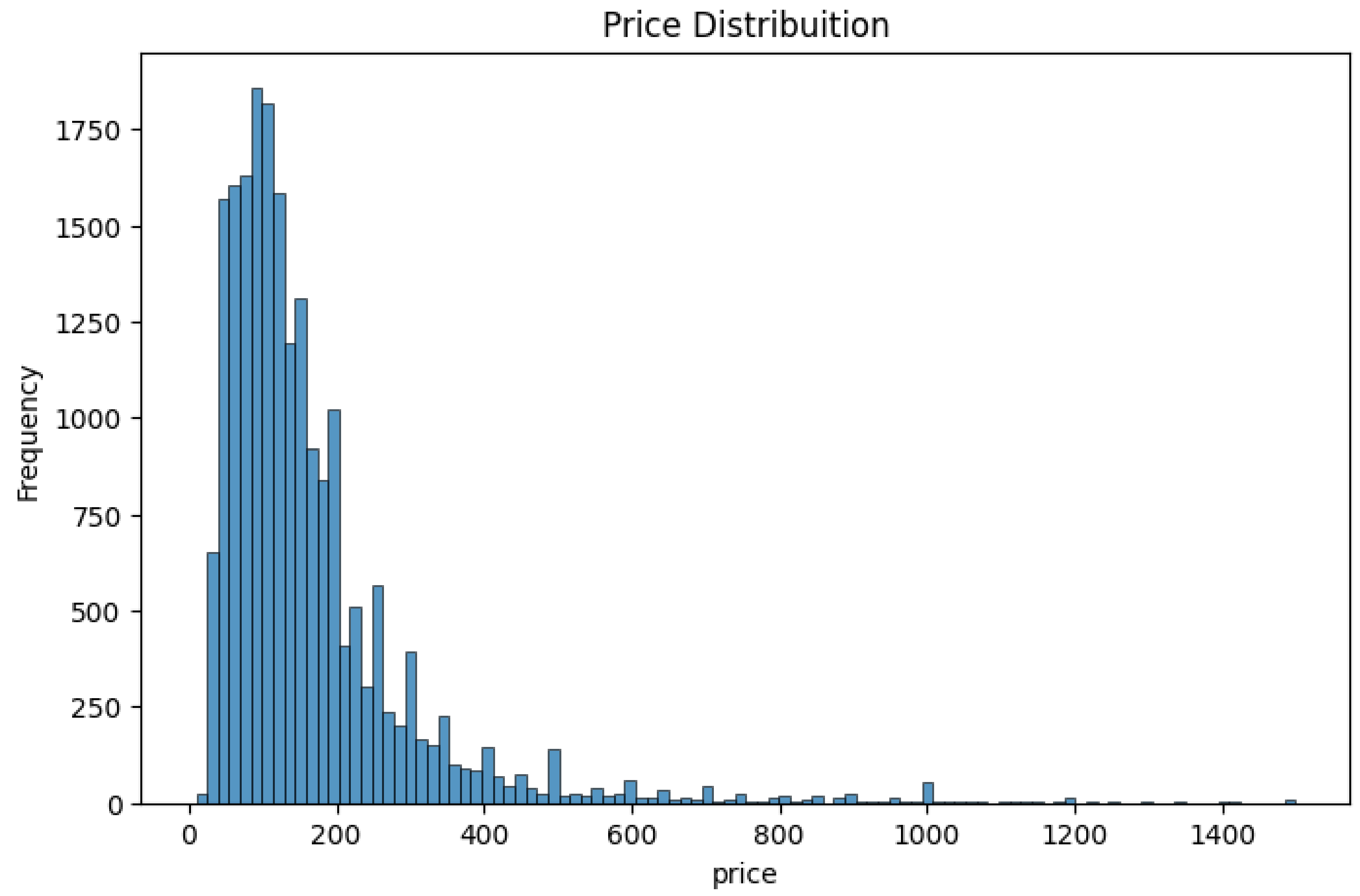
Out[32]:
```
<Axes: xlabel='price'>
```



In [41]:
```
#Price distribuion

plt.figure(figsize=(8, 5))
```

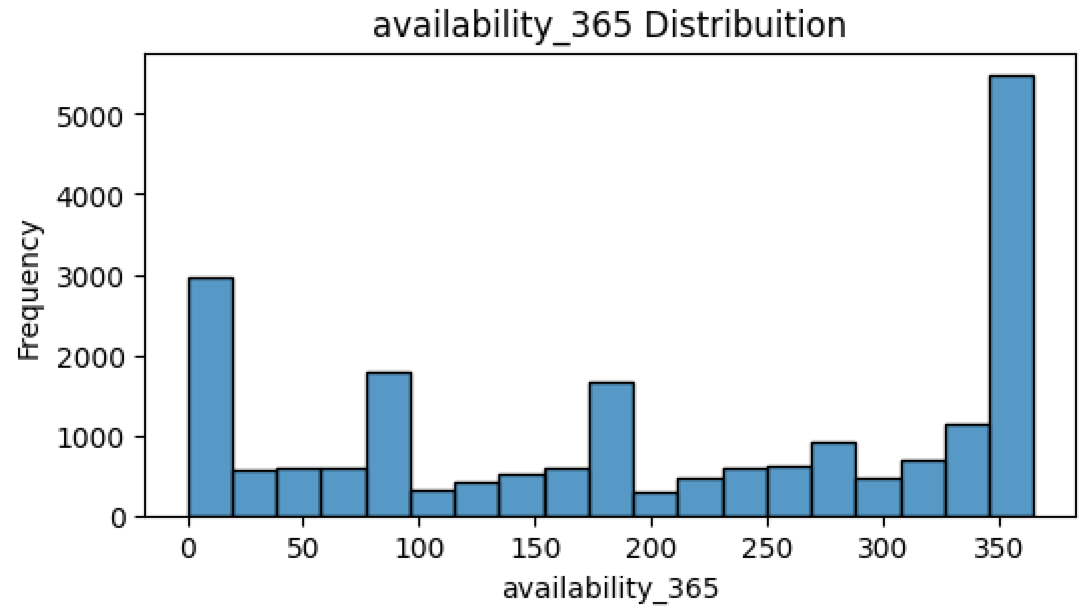
```
sns.histplot(data=df, x='price', bins=100)
plt.title('Price Distribuition')
plt.ylabel("Frequency")
plt.show()
```



```
In [39]: df.dtypes
```

```
Out[39]: 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
```

```
In [44]: #Price distribuion
         plt.figure(figsize=(6, 3))
         sns.histplot(data=df, x='availability_365')
         plt.title('availability_365 Distribuition')
         plt.ylabel("Frequency")
         plt.show()
```



```
In [53]: data.dtypes
```

Out[53]:
```
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
```

In [54]:
```
df.groupby(by='neighbourhood_group')['price'].mean()
```

Out[54]:
```
neighbourhood_group
Bronx            107.990506
Brooklyn         155.138317
Manhattan        204.146014
Queens           121.681939
Staten Island    118.780069
Name: price, dtype: float64
```

Feature Engineering

In [57]:
```
# average price per bed
df.groupby(by='neighbourhood_group')['price per bed'].mean()
```

Out[57]:
```
neighbourhood_group
Bronx             74.713639
Brooklyn          99.788493
Manhattan        138.708057
Queens            76.336210
Staten Island     67.728101
Name: price per bed, dtype: float64
```

In [56]:
```
# ['price per bed']

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

```
/var/folders/r6/w18clwvn0bv96s8gbr9087c40000gn/T/ipykernel_67159/2324310957.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['price per bed']= df['price']/df['beds']
```

Out[56]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_ltm	license	rating	bedrooms	beds	baths	price per bed
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	55.0	...	0.03	1.0	0.0	0.0	No License	5	1	1	Not specified	55.0
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	144.0	...	0.24	139.0	364.0	2.0	No License	4.67	2	1	1	144.0
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	187.0	...	1.67	1.0	343.0	6.0	Exempt	4.17	1	2	1	93.5
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	120.0	...	1.38	2.0	363.0	12.0	No License	4.64	1	1	1	120.0
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	85.0	...	0.24	133.0	335.0	3.0	No License	4.91	Studio	1	1	85.0

5 rows × 23 columns

In [55]:
```
df.head()
```

Out[55]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_ltm	license	rating	bedrooms	beds	baths
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	55.0	...	20/12/15	0.03	1.0	0.0	0.0	No License	5	1	1	Not specified
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	144.0	...	01/05/23	0.24	139.0	364.0	2.0	No License	4.67	2	1	1
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	187.0	...	18/12/23	1.67	1.0	343.0	6.0	Exempt	4.17	1	2	1
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	120.0	...	17/09/23	1.38	2.0	363.0	12.0	No License	4.64	1	1	1
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	85.0	...	03/12/23	0.24	133.0	335.0	3.0	No License	4.91	Studio	1	1

5 rows × 22 columns

Bi Variable Analysis One variable depenency in another variable

In [58]:

```
df.columns
```

Out[58]:

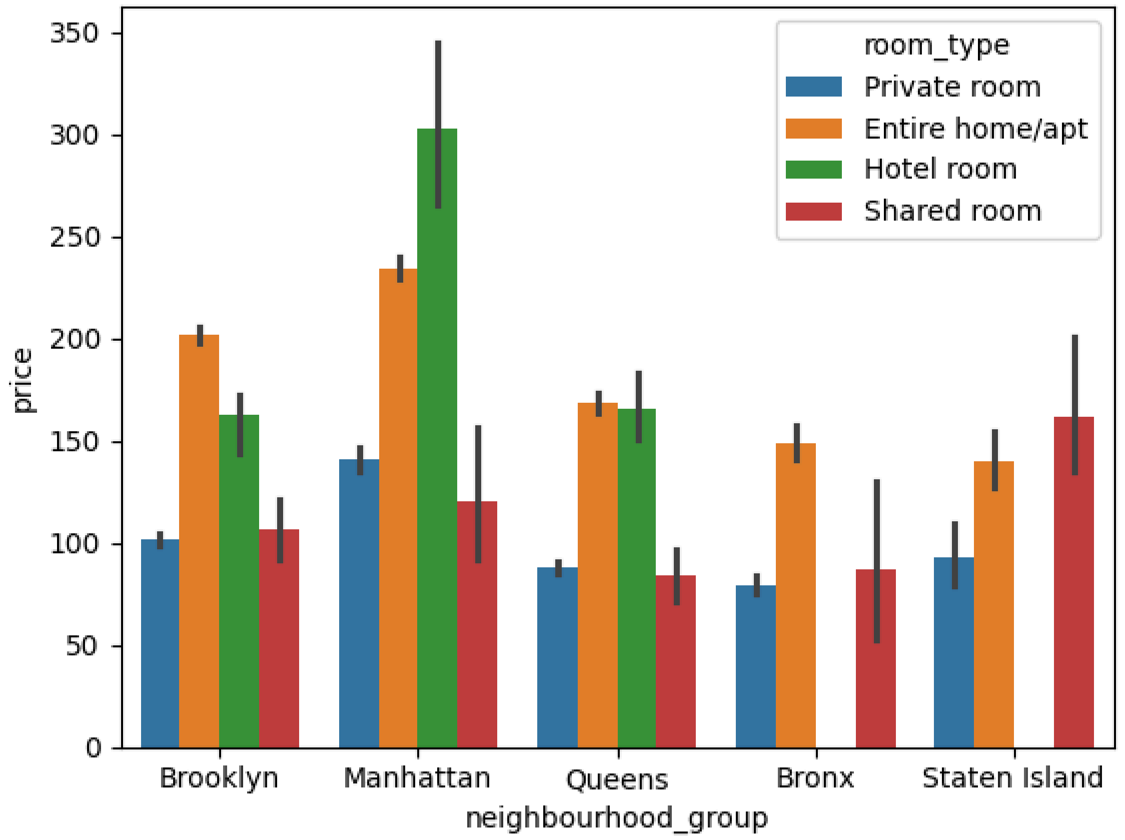
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'number_of_reviews_ltm', 'license', 'rating',
       'bedrooms', 'beds', 'baths', 'price per bed'],
      dtype='object')
```

In [60]:

```
# price dependency on neighbourhood
sns.barplot(data=df, x='neighbourhood_group', y='price', hue='room_type')
```

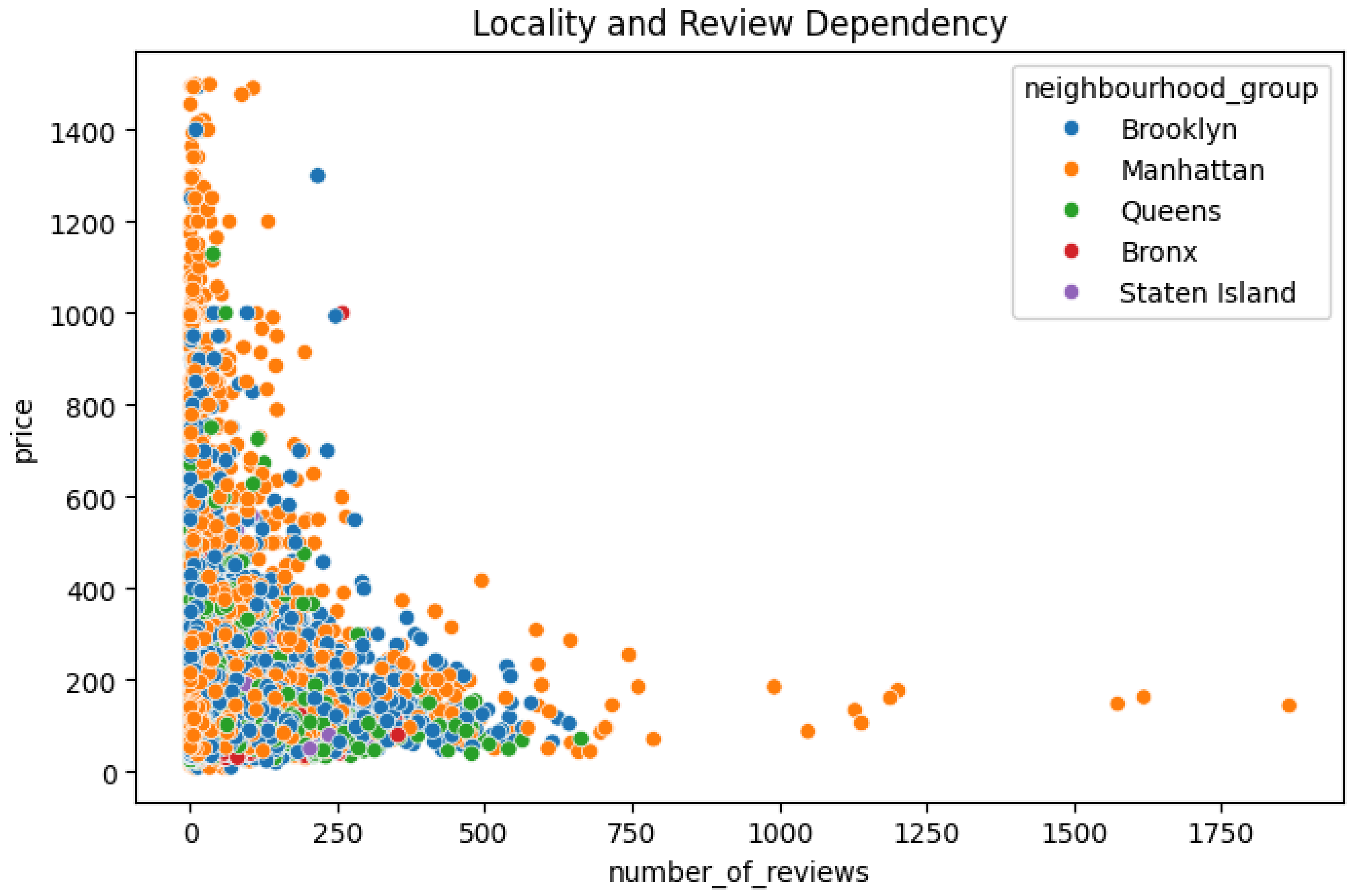
Out[60]:

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



In [66]:

```
# number of reviews and price rel
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()
```

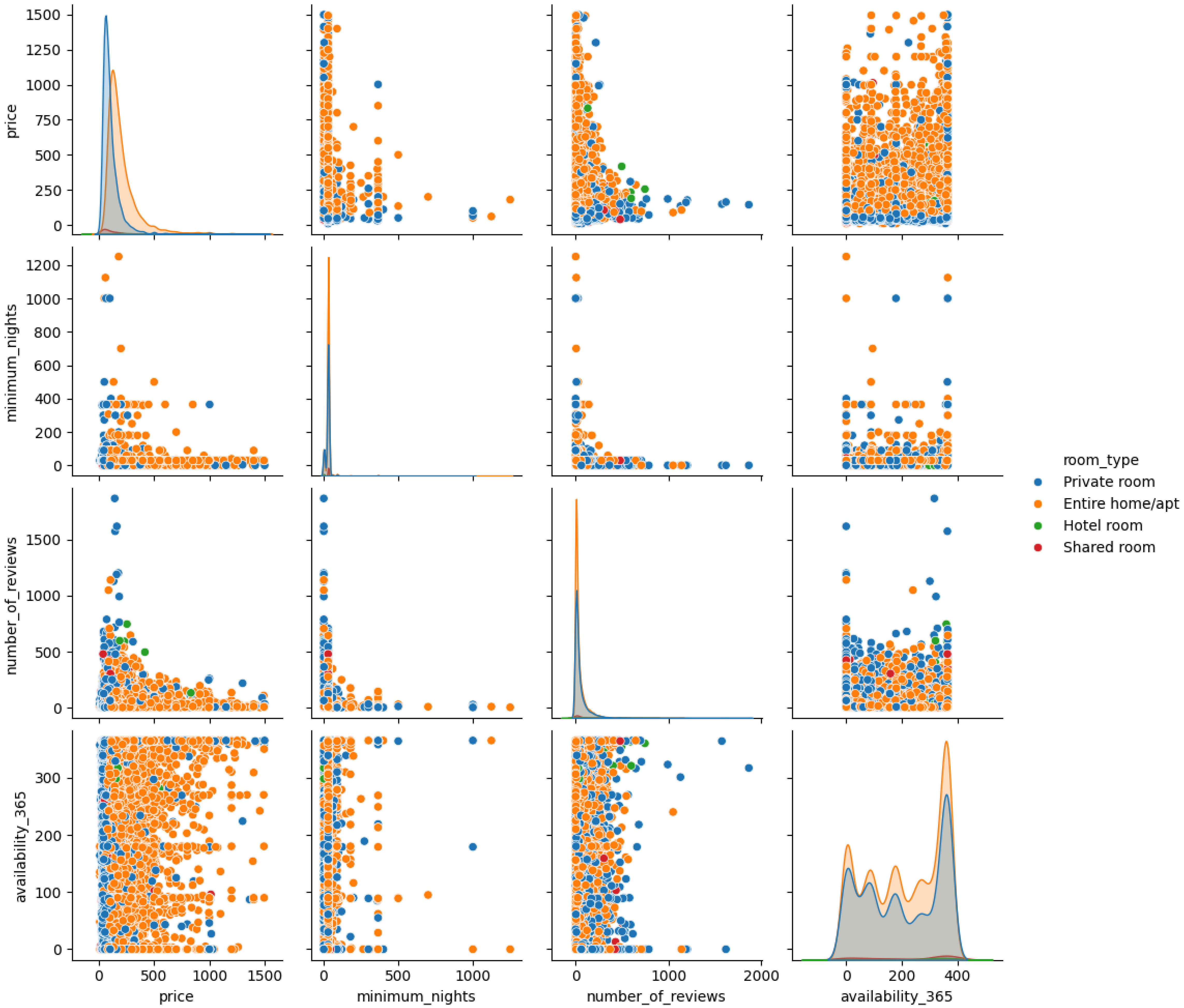


In [68]:
```
df.dtypes
```

Out[68]:
```
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
price per bed                     float64
dtype: object
```

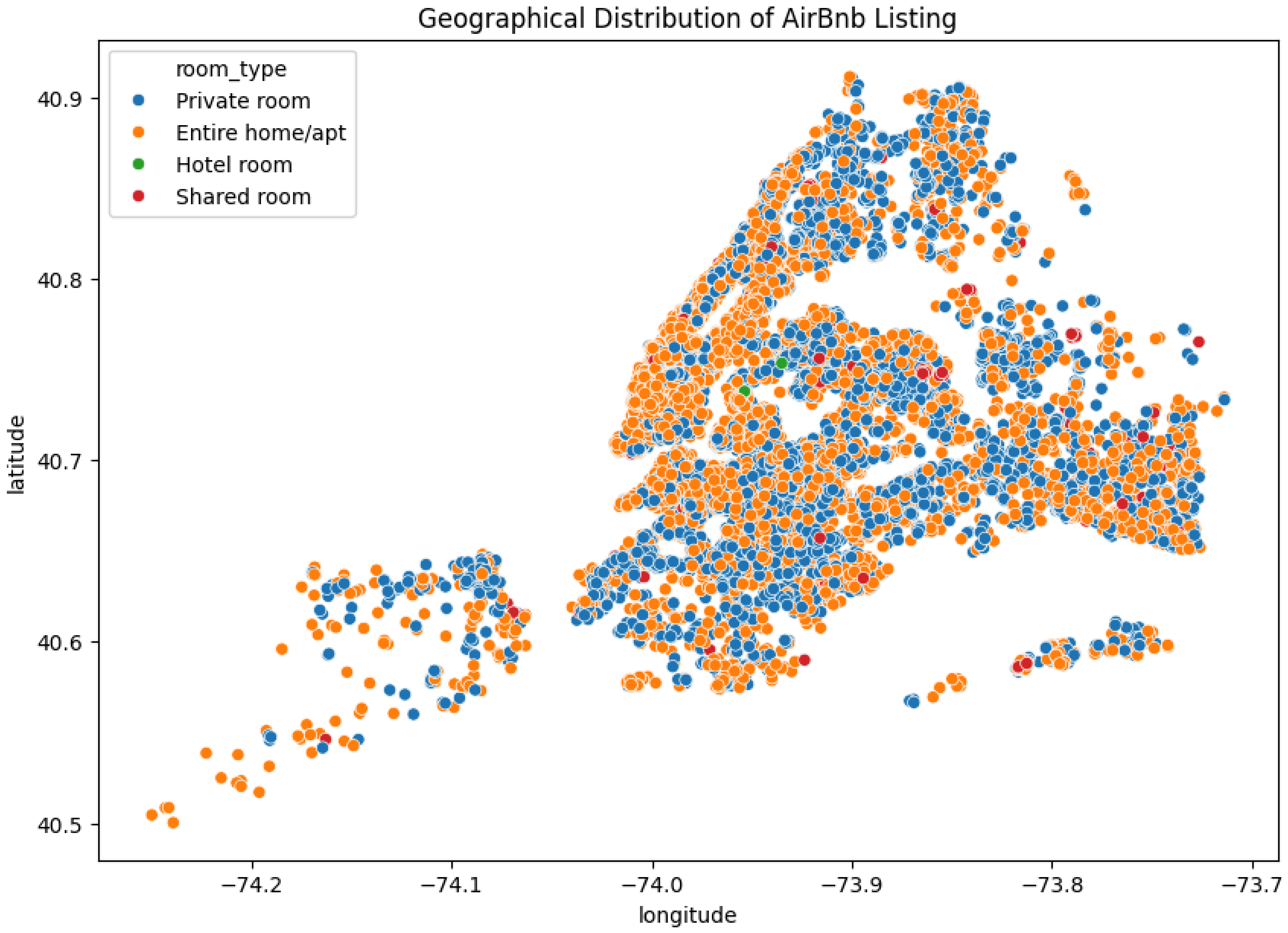
In [70]:
```
sns.pairplot(data=df, vars=['price', 'minimum_nights', 'number_of_reviews', 'availability_365'], hue='room_type')
```

Out[70]:
```
<seaborn.axisgrid.PairGrid at 0x1495a4dd0>
```



In [75]:

```
#Geographical Distribution of AirBnb Listing
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()
```



In [76]:

```
df.dtypes
```

Out[76]:

```
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
price per bed                     float64
dtype: object
```

In [80]:

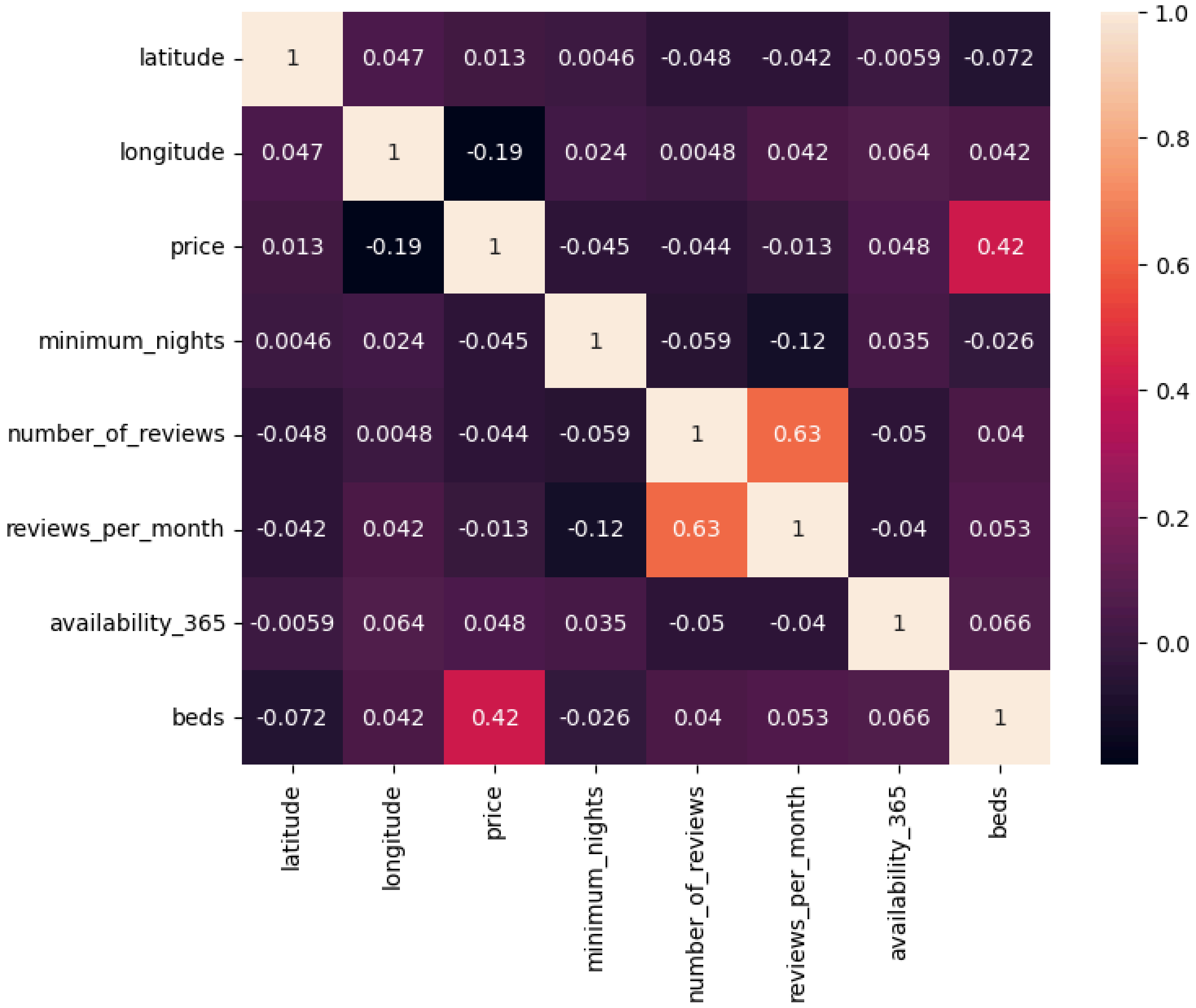
```
# heat map - correlation of one variable with others for numerical column

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

plt.figure(figsize=(8, 6))
sns.heatmap(data=corr, annot=True)
```

Out[80]:

```
<Axes: >
```


latitude
longitude
price
minimum_nights
number_of_reviews
reviews_per_month
availability_365
beds
1 0.047 0.013 0.0046 -0.048 -0.042 -0.0059 -0.072
0.047 1 -0.19 0.024 0.0048 0.042 0.064 0.042
0.013 -0.19 1 -0.045 -0.044 -0.013 0.048 0.42
0.0046 0.024 -0.045 1 -0.059 -0.12 0.035 -0.026
-0.048 0.0048 -0.044 -0.059 1 0.63 -0.05 0.04
-0.042 0.042 -0.013 -0.12 0.63 1 -0.04 0.053
-0.0059 0.064 0.048 0.035 -0.05 -0.04 1 0.066
-0.072 0.042 0.42 -0.026 0.04 0.053 0.066 1
latitude
longitude
price
minimum_nights
number_of_reviews
reviews_per_month
availability_365
beds
1.0
0.8
0.6
0.4
0.2
0.0