Double-click (or enter) to edit

Task 1. Importing All Dependencies

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

%matplotlib inline
```

Task 2 : Loading Datasets

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

Task 3: Initial Exploration

```
data.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_ty
0	1.312228e+06	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382.0	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Priv ro
1	4.527754e+07	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835.0	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100	En home/
2	9.710000e+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354.0	Joshua	Manhattan	Chelsea	40.750764	-73.994605	En home/
3	3.857863e+06	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271.0	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500	Priv ro
4	4.089661e+07	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963.0	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	En home/

5 rows × 22 columns

```
data.tail()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	roo
10139	2.090054e+07	Home in Brooklyn · ★4.91 · 4 bedrooms · 5 beds...	150159094.0	Domonique	Brooklyn	Bedford-Stuyvesant	40.69042	-73.93488	h
10140	8.980000e+17	Home in Queens · ★5.0 · 1 bedroom · 1 bed · 1 ...	351627173.0	Kevin	Queens	Elmhurst	40.72805	-73.88026	h
10141	7.570000e+17	Home in Queens Village · ★4.57 · 1 bedroom · 1...	139200985.0	Nashita	Queens	Queens Village	40.72603	-73.74894	
10142	1.088832e+07	Rental unit in Bronx · ★4.62 · 1 bedroom · 2 b...	971075.0	Jabari	Bronx	Mount Hope	40.85080	-73.90218	h
10143	3.921132e+07	Rental unit in New Yor	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 22 columns

```
data.shape
```

```
(10144, 22)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10144 entries, 0 to 10143
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              10144 non-null  float64
 1   name                            10144 non-null  object
 2   host_id                         10143 non-null  float64
 3   host_name                       10143 non-null  object
 4   neighbourhood_group             10143 non-null  object
 5   neighbourhood                   10143 non-null  object
 6   latitude                        10143 non-null  float64
 7   longitude                       10143 non-null  float64
 8   room_type                       10143 non-null  object
 9   price                           10143 non-null  float64
 10  minimum_nights                  10143 non-null  float64
 11  number_of_reviews               10143 non-null  float64
 12  last_review                     10143 non-null  object
 13  reviews_per_month               10143 non-null  float64
 14  calculated_host_listings_count  10143 non-null  float64
 15  availability_365                10143 non-null  float64
 16  number_of_reviews_ltm           10143 non-null  float64
 17  license                         10143 non-null  object
```

```
 18  rating                          10143 non-null  object
 19  bedrooms                        10143 non-null  object
 20  beds                            10143 non-null  float64
 21  baths                           10143 non-null  object
dtypes: float64(12), object(10)
memory usage: 1.7+ MB
```

```
# Statistical Summary
data.describe()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	rev
count	1.014400e+04	1.014300e+04	10143.000000	10143.000000	10143.000000	10143.000000	10143.000000	
mean	2.950021e+17	1.754408e+08	40.726913	-73.942560	204.087548	29.127083	42.344671	
std	3.879457e+17	1.729280e+08	0.059399	0.057681	1440.465720	39.816585	75.827590	
min	5.121000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	1.000000	
25%	2.753530e+07	1.948131e+07	40.685258	-73.981455	80.000000	30.000000	4.000000	
50%	4.964309e+07	1.100174e+08	40.722350	-73.950430	126.000000	30.000000	14.000000	
75%	7.100000e+17	3.133223e+08	40.762605	-73.920700	200.000000	30.000000	48.000000	
max	1.050000e+18	5.489914e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	

Task 4: Data Cleaning

```
data.isnull().sum()

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

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

	0
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

```
# dealing with duplicates rows
data.duplicated().sum()

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

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

```
np.int64(0)
```

```
# 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
```

	0
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	float64
baths	object

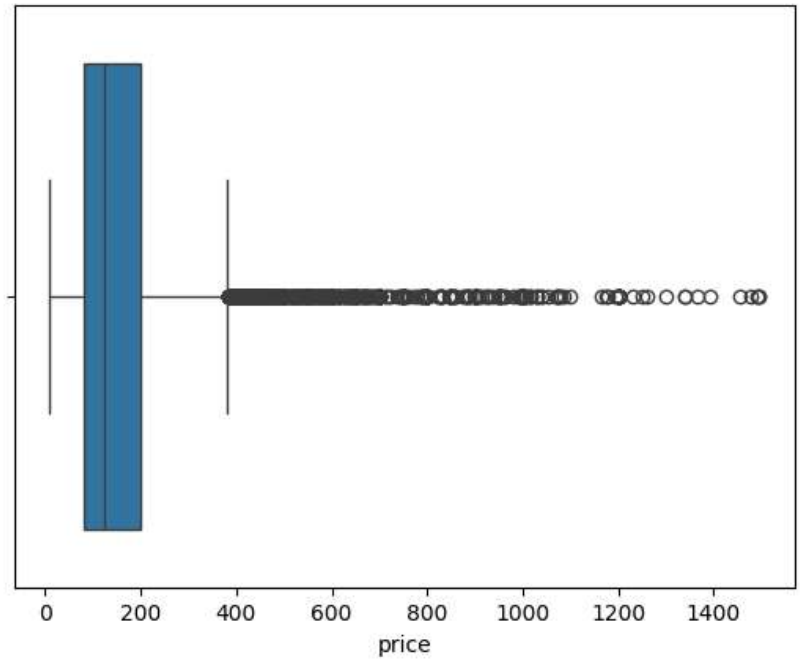
dtype: object

EDA Task 5: Data Analysis

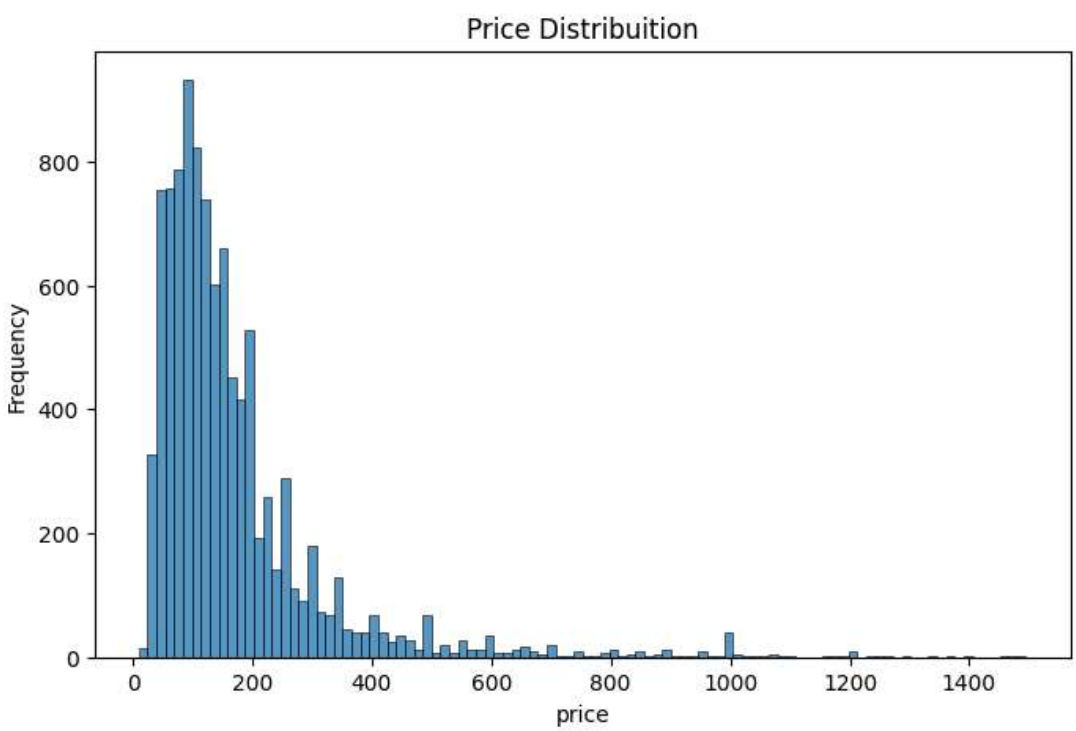
Univariate Analysis

```
# idenfying outliers in price

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

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

```
<Axes: xlabel='price'>
```




```
#Price distribuion

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



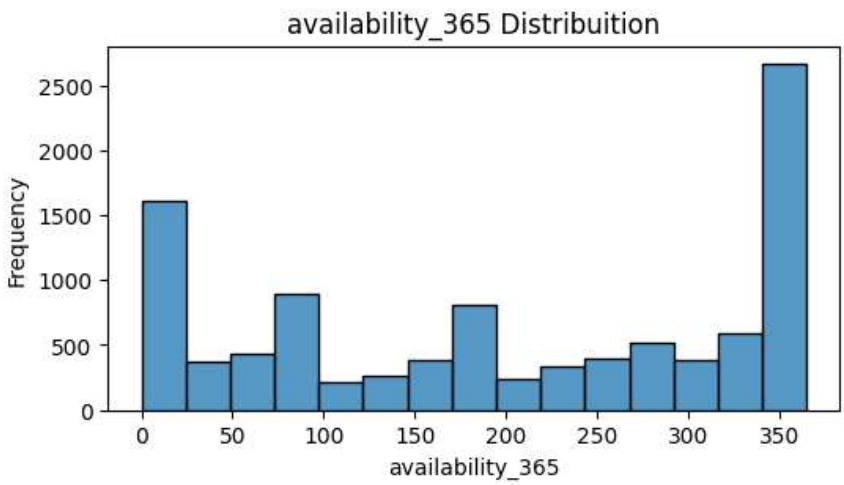
Double-click (or enter) to edit

```
df.dtypes
```

	0
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	float64
baths	object

dtype: object

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




```
data.dtypes
```

	0
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	float64
baths	object

dtype: object

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

neighbourhood_group	price
Bronx	103.477974
Brooklyn	156.338383
Manhattan	210.416373
Queens	121.045788
Staten Island	137.037313

dtype: float64

```
# ['price per bed']

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

```
/tmp/ipython-input-21-2324310957.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#return
  df['price per bed']= df['price']/df['beds']
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382.0	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610
1	45277537.0	Rental unit in New York · ★4.67 · 2 bedrooms · ...	51501835.0	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354.0	Joshua	Manhattan	Chelsea	40.750764	-73.994605
3	3857863.0	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271.0	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500
4	40896611.0	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963.0	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600

5 rows × 23 columns

```
df.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382.0	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610
1	45277537.0	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835.0	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354.0	Joshua	Manhattan	Chelsea	40.750764	-73.994605
3	3857863.0	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271.0	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500
4	40896611.0	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963.0	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600

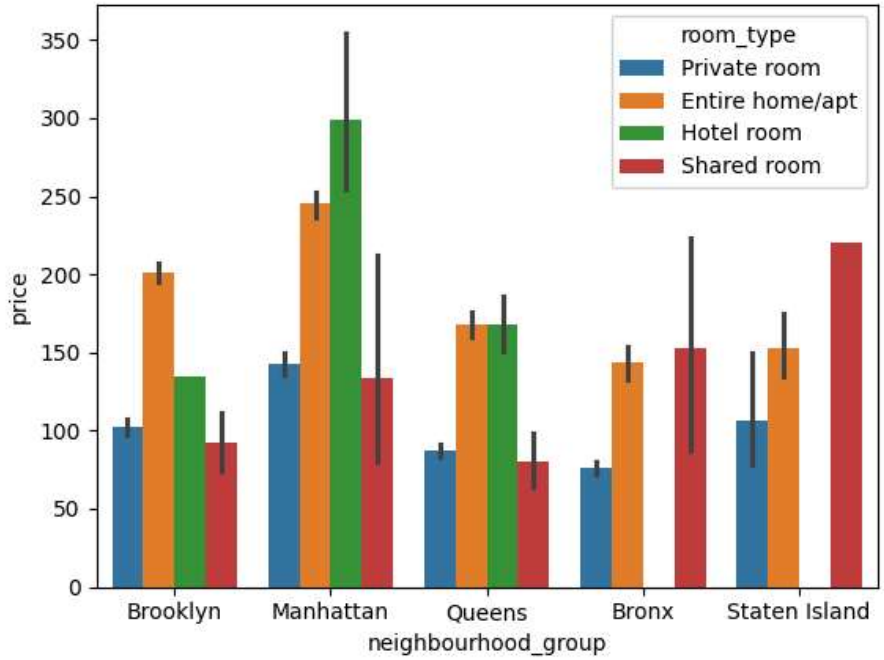
5 rows × 23 columns

```
df.columns
```

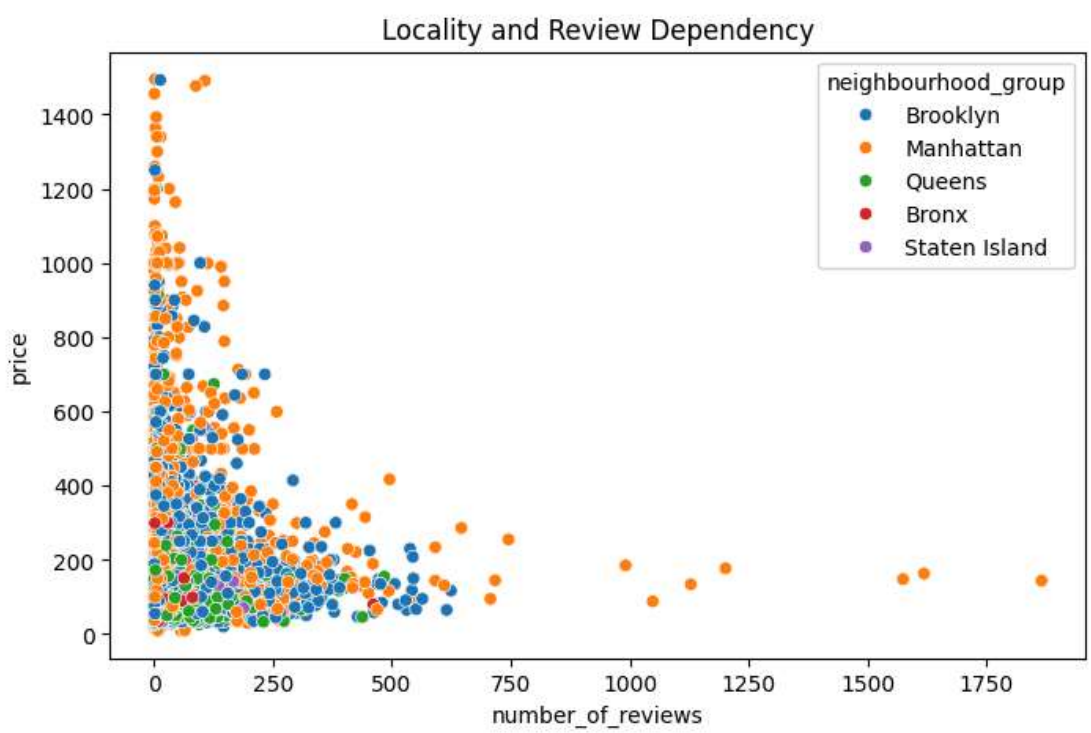
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'number_of_reviews_ltm',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

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

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




```
# 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()
```



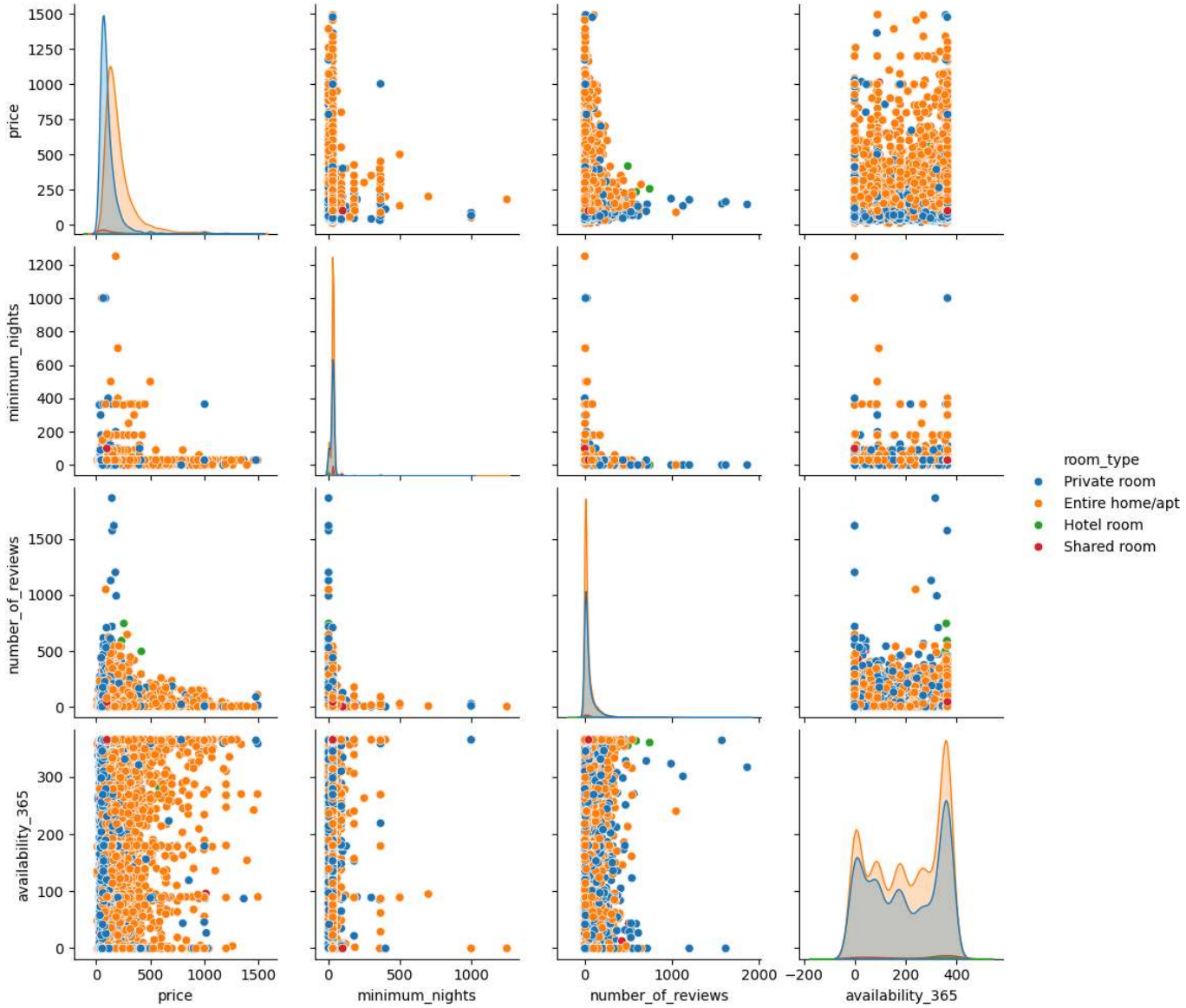

```
df.dtypes
```

	0
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	float64
baths	object
price per bed	float64

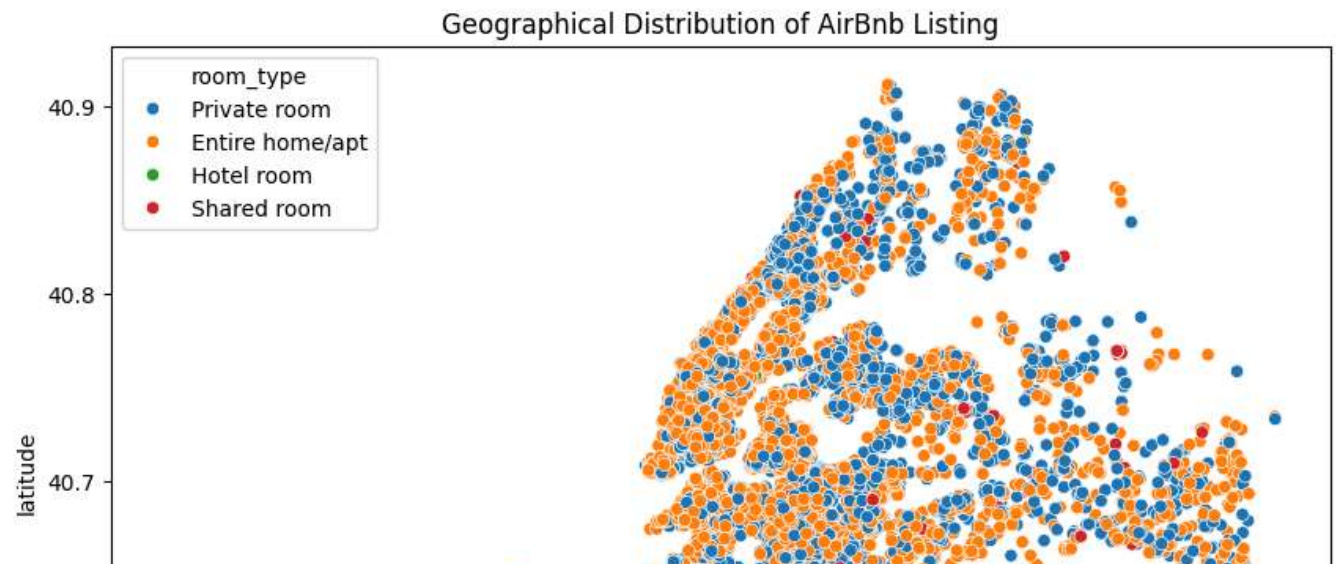
dtype: object

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

```
<seaborn.axisgrid.PairGrid at 0x78df58a6b010>
```




```
#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()
```




```
df.dtypes
```

id	object
name	object
host_id	object
host_name	object
neighbourhood_group	object
neighbourhood	object