1. Importing Dependencies & Loading Dataset

```
#Importing Dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#Loading Dataset
data=pd.read_csv("/content/sample_data/airbnb_dataset.csv",encoding_errors='ignore')
```

2. Initial Exploration

```
#Print top 5 rows
data.head()
```

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

5 rows × 22 columns

```
#Print last 5 rows
data.tail()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	r
20765	24736896.0	Rental unit in New York · ★4.75 · 1 bedroom · ...	186680487	Henry D	Manhattan	Lower East Side	40.711380	-73.991560	
20766	2835711.0	Rental unit in New York · ★4.46 · 1 bedroom · ...	3237504	Aspen	Manhattan	Greenwich Village	40.730580	-74.000700	
20767	51825274.0	Rental unit in New York · ★4.93 · 1 bedroom · ...	304317395	Jeff	Manhattan	Hell's Kitchen	40.757350	-73.993430	
20768	783000000000000000.0	Rental unit in New York · ★5.0 · 1 bedroom · 1...	163083101	Marissa	Manhattan	Chinatown	40.713750	-73.991470	
20769	566000000000000000.0	Rental unit in Queens · ★4.89 · 1 bedroom · 1 ...	93827372	Glenroy	Queens	Rosedale	40.658874	-73.728651	

5 rows × 22 columns

```
#Print total Rows & Columns
data.shape
```

```
(20724, 22)
```

```
# Display concise summary of the DataFrame
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 20724 entries, 0 to 20769
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20724 non-null  object
 1   name                            20724 non-null  object
 2   host_id                         20724 non-null  object
 3   host_name                       20724 non-null  object
 4   neighbourhood_group             20724 non-null  object
 5   neighbourhood                   20724 non-null  object
 6   latitude                        20724 non-null  float64
 7   longitude                       20724 non-null  float64
 8   room_type                       20724 non-null  object
 9   price                           20724 non-null  float64
 10  minimum_nights                  20724 non-null  float64
 11  number_of_reviews               20724 non-null  float64
 12  last_review                     20724 non-null  object
 13  reviews_per_month               20724 non-null  float64
 14  calculated_host_listings_count  20724 non-null  float64
 15  availability_365                20724 non-null  float64
 16  number_of_reviews_ltm           20724 non-null  float64
 17  license                         20724 non-null  object
```

```
 18  rating                          20724 non-null  object
 19  bedrooms                        20724 non-null  object
 20  beds                            20724 non-null  int64
 21  baths                           20724 non-null  object
dtypes: float64(9), int64(1), object(12)
memory usage: 3.6+ MB
```

```
# Generate descriptive statistics for numeric columns (count, mean, std, min, quartiles, max)
data.describe()
```

	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_
count	20724.000000	20724.000000	20724.000000	20724.000000	20724.000000	20724.000000	
mean	40.726843	-73.939155	187.732195	28.566396	42.592646	1.257529	
std	0.060320	0.061442	1023.539393	33.560272	73.534712	1.905221	
min	40.500314	-74.249840	10.000000	1.000000	1.000000	0.010000	
25%	40.684150	-73.980760	80.000000	30.000000	4.000000	0.210000	
50%	40.722937	-73.949599	125.000000	30.000000	14.000000	0.650000	
75%	40.763132	-73.917430	199.000000	30.000000	49.000000	1.800000	
max	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	75.490000	

3. Data Cleaning

```
#Checking null values
data.isnull().sum()
```

	0
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

```
#Dropping all null rows
data.dropna(inplace=True)
```

```
#Checking total Duplicate rows
data.duplicated().sum()
```

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

```
#printing all duplicate rows
data[data.duplicated()]
```

id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_re

0 rows × 22 columns

```
#Dropping all duplicate rows
data.drop_duplicates(inplace=True)
```

```
#Checking Data type of Columns
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	int64
baths	object

dtype: object

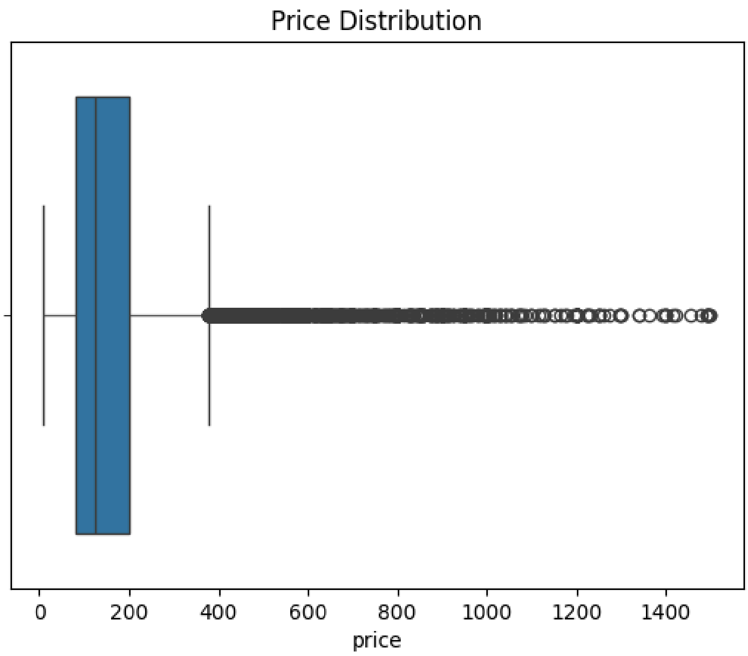
```
#Changing the data-type of column 'id' to object
data['id']=data['id'].astype(object)
```

```
#Changing the data-type of column 'host_id' to object
data['host_id']=data['host_id'].astype(object)
```

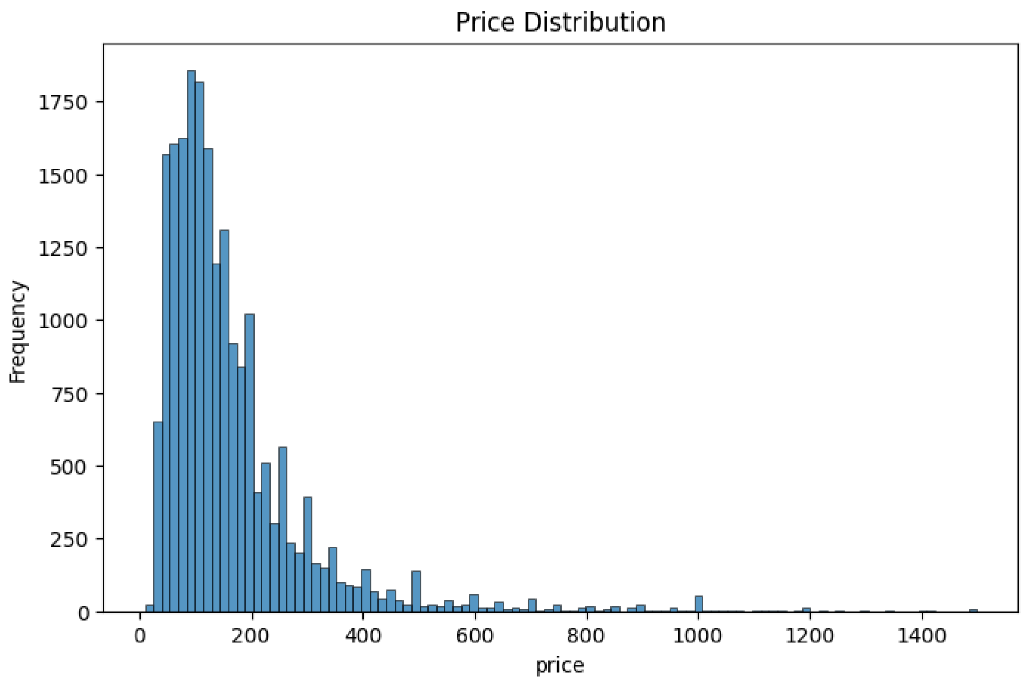
4. Data Analysis

4.A. Univariate Analysis

```
#new dataframe with price less than 1500 to remove price outliers
df=data[data['price']<1500]
#Boxplot of Price Distribution
plt.title('Price Distribution')
sns.boxplot(data=df,x='price')
plt.show()
```

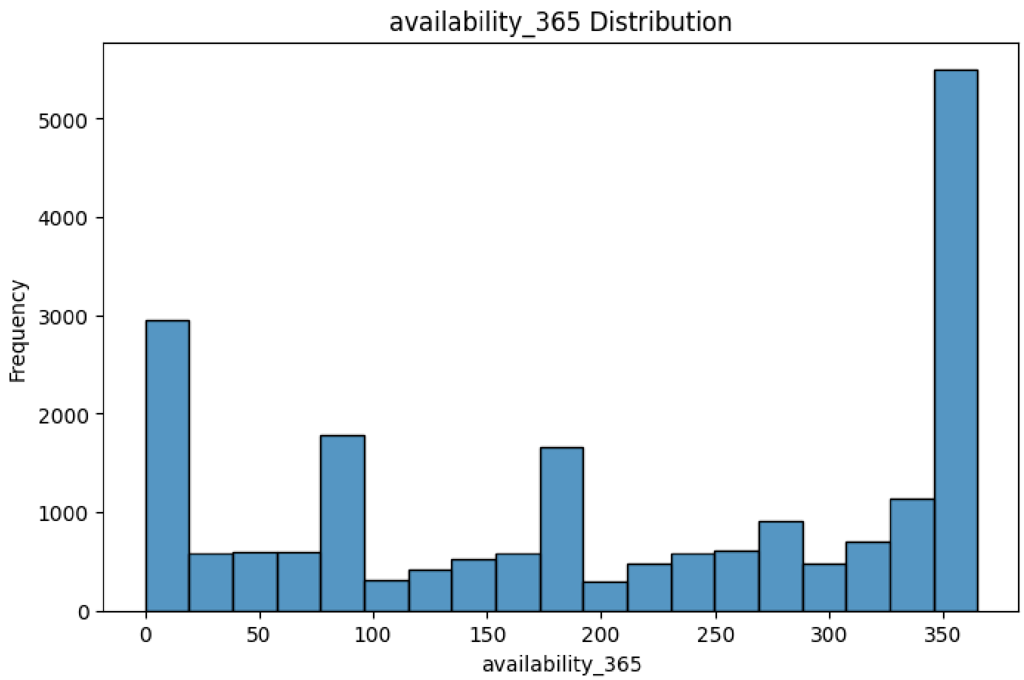



```
#Histogram of Price Distribution
plt.figure(figsize=(8,5))
plt.title('Price Distribution')
sns.histplot(data=df,x='price',bins=100)
plt.ylabel('Frequency')
plt.show()
```




```
#Histogram of availability_365 Distribution
plt.figure(figsize=(8,5))
```

```
plt.title('availability_365 Distribution')
sns.histplot(data=df,x='availability_365')
plt.ylabel('Frequency')
plt.show()
```




```
#Average Price of each neighbourhood group
df.groupby(by='neighbourhood_group')['price'].mean()
```

neighbourhood_group	price
Bronx	107.990506
Brooklyn	155.138317
Manhattan	204.076470
Queens	121.681939
Staten Island	118.780069

dtype: float64

4.B. Feature Engineering

```
#Adding a new column named 'price per bed' to dataframe 'df'
df['price_per_bed']=df['price']/df['beds']
```

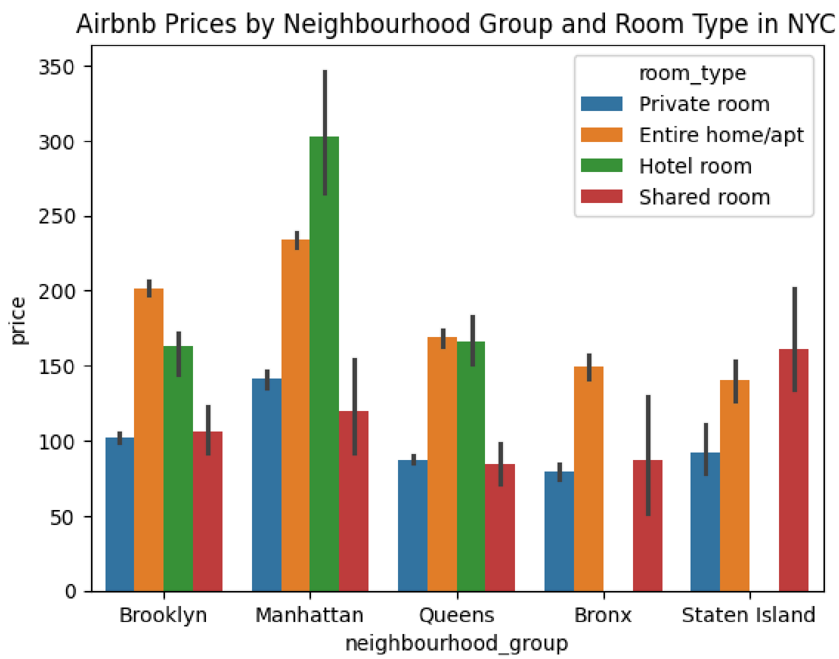
```
#Average Price per bed of each neighbourhood group
df.groupby(by='neighbourhood_group')['price_per_bed'].mean()
```

neighbourhood_group	price_per_bed
Bronx	74.713639
Brooklyn	99.788493
Manhattan	138.662489
Queens	76.336210
Staten Island	67.728101

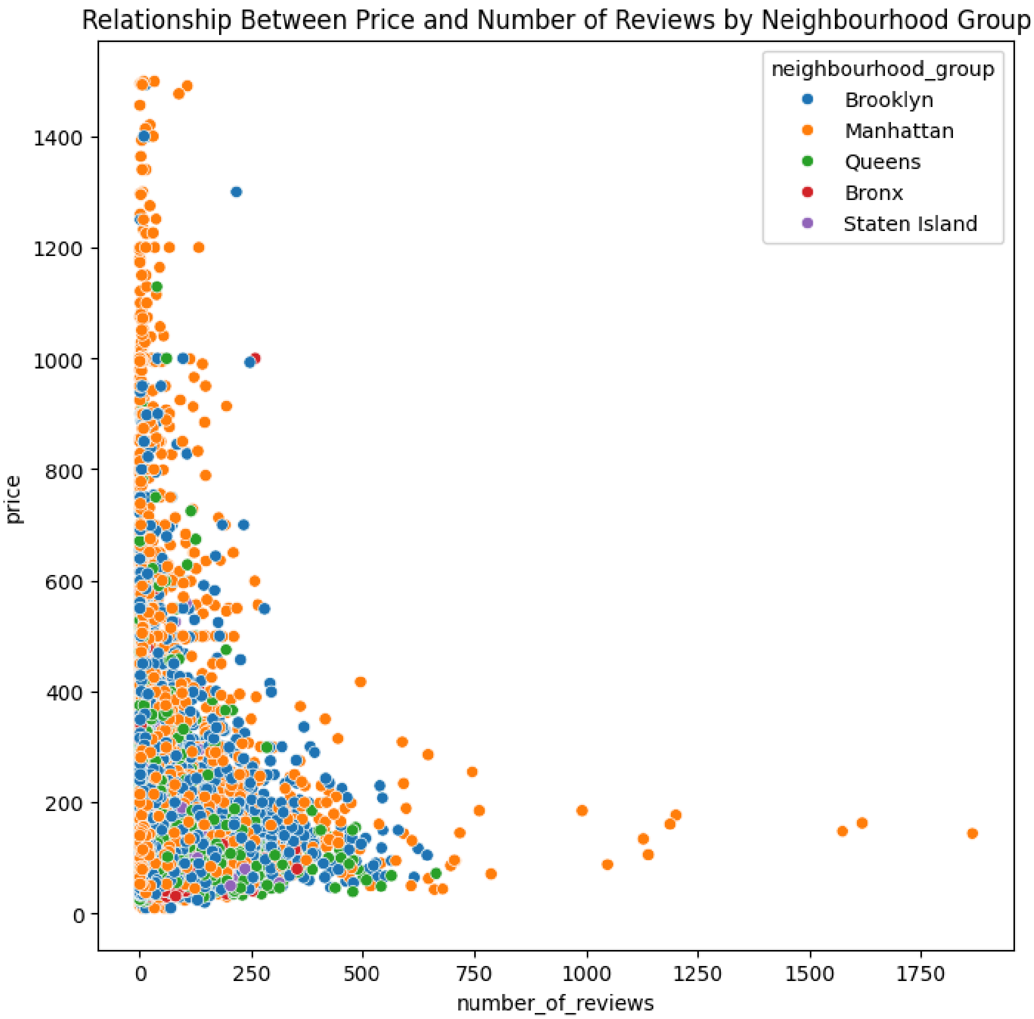
dtype: float64

4.C. Bivariate Analysis

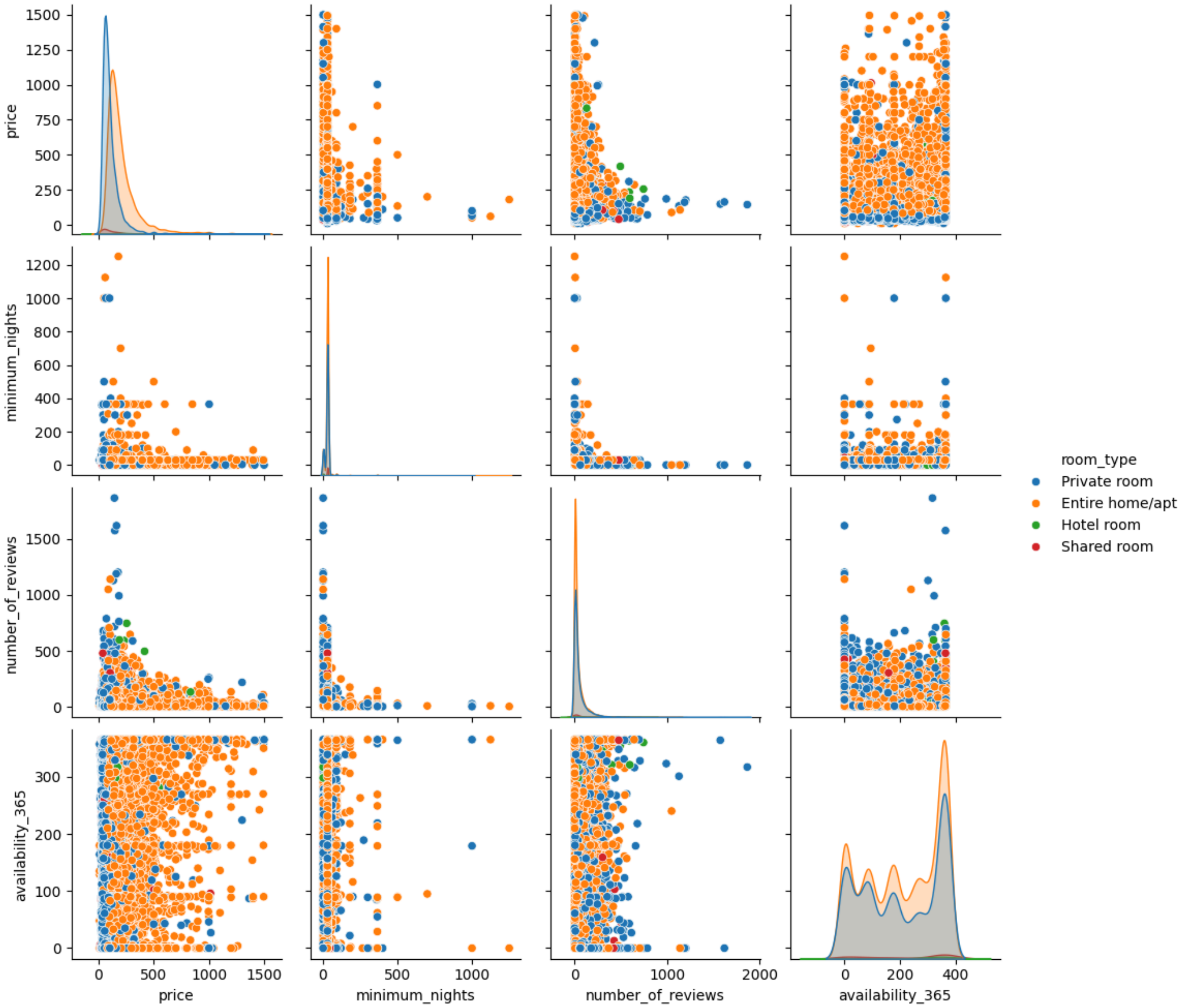
```
#Barplot of Airbnb Prices by Neighbourhood Group and Room Type in NYC
sns.barplot(data=df,x='neighbourhood_group',y='price',hue='room_type')
plt.title('Airbnb Prices by Neighbourhood Group and Room Type in NYC')
plt.show()
```




```
#Relationship Between Price and Number of Reviews by Neighbourhood Group
plt.figure(figsize=(8,8))
plt.title('Relationship Between Price and Number of Reviews by Neighbourhood Group')
sns.scatterplot(data=df,x='number_of_reviews',y='price', hue='neighbourhood_group')
plt.show()
```




```
#Pairwise Relationships Between Listing Attributes by Room Type
sns.pairplot(data=df,vars=['price','minimum_nights','number_of_reviews','availability_365'], hue='room_type')
plt.show()
```




```
# Geographical distribution of Airbnb listings
plt.figure(figsize=(10,7))
plt.title('Geographical distribution of Airbnb listings')
sns.scatterplot(data=df,x='longitude',y='latitude',hue='room_type')
plt.show()
```