

Task 1. Importing All Dependencies

```
[1]:  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
%matplotlib inline
```

Task 2: Loading Datasets

```
[2]: data = pd.read_csv('c:/Projects/Rajat Saxena/Data & Analytics Package/Data Projects/Python Project/new york listings 2024.csv', encoding='latin-1')
```

Task 3: Initial Exploration

[3]:	data.head()												
[3]:	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	reviews_per_month
0	1.312228e+06	Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	55.0	...	20/12/15	0.03

```
[5]: data.shape
```

```
[5]: (20770, 22)
```

```
[6]: 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
```

# Statistical Summary							
data.describe()							
	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.000000	20763.000000
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.558493	42.610605
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.532397	73.523401
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	0.010000
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.000000	4.000000
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.000000	14.000000
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.000000	49.000000
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000

Task 4: Data Cleaning

```
[8]: data.isnull().sum()
[8]: id          0
      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_1tm               7
license                         0
rating                          0
bedrooms                        0
beds                            0
baths                           0
dtype: int64
```

Handling Null Values

```
[9]: data['neighbourhood'] = data['neighbourhood'].fillna('Unknown')

[10]: data['latitude'] = data['latitude'].ffill()

[11]: data['longitude'] = data['longitude'].ffill()

[12]: data['room_type'] = data['room_type'].ffill()
```

```
[11]: data['longitude'] = data['longitude'].ffill()
```

```
[12]: data['room_type'] = data['room_type'].ffill()
```

Filling Median Values in Place Of null Values

```
[13]: data['price'] = data['price'].fillna(data['price'].median())
```

Filling mean in place of null values

```
[14]: data['minimum_nights'] = data['minimum_nights'].fillna(data['minimum_nights'].mean())
```

Filling mean in place of null values

```
[15]: data['number_of_reviews'] = data['number_of_reviews'].fillna(data['number_of_reviews'].mean())
```

Changing String To DateTime Format

```
[16]: data['last_review'] = pd.to_datetime(data['last_review'])
```

```
c:\Users\ASUS\AppData\Local\Temp\ipykernel_17924\897153227.py:1: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.  
data['last_review'] = pd.to_datetime(data['last_review'])
```

```
[17]: data['last_review'] = data['last_review'].ffill()
```

```
[18]: data['reviews_per_month'] = data['reviews_per_month'].fillna(data['reviews_per_month'].mean())
```

```
[19]: data['calculated_host_listings_count'] = data['calculated_host_listings_count'].fillna(data['calculated_host_listings_count'].median())

[20]: data['availability_365'] = data['availability_365'].fillna(data['availability_365'].mean())

[21]: data['number_of_reviews_ltm'] = data['number_of_reviews_ltm'].fillna(data['number_of_reviews_ltm'].mean())

[22]: data.isnull().sum()

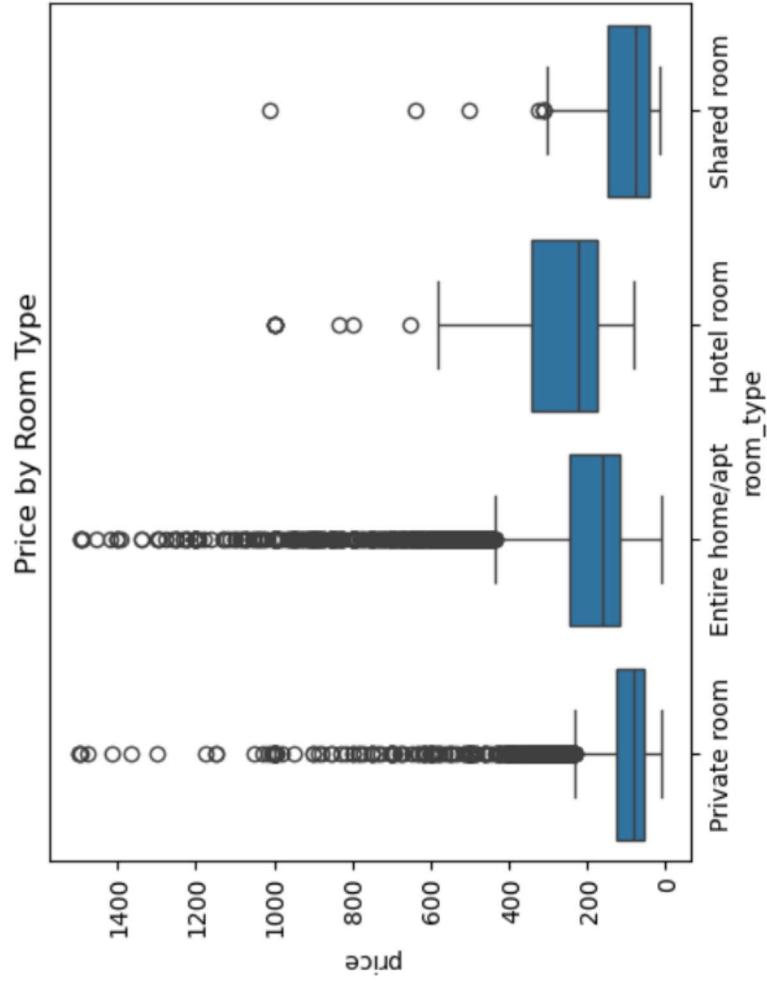
[22]: id          0
      name        0
      host_id      0
      host_name     0
      neighbourhood_group 0
      neighbourhood    0
      latitude       0
      longitude      0
      room_type      0
      price          0
      minimum_nights 0
      number_of_reviews 0
      last_review     0
      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
      dtvne: int64
```

```
[23]: # type casting
# changing data types
data['id'] = data['id'].astype(object)

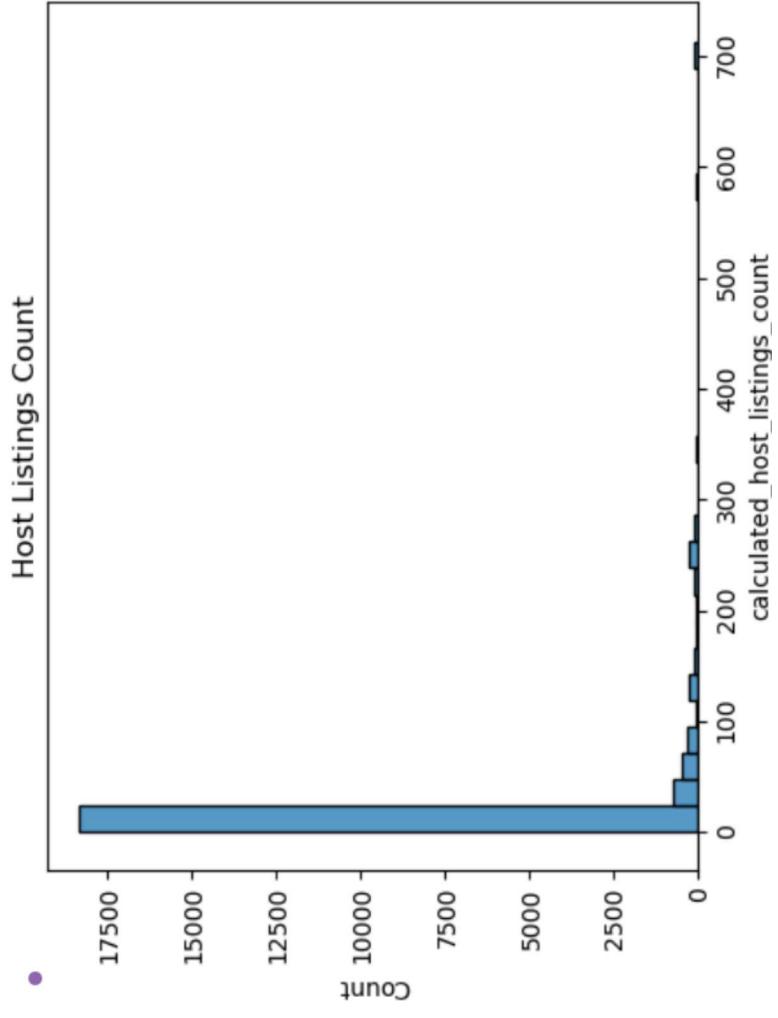
data['host_id'] = data['host_id'].astype(object)
data.dtypes
```

```
[23]: 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    datetime64[ns]
      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
```

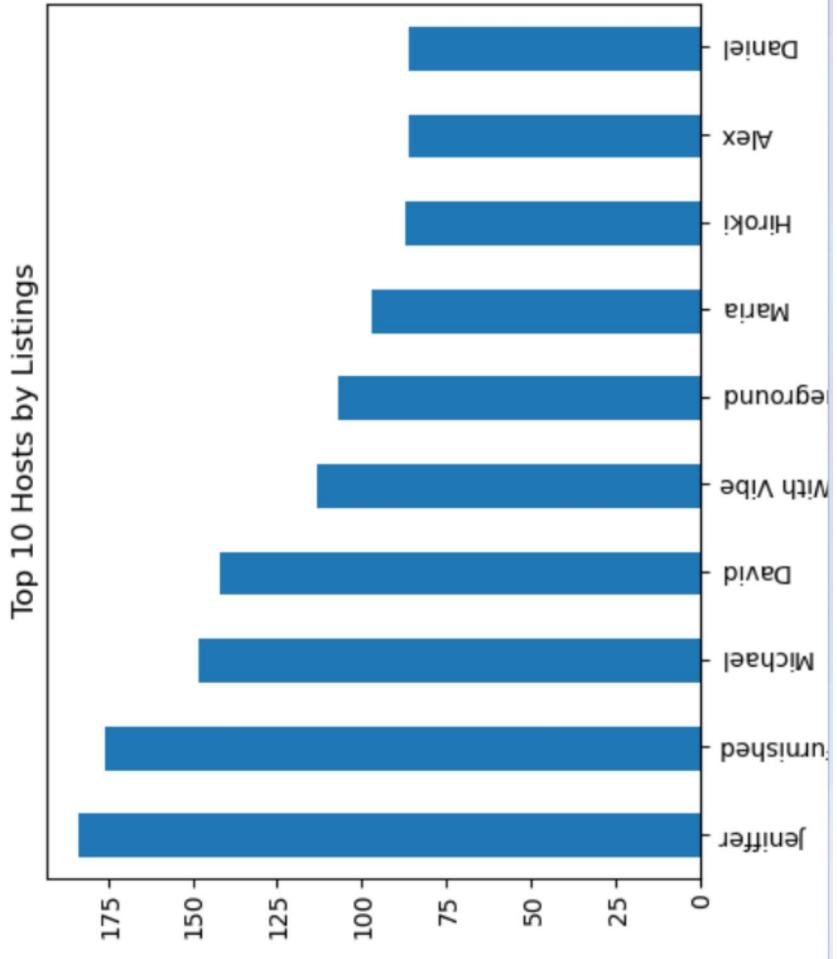
```
[43]: sns.boxplot(x='room_type', y='price', data=df)
plt.title("Price by Room Type")
plt.show()
```



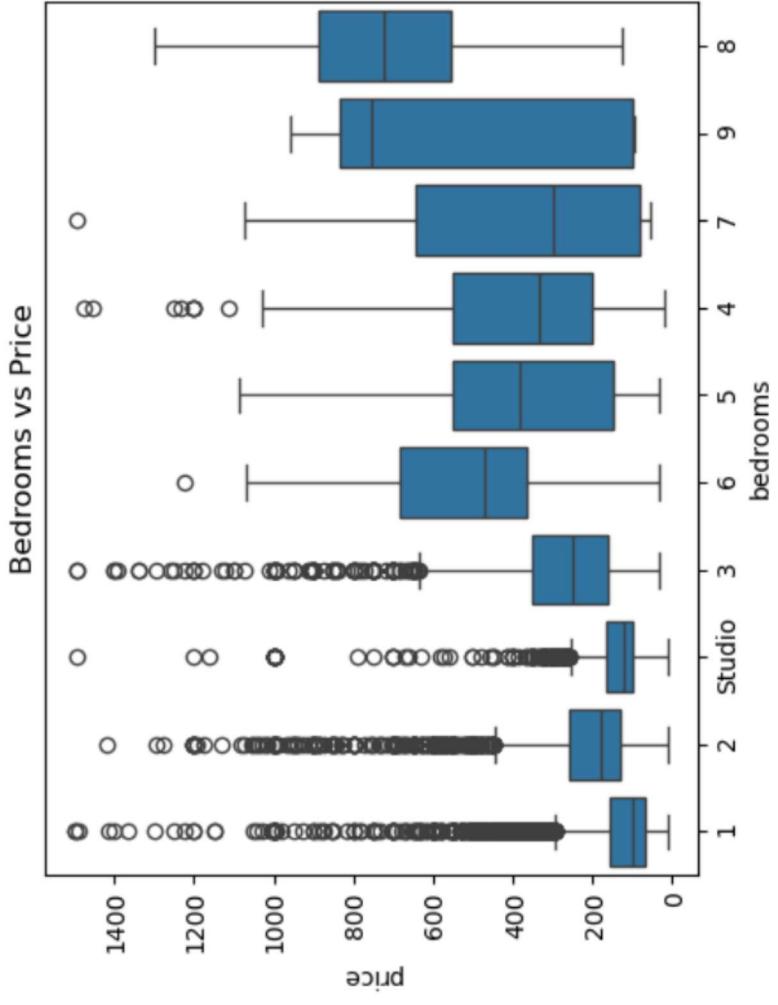
```
[50]:  
sns.histplot(df[ 'calculated_host_listings_count' ], bins=30)  
plt.title("Host Listings Count")  
plt.show()
```



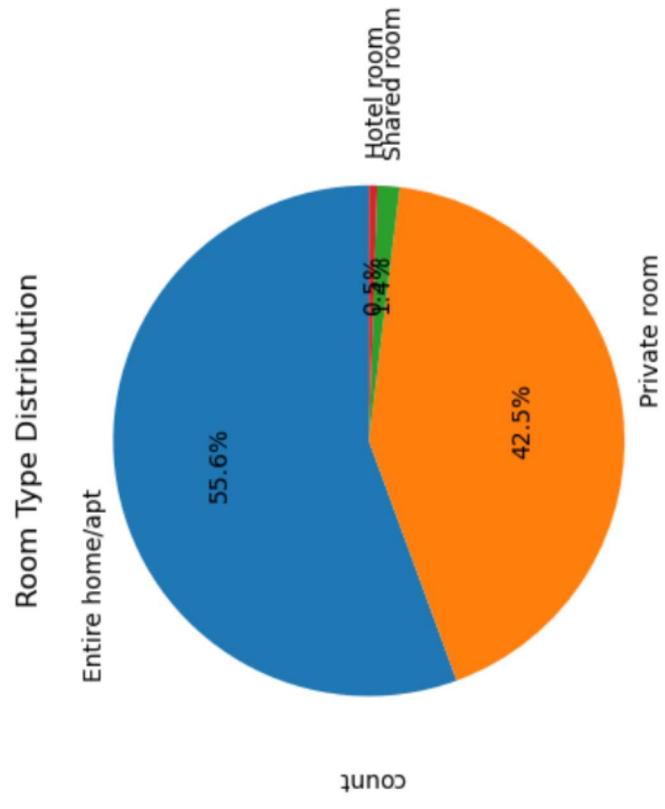
```
plt.title("Top 10 Hosts by Listings")
plt.show()
```



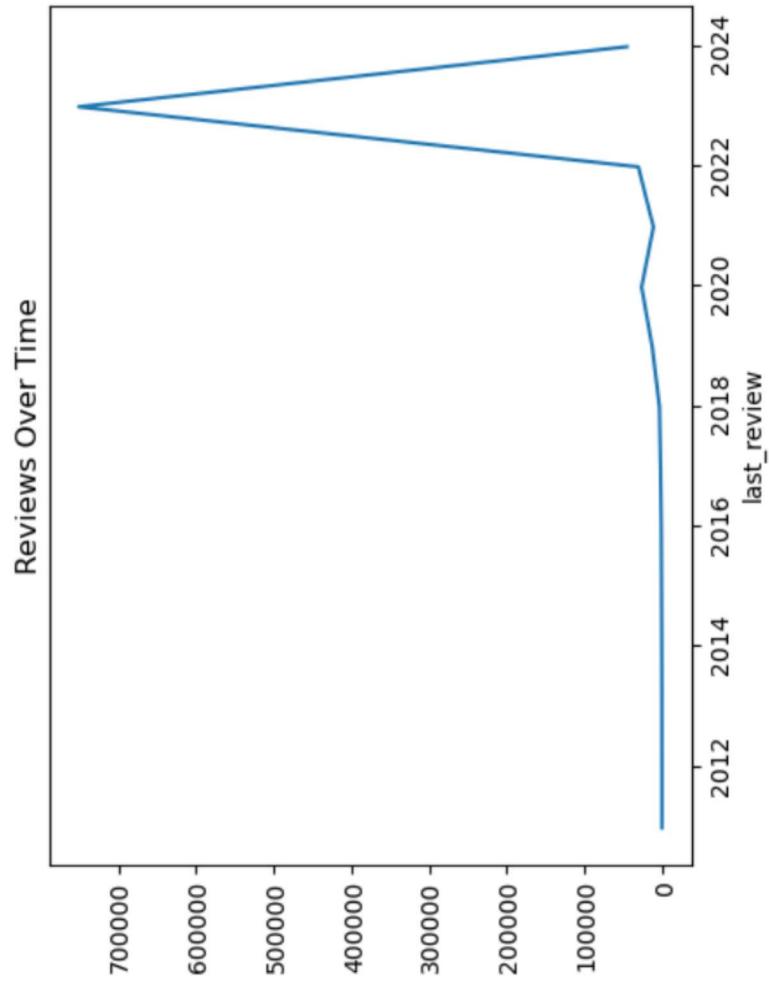
```
[52]:  
    sns.boxplot(x='bedrooms', y='price', data=df)  
    plt.title("Bedrooms vs Price")  
    plt.show()
```



```
[53]:  
df['room_type'].value_counts().plot(kind='pie', autopct='%1.1f%%')  
plt.title("Room Type distribution")  
plt.show()
```



```
[54]:  
df.groupby(df['last_review'].dt.year)[['number_of_reviews']].sum().plot()  
plt.title("Reviews Over Time")  
plt.show()
```



AirBnb Python Project - AirBnB Listing 2024(New York)

Steps

1. Importing all dependencies (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 Datasets

```
In [3]: data = pd.read_csv('new_york_listings_2024.csv', encoding='errors ignore')
```

Task 3: Initial Exploration

```
In [6]: data.head()
```

	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	item	license	rating	bedrooms	beds	baths
0	1.312228e+06	Rental unit in Brooklyn - ★3.0 - 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	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	144.0 ...	01/05/23	0.24	139.0	364.0	2.0	No	4.67	2	1	1	
2	9.710000e+17	Rental unit in New York - ★4.17 - 1 bedroom...	52087154	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire	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	4.64	1	1	1	
4	4.089661e+07	Condo in New York - ★4.91 - Studio - 1 bed - 1...	61391963	Say With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	Entire	85.0 ...	03/12/23	0.24	133.0	335.0	3.0	No	4.91	Studio	1	1	

5 rows × 22 columns

```
In [7]: data.tail()
```

	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	item	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	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	40.75750	-73.993430	Entire	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 - ★4.0 - 1 bedroom - 1...	163083101	Marissa	Manhattan	Chinatown	40.713750	-73.991470	Entire	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	40.65874	-73.728651	Private room	102.0 ...	10/12/23	4.5C	1.0	0.0	62.0	OSE-STREG-0000013	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    object  
 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    float64 
 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    float64 
 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_1m 20763 non-null    float64 
 17  license          20770 non-null    object  
 18  rating           20770 non-null    float64 
 19  bedrooms         20770 non-null    float64 
 20  beds             20770 non-null    float64 
 21  baths             20770 non-null    object  
dtypes: float64(10), int64(2), object(10)
memory usage: 3.5+ MB

```

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

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365	number_of_reviews_1m	beds
count	2.07700e+04	2.07700e+04	2.076300000	2.073600000	2.076300000	2.076300000	2.076300000	2.076300000	2.076300000	2.076300000	2.076300000	2.076300000
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.558493	42.610605	1.257589	18.866686	20.6067957	10.848962	1.723592
std	3.901221e+17	1.725674e+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	0.100000	0.100000	0.000000	0.000000	0.000000	1.000000
25%	2.07250e+07	2.041184e+07	40.684159	-73.98075	80.000000	30.000000	4.000000	0.210000	0.100000	87.000000	1.000000	1.000000
50%	4.992852e+07	1.08690e+08	40.722890	-73.94957	125.000000	30.000000	14.000000	0.650000	2.000000	25.000000	3.000000	1.000000
75%	7.22000e+17	3.14397e+08	40.763106	-73.917475	199.000000	30.000000	49.000000	1.800000	5.000000	153.000000	15.000000	2.000000
max	1.050000e+18	5.04035e+08	40.911147	-73.73650	100000.000000	1250.000000	1865.000000	75.490000	713.000000	365.000000	1075.000000	42.000000

Task 4: Data Cleaning

```

In [13]: 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_1m 0
license      0
rating       0
bedrooms    0
baths        0
dtype: int64

```

```

In [15]: # dealing with duplicates rows
data.duplicated().sum()

```

```
# deleting all duplicated rows
# data.drop_duplicates()
data.drop_duplicates(inplace=True)
data.duplicated().sum()

Out[29]: 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
```

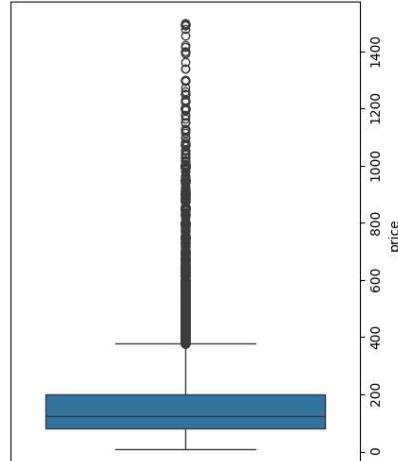
EDA

Task 5: Data Analysis

Univariate Analysis

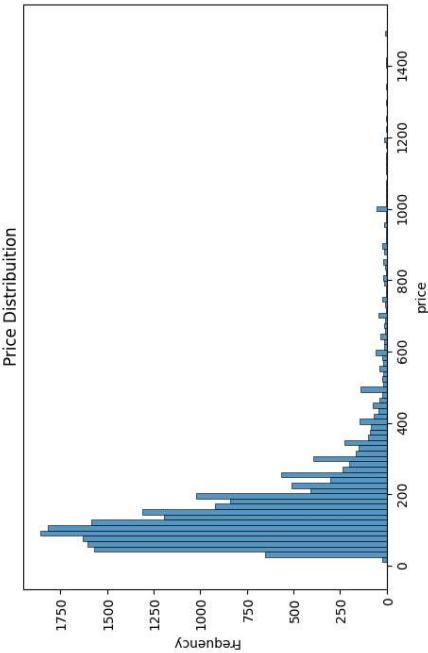
```
In [32]: # identifying outliers in price
df = data[data['price'] < 1000]
sns.boxplot(data=df, x='price')

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x1d3e3a0>
```



```
In [41]: #price distribution
plt.figure(figsize=(8, 5))
```

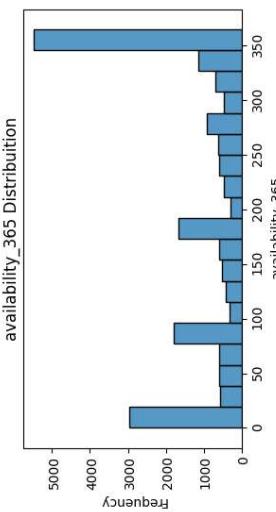
```
sns.histplot(data=df, x='price', bins=100)
plt.title('Price Distribution')
plt.xlabel('Frequency')
plt.show()
```



```
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_1m  object
license        object
rating         object
bedrooms       int64
beds           int64
baths          object
dtype: object
```

```
In [40]: #Price distribution
plt.figure(figsize=(6, 3))
sns.histplot(data=df, x='availability_365')
plt.title('availability_365 Distribution')
plt.xlabel('Frequency')
plt.show()
```



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

```

Out[55]: 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_1m
rating
license
rating
bedrooms
beds
baths
dtype: object

```

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

```

Out[54]: neighbourhood_group      price
Bronx           107.99096
Brooklyn        155.13837
Manhattan       284.168614
Queens          121.681939
Staten Island   118.78069
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      price_per_bed
Bronx           74.713639
Brooklyn        99.789893
Manhattan       138.78057
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(1)

```

```
/var/folders/r6/w418clwnb0v96s8gbt9887cd40909gn77ipykernel.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

```

Out[56]: id      name    host_id  neighbourhood_group  latitude  longitude  room_type  price ... reviews_per_month  calculated_host_listings_count  availability_365  number_of_reviews_1m  rating  license  rating  bedrooms  beds  baths  price_per_bed

```

0	13122280	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	5	1	1	Not specified	550
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	4.67	2	1	1	1440
2	9710000000000000000	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	935
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	License	4.54	1	1	1	1200
4	408966110.0	Condo in New York. ★4.91 - Studio - 1 bed - 1... ...	61391963	Say With Vibie	Manhattan	Murray Hill	40.751120	-73.978600	Entire home/apt	85.0	...	0.24	133.0	335.0	3.0	No	4.91	Studio	1	1	850

5 rows × 23 columns

```

In [55]: df.head()

```

	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	item	license	rating	bedrooms	beds	baths
0	1312280	Rental unit in Brooklyn •★5.0• 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683370	-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	45275370	Rental unit in New York •★4.67• 2 bedrooms...	51501835	Jennifer	Manhattan	Hell's Kitchen	40.766510	-73.988100	Entire home/apt	144.0	...	01/05/23	0.24	1390	364.0	2.0	No License	4.67	2	1	1	
2	971000000000000000	Rental unit in New York •★4.17• 1 bedroom...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.934605	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	408966110	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

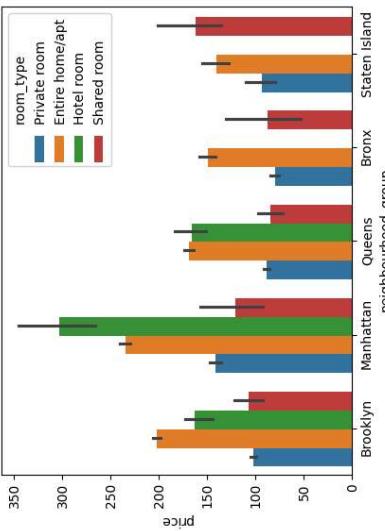
B1 Variable Analysis One variable dependency in another variable

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',
       'review_scores_per_month', 'calculated_host_listings_count',
       'availability_365', 'number_of_reviews_ltm', 'license', 'rating',
       'bedrooms', 'beds', 'baths', 'price_per_bed'],
      dtype='object')
```

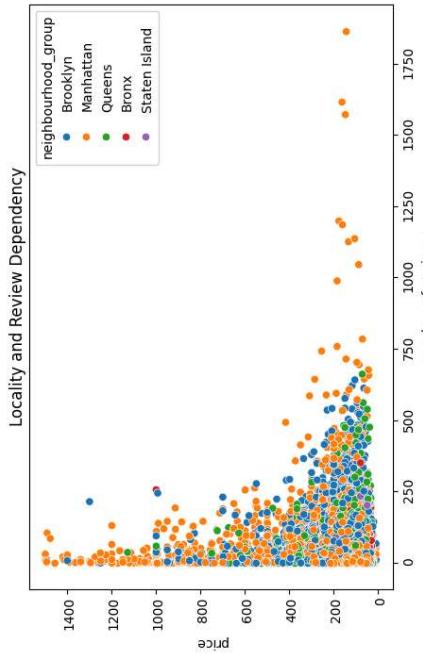
```
In [59]: # price dependency on neighbourhood_group, y=price, hue='room_type'
sns.barplot(data=df, x='neighbourhood_group', y='price', hue='room_type')
```

Out[59]: <Axes: xlabel='neighbourhood_group', ylabel='price'>



```
In [60]: # 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 [61]:



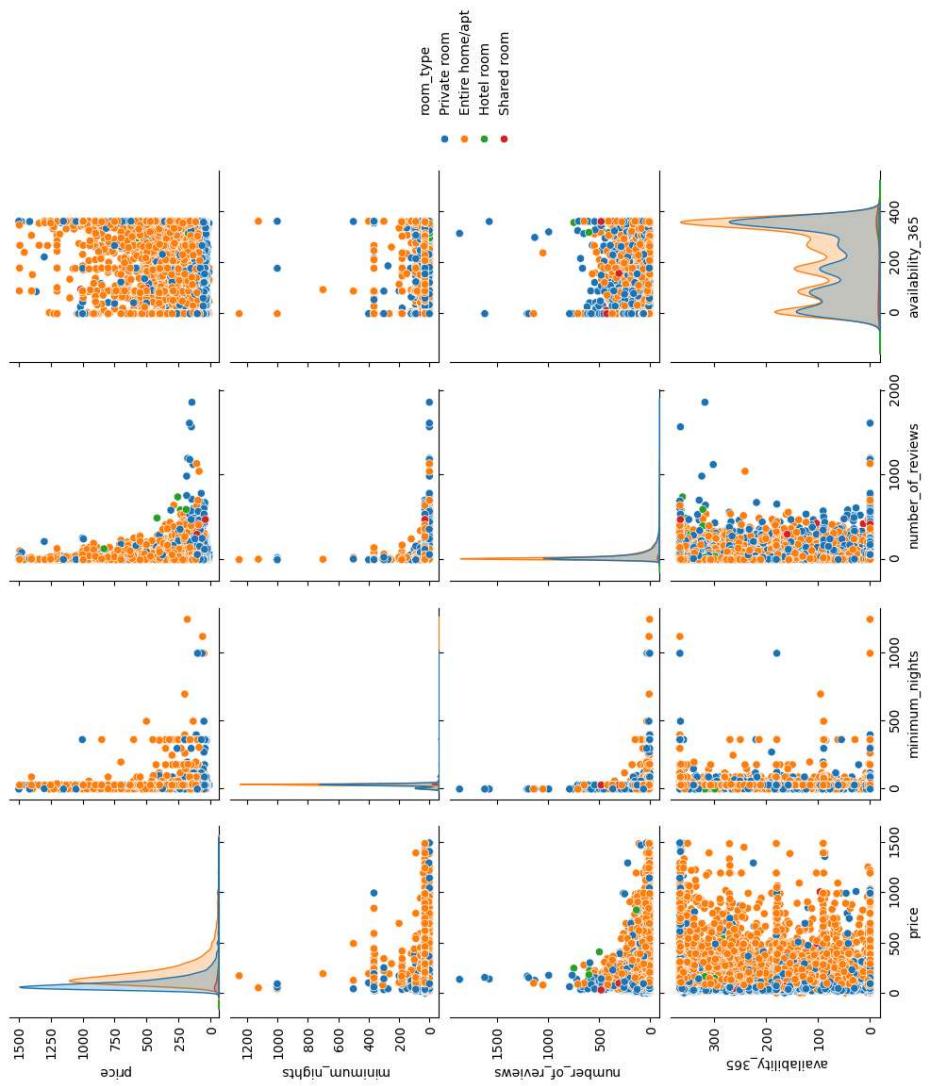
In [68]:

```
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_1tm float64
      license          object
      rating           object
      bedrooms         object
      beds              int64
      baths              int64
      price_per_bed    object
      dtype: object
```

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

Out[70]:

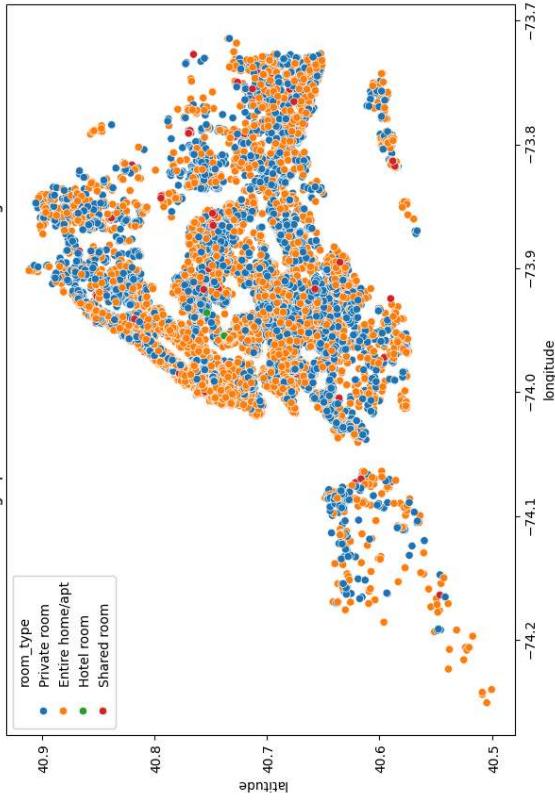
`<seaborn.axisgrid.PairGrid at 0x1495aadd0>`



```
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]: %geographical Distribution of Airbnb Listing
sns.scatterplot(data=df, x='longitude', y='latitude', hue='room_type')
plt.title('Geographical Distribution of Airbnb Listing')
plt.show()
```

Geographical Distribution of Airbnb Listing



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_1m float64
license        object
rating         object
bedrooms       int64
baths          object
price_per_bed  float64
dtype: object
```

In [88]: # heat map - correlation of one variable with others for numerical column
corr = df[['longitude', 'latitude', '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)
```

```
<ipython> Out[88]:
```

