Airbnb Data Analysis

Project using Python

BY
Jatin Patidar

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, Seabornfor cleaning, visualization, and analysis.

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.

Dataset

- The dataset contains 20,765 entries and 22 features, including:
- id: Unique identifier for each listing
- name: Title of the Airbnb listing
- host_name: Name of the host
- neighborhood_group: Group (borough) where the listing is located
- latitude/longitude: Geolocation of listings
- price: Nightly rental price
- room_type: Type of accommodation (e.g., entire home, private room)
- reviews_per_month: Average monthly reviews for the listing
- availability_365: Number of available days in the year

Steps and Workflow

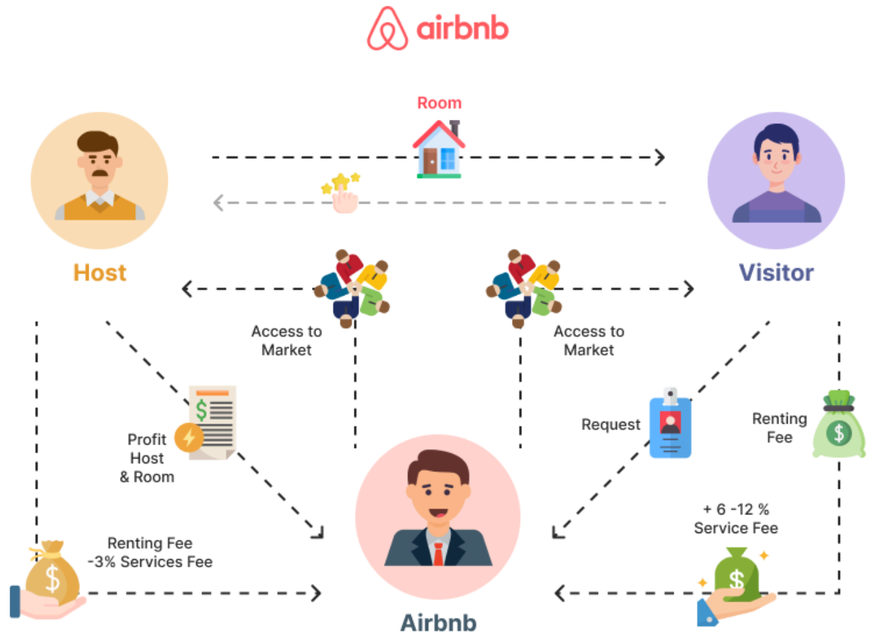
1. Data Cleaning

- Handle missing data: price, neighborhood, and beds columns had null values.
- Fix data types: Converted last_review to a datetime object.
- Remove outliers: Listings with prices > $1,000 were capped to avoid skewed visualizations.

2. EDA (Exploratory Data Analysis)

1. Room type distribution:
    - Visualized the count of each room type using bar plots.
    - Identified Entire home/apt as the most common room type.

2. Neighborhood group insights:
    - Analyzed price variations by boroughs.
    - Manhattan had the highest average prices.



3. Availability trends:
    - Used heatmaps to show correlations among price, availability_365, number_of_reviews, and beds.

- Price distribution:
  - Used histograms to show the distribution of prices.
  - Majority of the listings were priced between $50 - $300.
- Host listings:
  - Analyzed hosts with multiple listings using boxplots to identify key contributors.
- Review behavior:
  - Used pair plots to show relationships between number of reviews, price, and availability.

3. Data Visualization

- Pairplot: To see correlations among price, availability, and number of reviews.
- Heatmap: Showing correlations among numerical features.
- Histograms and Boxplots: To detect outliers in price.
- Bar Charts: Displaying room types and neighborhood group distributions.

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.

Imorting All Dependencies

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

Loading Dataset

```
Data = pd.read_csv(r"C:\Users\a\Downloads\datasets Airbnb row.csv", encoding_errors='ignore')
Data.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_review	reviews_per_month
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	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
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
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

Dataframe information with code

```
Data.size
```

```
456940
```

```
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
```

Descriptive statistics of the Datasets

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

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.000000	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	18.866686
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.532697	73.523401	1.904472	70.921443
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	1.000000	0.010000	1.000000
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.000000	4.000000	0.210000	1.000000
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.000000	14.000000	0.650000	2.000000
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.000000	49.000000	1.800000	5.000000
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	75.490000	713.000000

```
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'],
      dtype='object')
```

Data Cleaning

Null Values and How to Handle Them

```
Data.isnull().sum()
```

```
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
```

Handling Null Values

```
Data.isnull().sum()

# dropping all missing value and rows
Data.dropna(inplace=True)

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

```
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
```

```
Data.shape
```

```
(20736, 22)
```

Dealing With Duplicate Rows

```
Data.duplicated().sum()
```

```
np.int64(12)
```

```
# dealing with duplicete rows
Data.duplicated().sum()

# deleted all duplicetd rows

Data.drop_duplicates(inplace=True)

Data.duplicated().sum()
```

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

Data Types and Changing Data Types

```
Data.dtypes
```

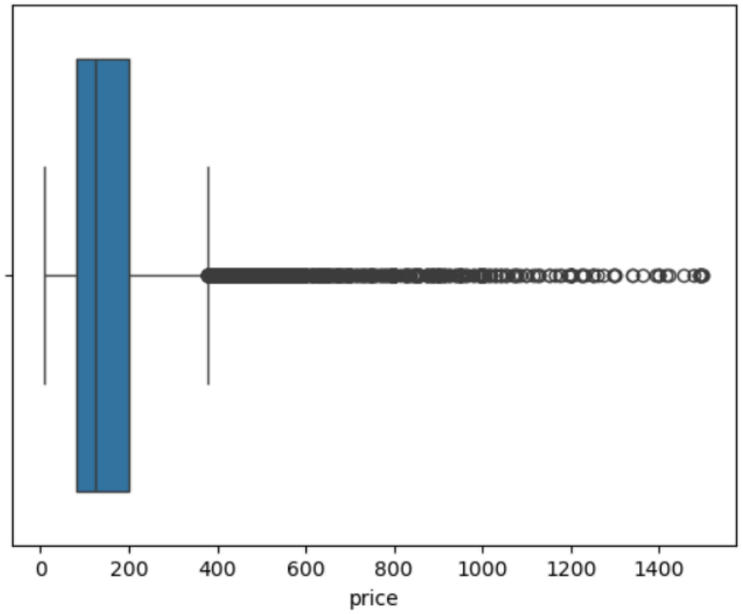
```
id                                 float64
name                                object
host_id                              int64
host_name                           object
neighbourhood_group                 object
neighbourhood                       object
latitude                           float64
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
```

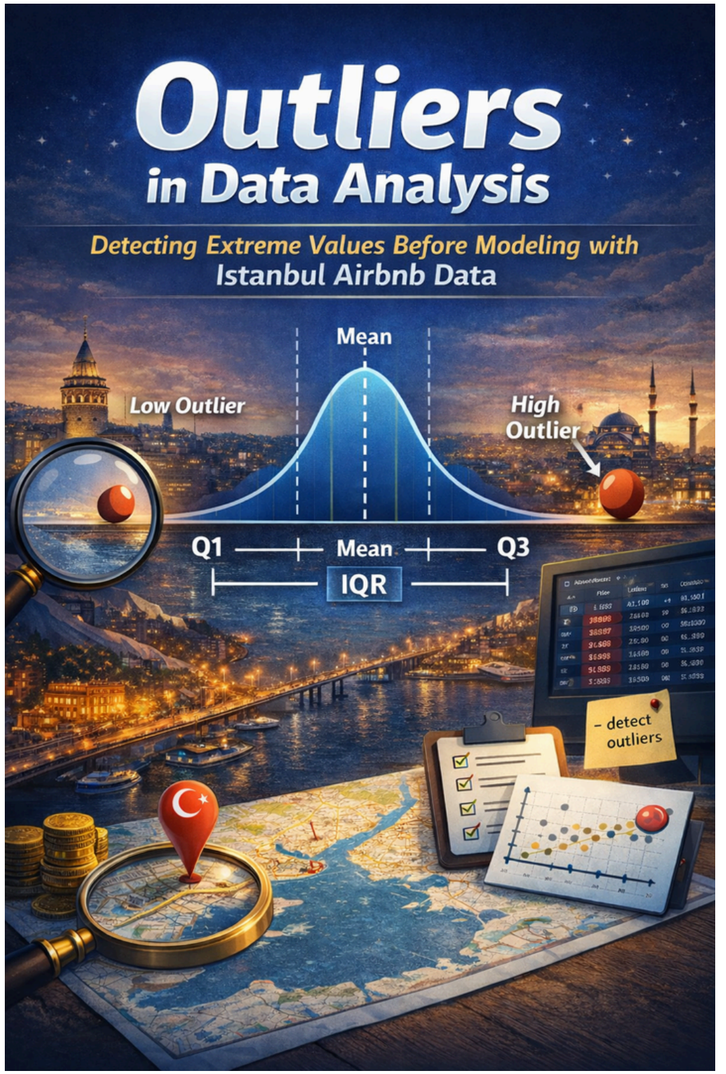
```
# changing typecasting
# changing data type

Data.dtypes

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

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

```
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
Univariate Analysis
Identifying Outliers In Price

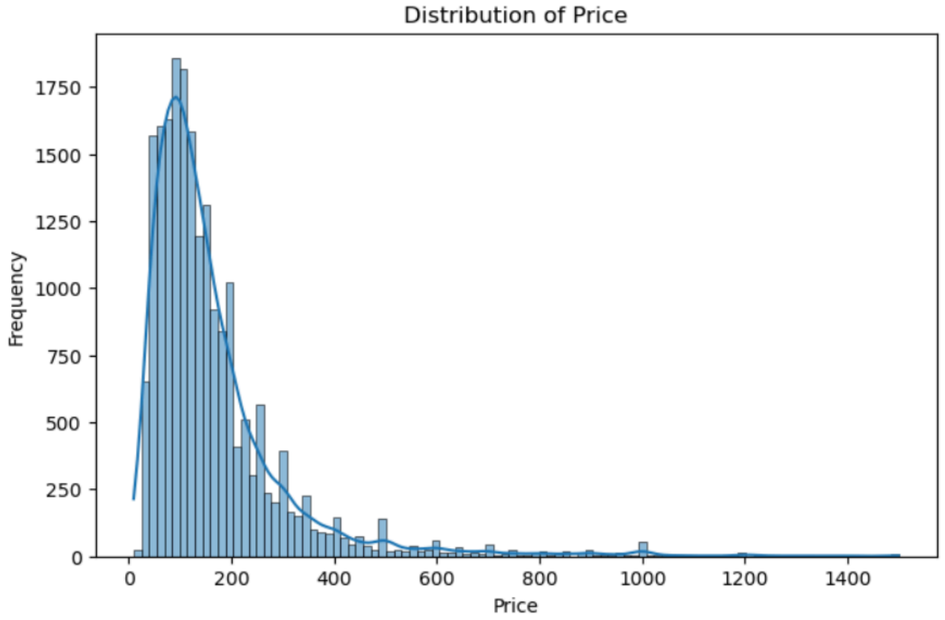
```python
# Calculate IQR
# idenfying outliers in price
Q1 = df['price'].quantile(0.25)
Q3 = df['price'].quantile(0.75)
IQR = Q3 - Q1
# Define outlier boundaries
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers_iqr = df[(df['price'] < lower_bound) | (df['price'] > upper_bound)]
# Create boxplot using the filtered DataFrame
sns.boxplot(data=df, x='price')
plt.show()
```





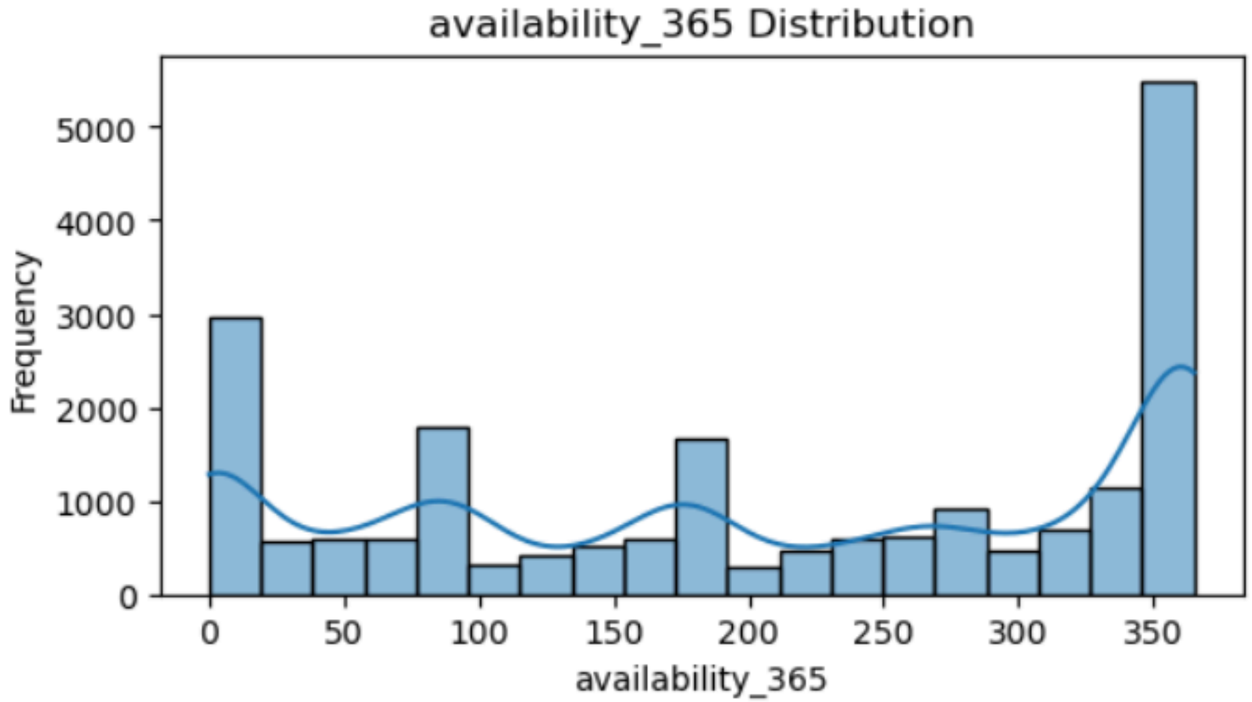
Distribution of Prices Across Listings

```
# price distribution

plt.figure(figsize=(8,5))
sns.histplot(data=df, x='price', bins=100, kde=True)  # bins for granularity, kde for smooth curve
plt.xlabel("Price")
plt.ylabel("Frequency")
plt.title("Distribution of Price")
plt.show()
```



365-Day Availability Distribution

```
# Availability_365 Distribution
plt.figure(figsize=(6,3))
sns.histplot(data=df, x='availability_365', kde=True)  # kde for smoother visualization
plt.xlabel("availability_365")
plt.ylabel("Frequency")
plt.title("availability_365 Distribution")
plt.show()
```



Create Column Price Per Beds

```
# ['price per bed']

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

Average Price Per Bed

```
# average price per bed
df.groupby('neighbourhood_group')['Price per beds'].mean()
```

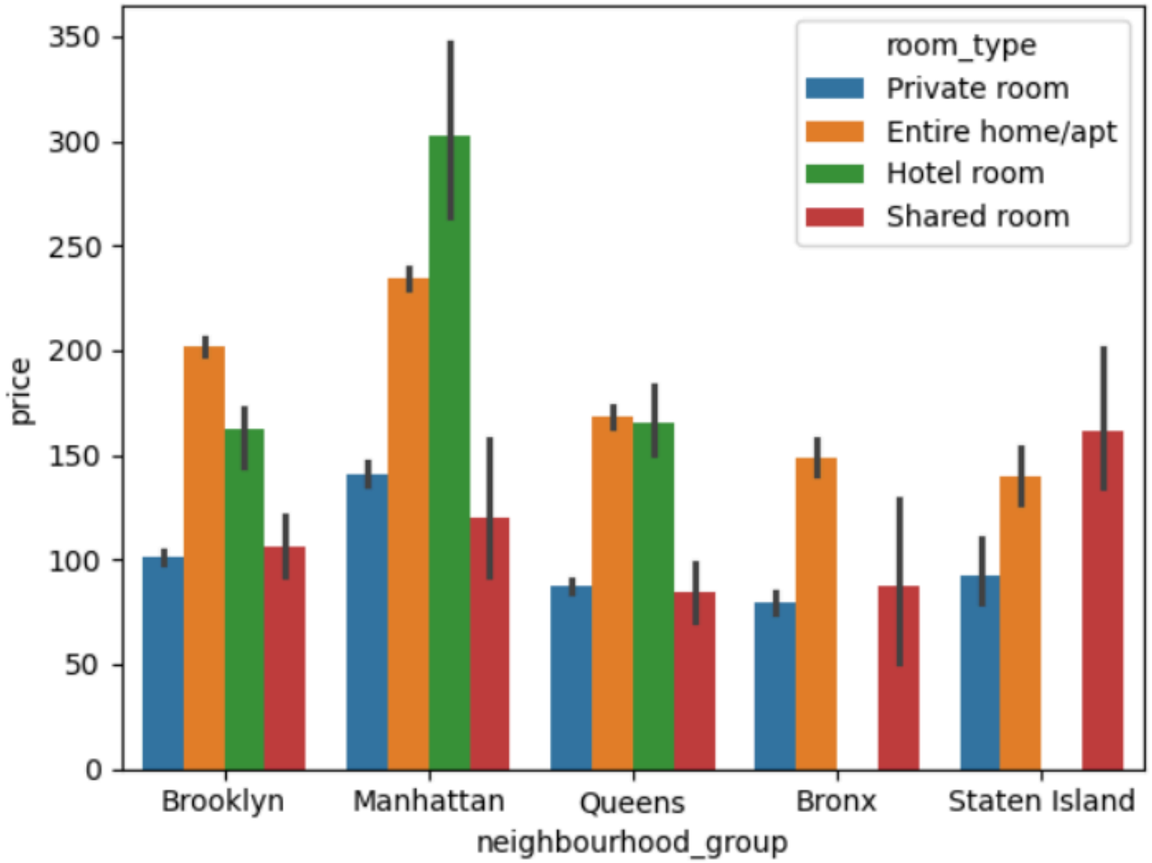
```
neighbourhood_group
Bronx             74.713639
Brooklyn          99.788493
Manhattan        138.708057
Queens            76.336210
Staten Island     67.728101
Name: Price per beds, dtype: float64
```

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

By-Variable Analysis (Grouped Analysis)

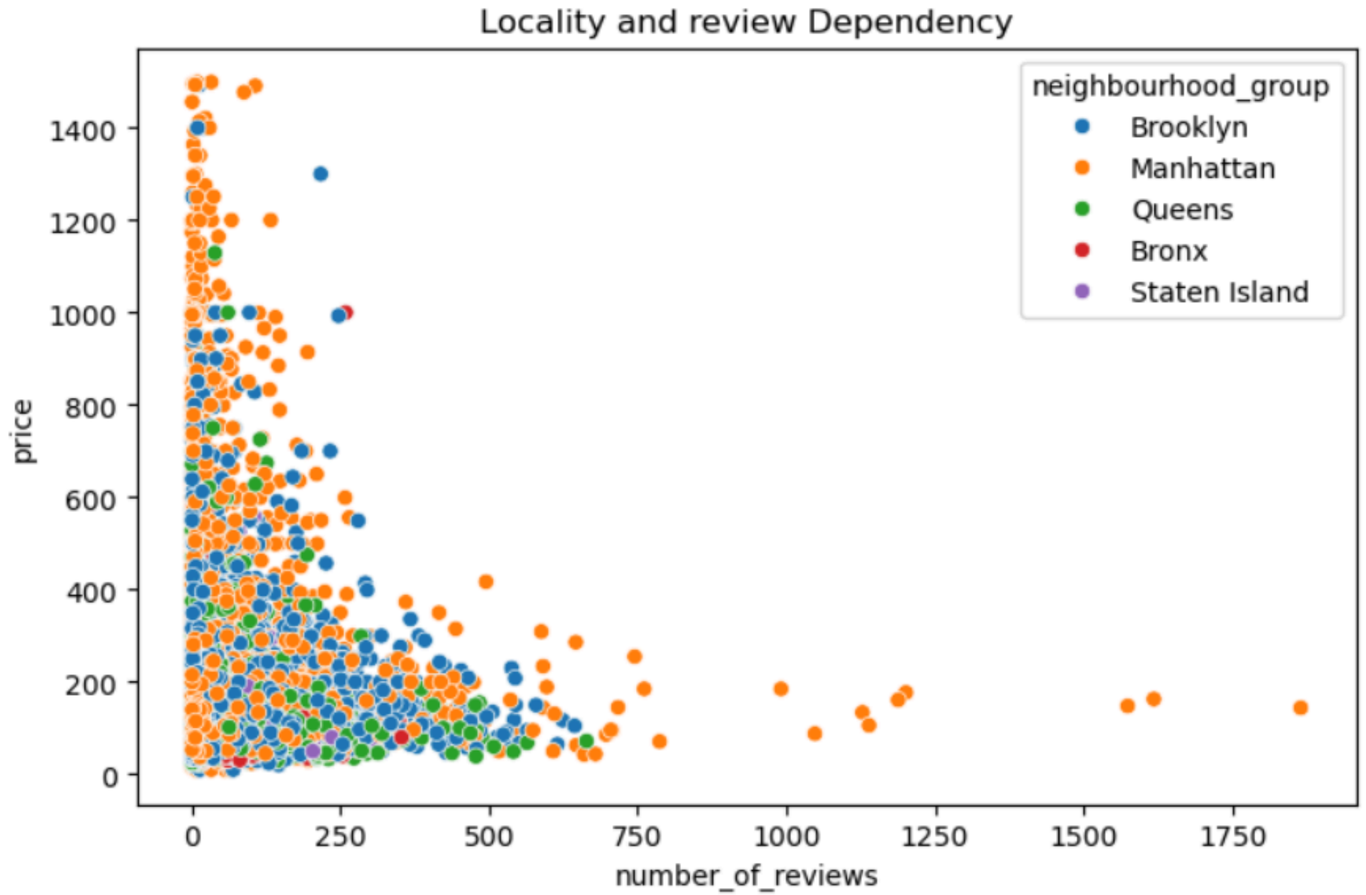
Price Comparison of Room Types Across Neighbourhood Groups

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

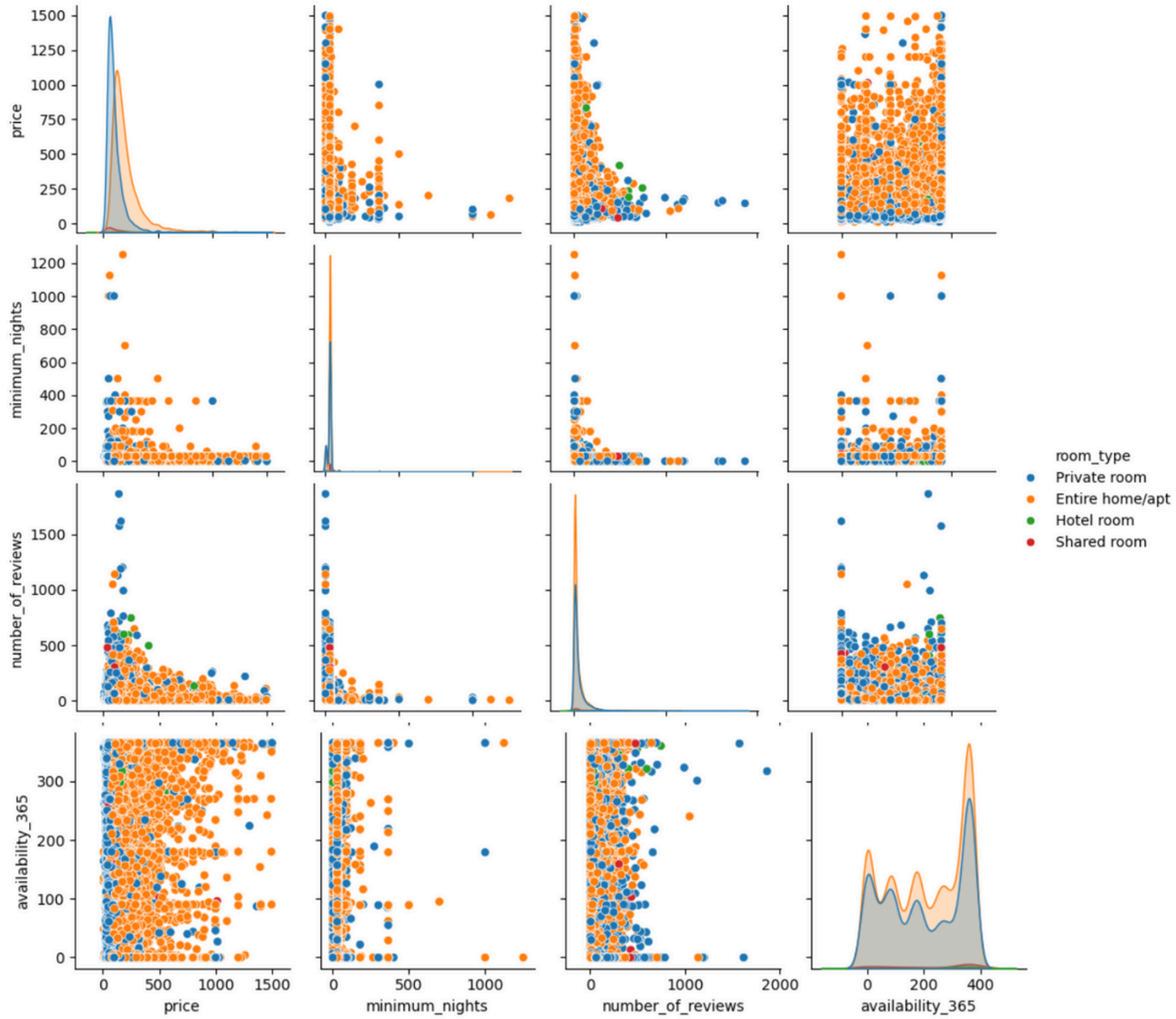


Analysis of Price and Review Counts Across Neighbourhood Groups

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



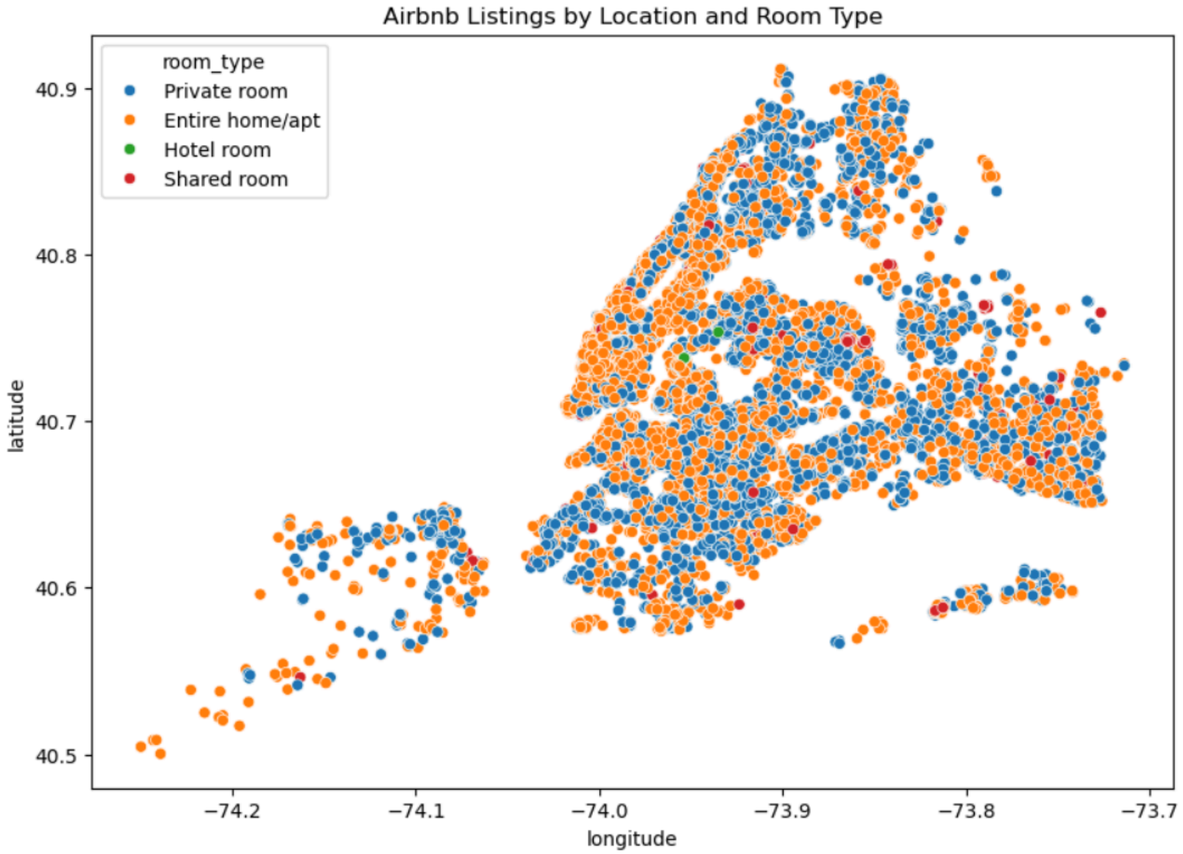
Relationship Between Key Airbnb Numerical Variables




```python
# Pairwise Relationship Analysis of Numerical Features
sns.pairplot(
    data=df,
    vars=['price', 'minimum_nights', 'number_of_reviews', 'availability_365'],
    hue='room_type'
)
plt.show()
```

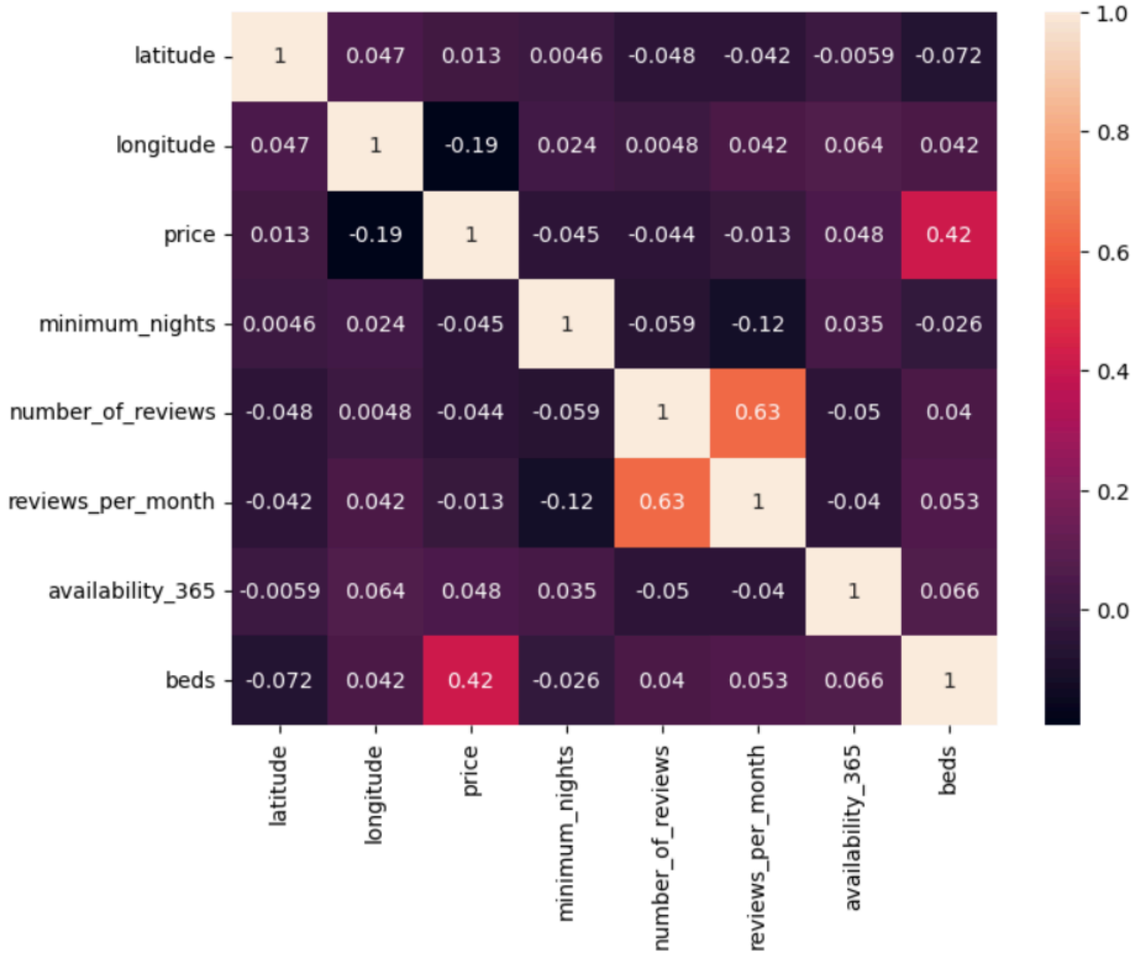
Location_Wise Distribution of Airbnb Room Types

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

)
plt.title("Airbnb Listings by Location and Room Type")   # geographical Distribution
plt.show()
```



Correlation Heatmap Of Airbnb Numerical Variables

```
# heat map - correlation of one variable with others for numerical columns
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)
plt.show()
```



Airbnb Data Analysis – EDA Graph Explanations

1. Box Plot – Outlier Detection

Purpose: Detect extreme values in Airbnb prices using the IQR method.

Insight: Highlights unusually high or low prices that may distort analysis.

Usefulness: Ensures cleaner data for accurate trend and pattern discovery.

2. Histogram + KDE – Price Distribution

Purpose: Show the frequency distribution of listing prices.

Insight: KDE curve smooths the distribution, revealing common price ranges and extreme values.

Usefulness: Helps identify typical pricing bands and anomalies.

3. Histogram + KDE – Availability Distribution

Purpose: Display how listings vary in terms of annual availability.

Insight: KDE curve highlights overall trends in availability patterns.

Usefulness: Shows whether most listings are short-term, seasonal, or year-round.

4. Bar Plot – Price by Neighbourhood & Room Type

Purpose: Compare average prices across neighbourhood groups segmented by room type.

Insight: Reveals how location and accommodation type influence pricing.

Usefulness: Helps understand market segmentation and pricing strategies.

5. Scatter Plot – Reviews vs Price

Purpose: Visualize relationship between number of reviews and listing price.

Insight: Different colors represent neighbourhood groups, showing variation in review patterns.

Usefulness: Identifies whether popular listings (high reviews) are priced differently across locations.

6. Pair Plot – Feature Relationships

Purpose: Explore relationships between numerical features (price, minimum nights, reviews, availability).

Insight: Colored by room type for easy comparison of trends and distributions.

Usefulness: Detects correlations and differences across listing categories.

7. Scatter Plot – Geographical Distribution

Purpose: Map listings using latitude and longitude.

Insight: Colors represent room types, showing spatial concentration of accommodation types.

Usefulness: Identifies hotspots and distribution patterns across the city.

8. Heatmap – Feature Correlation

Purpose: Show correlation between numerical features.

Insight: Highlights strong positive or negative relationships.

Usefulness: Guides feature selection for predictive modeling and deeper analysis.

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.

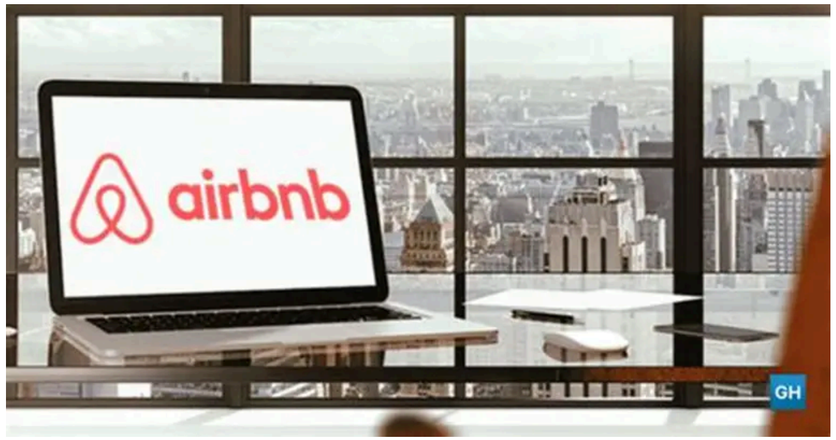
airbnb

Future Work

- Use machine learning to predict prices based on room type and location.
- Perform sentiment analysis on reviews to better understand guest experiences.
- Create an interactive dashboard using Plotly or Tableau for live monitoring.

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.



airbnb

Thank You

jatinpatidar717@gmail.com