TEAM ALPHA

AIRBNB DATA ANALYSIS

Comprehensive Airbnb Data Analysis: Unveiling Trends, Insights, and Opportunities for Enhanced Hosting and Guest Experiences.

Presented for:
Sir Muhammad Rizwan

TABLE OF CONTENTS

About Us

Name	CNIC
Syed Mansoor ul Hassan Bukhari	[redacted]
Mirza Abdul Rehman Baig	[redacted]
Muhammad Asfand Khan	[redacted]

Introduction

Airbnb has changed the way people travel and stay in different places by offering a wide range of short-term rentals. As Airbnb continues to grow, it becomes important for hosts, guests, and the platform itself to understand what makes a listing successful. This report is an Exploratory Data Analysis (EDA) that looks closely at Airbnb listings to find useful insights and trends.

Purpose and Goals:

The main aim of this analysis is to explore the data from Airbnb listings to understand key factors like pricing, availability, guest satisfaction, and overall market trends. Through this analysis, we try to answer important questions like:

- How do prices differ across various neighborhoods and room types?
- Which neighborhoods are the most popular, and why?
- How does the availability of listings change over time, and what affects it?
- What is the connection between guest reviews and the success of a listing?

Scope of the Analysis:

This analysis covers a wide range of Airbnb listings from different cities and neighborhoods, giving us a complete view of the market. The dataset we used includes detailed information about the listings, such as:

- **Location:** The analysis looks at listings from different neighborhoods to compare how they perform.
- **Time Period:** The data covers several years, helping us see trends and patterns over time.
- **Key Features:** We focused on important features like room type, price, service fees, minimum stay requirements, number of reviews, and availability.

How We Analyzed the Data:

We followed a step-by-step process to make sure the data was well-examined and the insights were accurate:

1. Data Cleaning: We started by cleaning the data, which included fixing missing information, correcting data types, and dealing with outliers. This step ensured that the data was accurate and ready for analysis.

2. Univariate Analysis: To understand each feature on its own, we performed univariate analysis. This included looking at the distribution of prices, room types, and other key metrics using graphs and statistics.

3. Bivariate and Multivariate Analysis: We also explored the relationships between different features. For example, we looked at how neighborhood affects pricing, how room type impacts availability, and how guest reviews relate to listing success. We used various charts and plots to visualize these relationships.

4. Geographical Analysis: Since location is crucial for Airbnb listings, we mapped the listings to see the trends in different neighborhoods. This helped us identify which areas are more favorable for hosts and guests.

5. Time Series Analysis: We analyzed how key metrics like prices and reviews change over time. This helped us understand seasonal patterns and identify periods of high and low demand.

Why This Analysis is Important:

The insights from this analysis are valuable not only for individual hosts but also for Airbnb as a platform. By understanding what makes guests happy and what helps hosts succeed, Airbnb can improve its services and stay competitive. The findings can also help potential investors and other stakeholders make informed decisions based on data.

Conclusion:

In short, this report provides a detailed look at the Airbnb market. It combines thorough data analysis with practical suggestions to offer useful insights for everyone involved in the Airbnb ecosystem. Whether you are a host looking to earn more, a guest searching for the best deals, or an Airbnb manager aiming to enhance the platform, this report has something for you.

Data Collection

About the Dataset:

The dataset used for this analysis is sourced from Kaggle's "New York City Airbnb Data Cleaning" project. This dataset provides detailed information about Airbnb listings, reviews, and availability in New York City. The data reflects the activity of homestays and short-term rentals on Airbnb, showcasing how guests and hosts interact on the platform. This information is particularly valuable for understanding the dynamics of the Airbnb marketplace in one of its most popular cities.

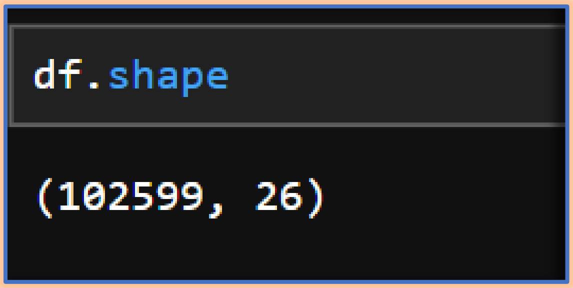
Context:

Airbnb, Inc., founded in 2008, is an American company that operates an online marketplace for lodging, primarily focusing on homestays for vacation rentals and tourism activities. The platform, accessible via both website and mobile app, connects guests with hosts who offer a range of accommodation options. Unlike traditional hotel chains, Airbnb does not own the properties listed on its platform. Instead, it acts as an intermediary, earning revenue by charging a commission on each booking made through the site. The dataset used in this analysis is part of the "Airbnb Inside" initiative, which aims to provide transparency and insights into the listing activity of homestays in various cities, including New York City.

Data Description:

The dataset consists of several key components that capture various aspects of Airbnb listings in New York City:

- **Listings:** This includes detailed information about each Airbnb property, such as its name, host details, neighborhood, latitude, longitude, price, and room type. There are 102,599 observations and 26 features in the dataset after initial cleaning.



```
df.shape
```

```
(102599, 26)
```

Figure 1 - Total Observations

- **Reviews:** This section contains guest reviews linked to each listing, providing insights into guest satisfaction and experiences.

- **Calendar:** The calendar data tracks the availability and pricing of each listing on a daily basis, helping to analyze trends in availability and pricing over time.

Why This Data Matters:

The dataset offers a comprehensive view of the Airbnb market in New York City, one of the most popular and competitive markets for short-term rentals. By analyzing this data, we can gain insights into:

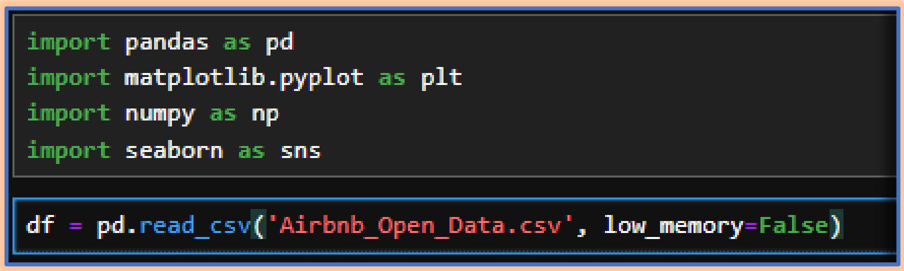
- How different neighborhoods perform in terms of pricing and occupancy.
- The factors that influence guest satisfaction and the likelihood of receiving positive reviews.
- The impact of pricing strategies on a listing's availability and overall success.

Data Limitations:

While the dataset is rich with information, it is important to note that it only covers Airbnb listings in New York City. Therefore, the findings may not be directly applicable to other cities or regions. Additionally, the data represents a snapshot of the Airbnb market at a specific time and may not capture longer-term trends or changes in the market.

Data Importation:

The following code snippets were used to load the dataset into the Python environment:



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

```
df = pd.read_csv('Airbnb_Open_Data.csv', low_memory=False)
```

Figure 2 - Import Libraries

Explanation:

- **Pandas Library:** The pandas library is utilized for data manipulation and analysis. It offers data structures like DataFrame, which makes it easier to handle and analyze large datasets.

- **pd.read_csv() Function:** This function is used to read a CSV file and load it into a DataFrame. The low_memory=False parameter is used to efficiently manage memory usage when dealing with large datasets.

- **Shape Verification:** The shape function is called to confirm the number of rows (observations) and columns (features) in the dataset.

Conclusion:

By understanding the data source, its structure, and how it was imported, we are well-equipped to perform a detailed exploratory data analysis to uncover insights and trends within the Airbnb market in New York City.

Data Cleaning

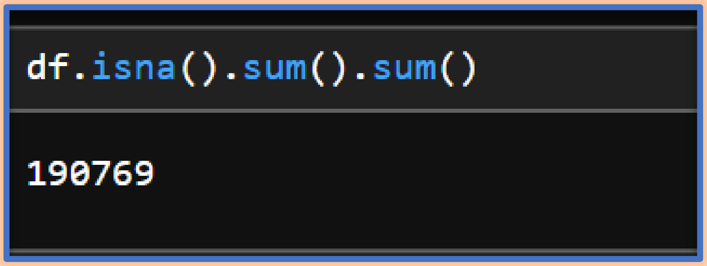
Data cleaning is a critical step in the data analysis process. It involves identifying and handling missing values, detecting and treating outliers, converting data types, and removing duplicates to ensure the dataset is accurate, consistent, and ready for analysis. Below is a detailed breakdown of the data cleaning process for the Airbnb dataset.

Handling Missing Values:

The Airbnb dataset contained several columns with missing values. Handling these missing values was crucial to maintaining the integrity of the analysis. The steps taken were:

1. Identifying Missing Values:

- The dataset was initially inspected for missing values across all columns. A total of 190,769 missing values were identified across different columns, with some columns having a significantly high percentage of missing values.



```
df.isna().sum().sum()
```

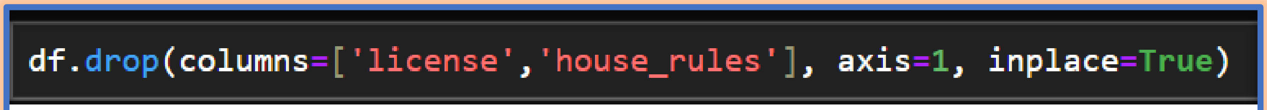
```
190769
```

Figure 3 - Total null values

2. Dropping Columns with Extensive Missing Values:

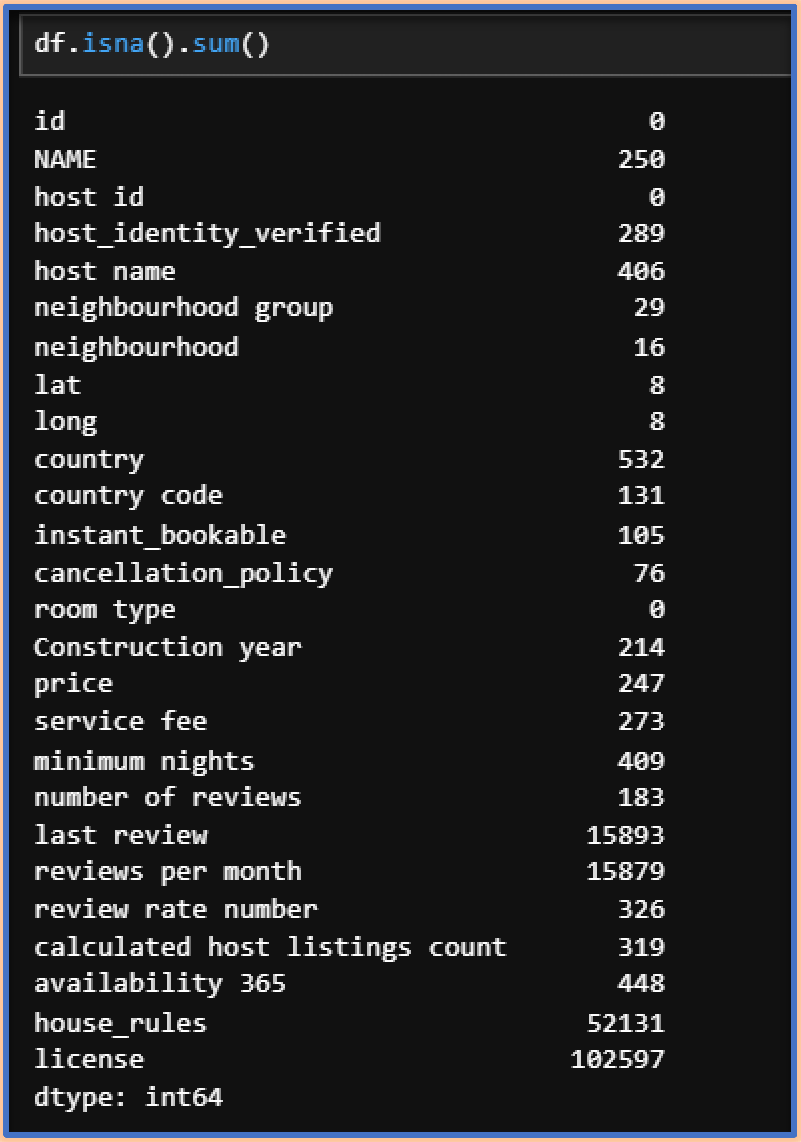
Columns with a high percentage of missing data were removed as they provided little to no value for the analysis. These included:

- **License:** Over 99% of values were missing.
- **House_rules:** More than 50% of values were missing.



```
df.drop(columns=['license','house_rules'], axis=1, inplace=True)
```

Figure 4 - Remove License and House rules, Step performed in figure 5 comes before it to verify missing values



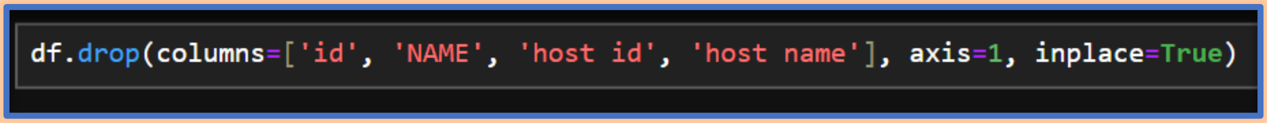
```
df.isna().sum()

id                                     0
NAME                                 250
host id                                0
host_identity_verified               289
host name                            406
neighbourhood group                   29
neighbourhood                         16
lat                                    8
long                                   8
country                              532
country code                         131
instant_bookable                     105
cancellation_policy                   76
room type                              0
Construction year                    214
price                                247
service fee                          273
minimum nights                       409
number of reviews                    183
last review                        15893
reviews per month                  15879
review rate number                   326
calculated host listings count       319
availability 365                     448
house_rules                        52131
license                           102597
dtype: int64
```

Figure 5 - Check For missing values

3. Dropping Columns Irrelevant to the Analysis:

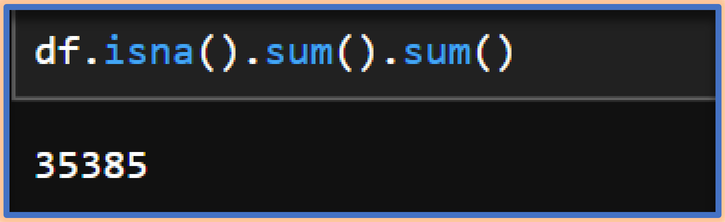
Columns that did not provide significant insights or were redundant were also removed. These included:

- id
- NAME
- host_id
- host_name



```
df.drop(columns=['id', 'NAME', 'host id', 'host name'], axis=1, inplace=True)
```

Figure 6 - Verify total missing values again



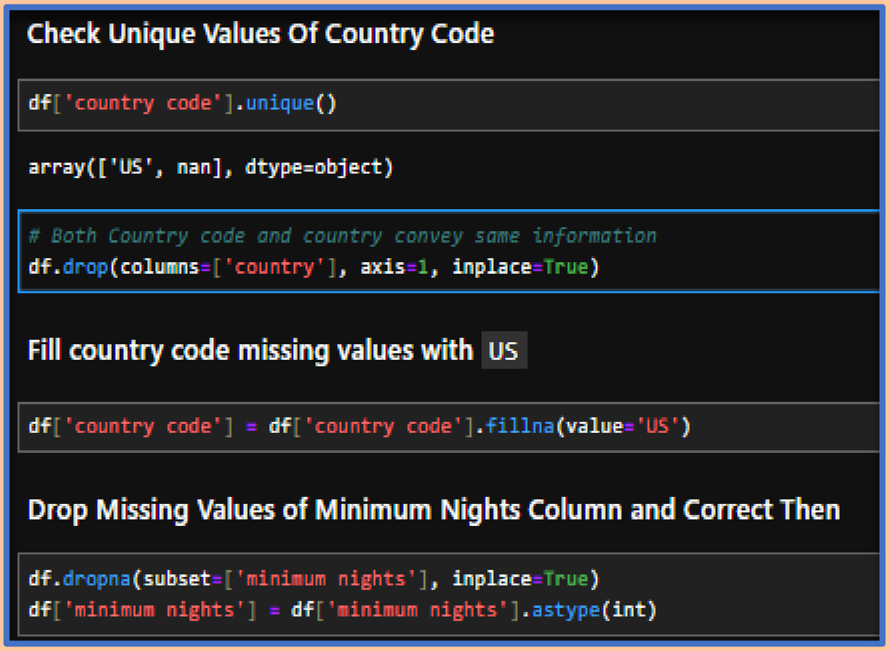
```
df.isna().sum().sum()

35385
```

Figure 7 - Drop Unnecessary Columns

4. Handling Remaining Missing Values:

For columns like `country`, `country_code`, `minimum_nights`, and others, specific strategies were used:

- ✓ **Country Code:** Missing values were filled with 'US', as the majority of listings were in the United States.
- ✓ **Minimum Nights:** Missing values were removed, and the column was converted to an integer type.



Check Unique Values Of Country Code

```
df['country code'].unique()

array(['US', nan], dtype=object)
```

```
# Both Country code and country convey same information
df.drop(columns=['country'], axis=1, inplace=True)
```

Fill country code missing values with US

```
df['country code'] = df['country code'].fillna(value='US')
```

Drop Missing Values of Minimum Nights Column and Correct Then

```
df.dropna(subset=['minimum nights'], inplace=True)
df['minimum nights'] = df['minimum nights'].astype(int)
```

5. Visualizing Missing Values:

A heatmap was used before and after cleaning to visualize the distribution of missing values and ensure that the cleaning process was effective.

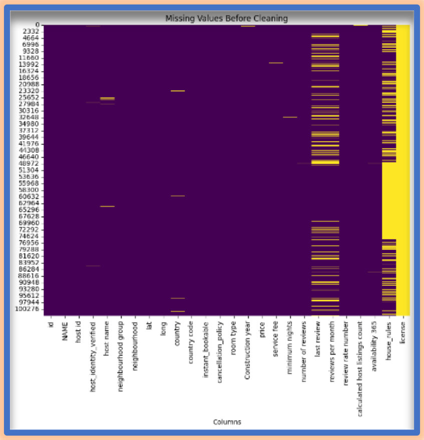


Figure 9 - Missing Value Before Cleaning

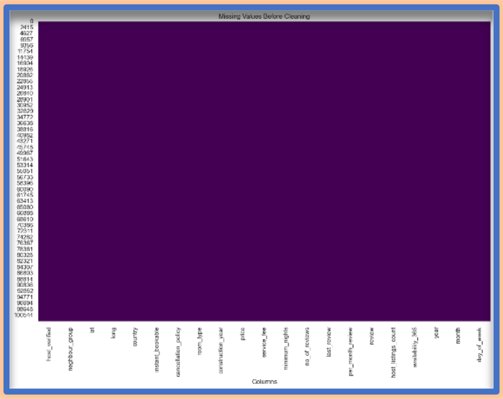


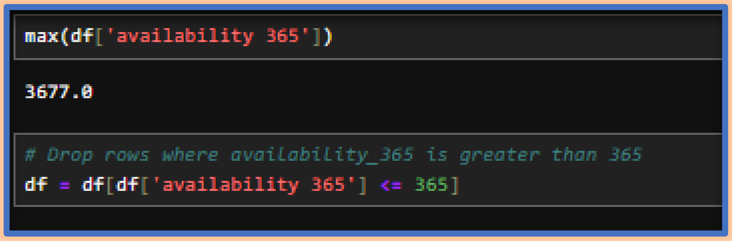
Figure 8 - Missing Values After Cleaning

Outlier Detection and Treatment:

Outliers can significantly impact the results of an analysis. They were detected and treated as follows:

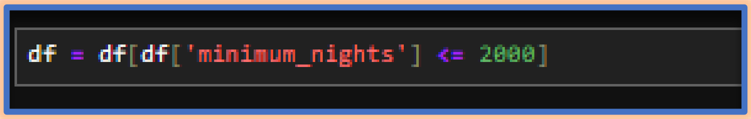
1. Identification of Outliers:

Columns like `minimum_nights` and `availability_365` were examined for extreme values. For instance, values of `minimum_nights` greater than 2000 were considered unrealistic and were removed.



```
max(df['availability 365'])

3677.0

# Drop rows where availability_365 is greater than 365
df = df[df['availability 365'] <= 365]
```

Figure 10 - Correct Availability



```
df = df[df['minimum_nights'] <= 2000]
```

Figure 11 - Correct minimum nights

2. Handling Outliers:

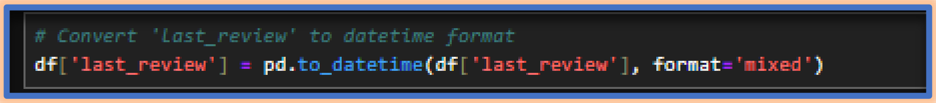
- Outliers in the `availability_365` column (values greater than 365) were removed since they were not realistic.

Data Type Conversion:

Ensuring that each column had the appropriate data type was essential for accurate analysis. The following conversions were made:

1. Conversion of Date Columns:

- ✓ The `last_review` column, which contained date values, was converted from object type to datetime format.

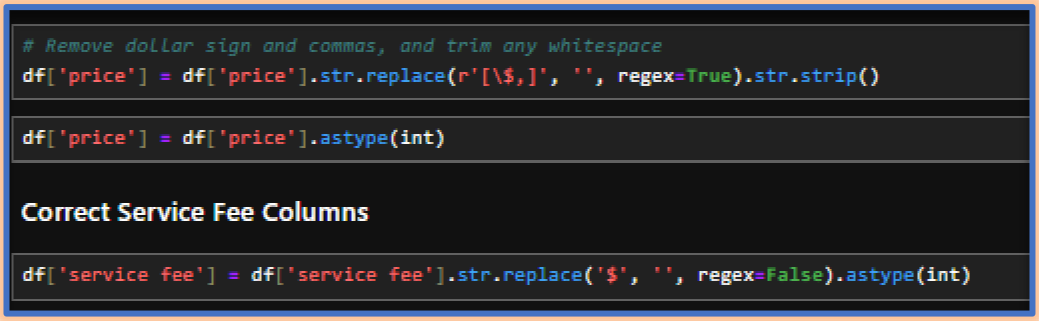


```
# Convert 'last_review' to datetime format
df['last_review'] = pd.to_datetime(df['last_review'], format='mixed')
```

Figure 12 - Convert Object to info

2. Conversion of Numeric Columns:

- ✓ Columns such as `price`, `service_fee`, and `minimum_nights` were converted to integer types after removing any non-numeric characters (e.g., dollar signs and commas).



```
# Remove dollar sign and commas, and trim any whitespace
df['price'] = df['price'].str.replace(r'[\$,]', '', regex=True).str.strip()
```

```
df['price'] = df['price'].astype(int)
```

Correct Service Fee Columns

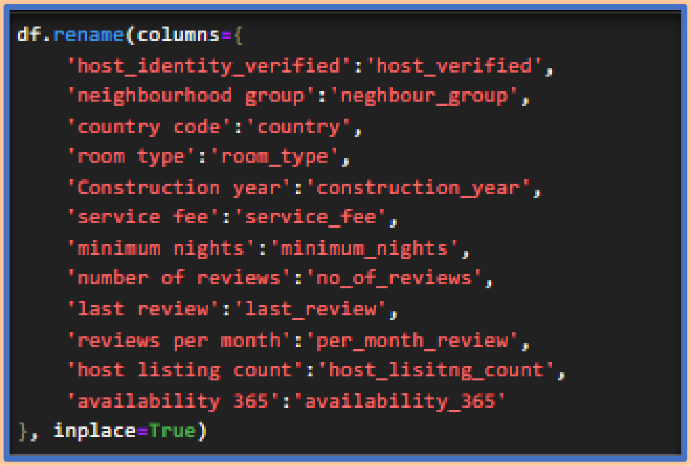
```
df['service fee'] = df['service fee'].str.replace('$', '', regex=False).astype(int)
```

Figure 13 - Correct price and service fee

3. Renaming Columns for Consistency:

Several columns were renamed to make them more descriptive and consistent. For example:

- ✓ `calculated_host_listings_count` was renamed to `host_listings_count`.
- ✓ `availabilty_365` was corrected to `availability_365`.



```
df.rename(columns={
    'host_identity_verified':'host_verified',
    'neighbourhood group':'neghbour_group',
    'country code':'country',
    'room type':'room_type',
    'Construction year':'construction_year',
    'service fee':'service_fee',
    'minimum nights':'minimum_nights',
    'number of reviews':'no_of_reviews',
    'last review':'last_review',
    'reviews per month':'per_month_review',
    'host listing count':'host_lisitng_count',
    'availability 365':'availability_365'
}, inplace=True)
```

Figure 14 - Rename columns

Summary of Changes:

After the data cleaning process, the following key changes were made to the dataset:

- ✓ Removed columns with excessive missing values or irrelevant information.
- ✓ Handled missing values by dropping or imputing data where necessary.
- ✓ Detected and removed outliers to prevent skewed results.
- ✓ Converted data types to ensure that columns were in the correct format.
- ✓ Renamed columns for clarity and consistency.

These steps resulted in a cleaner and more reliable dataset, ready for detailed analysis and visualization.

Data Exploration

In the Data Exploration phase, we dive deep into the dataset to uncover patterns, relationships, and insights that will guide our analysis. This process is crucial as it helps us understand the underlying structure of the data and prepares us for more advanced modeling and analysis tasks.

Overview of the Dataset

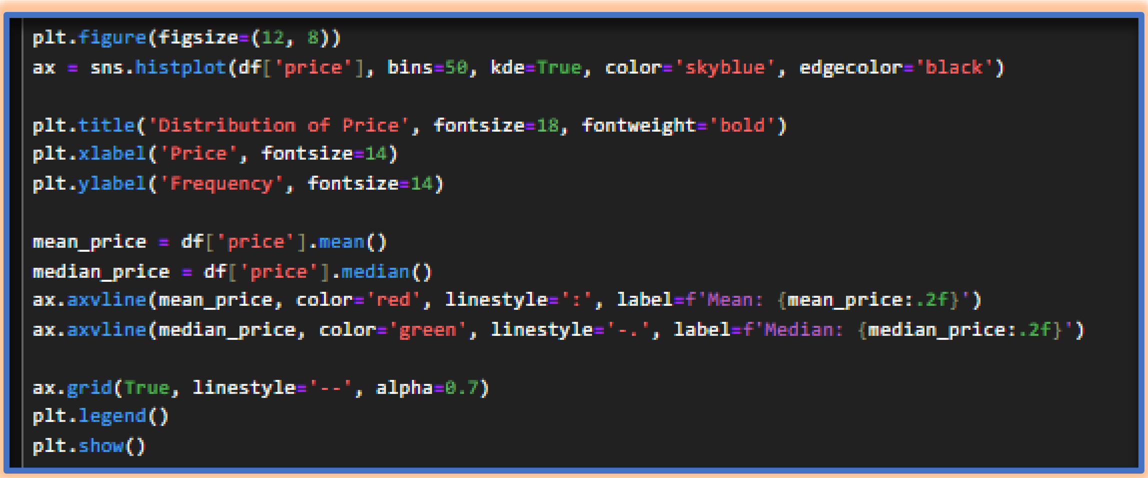
Before we begin exploring specific features, it's important to get a general understanding of the dataset. This involves looking at the basic structure, such as the number of observations (rows) and features (columns), as well as identifying the data types for each feature. In previous section we already discussed about datatype, shapes and missing values.

- **Shape:** The `shape` attribute provides the number of rows and columns in the dataset.
- **Data Types:** Knowing the data types helps us understand what kind of operations can be performed on each feature.
- **Missing Values:** Identifying missing values early on is crucial for deciding how to handle them later in the analysis.

Univariate Analysis

Univariate analysis involves exploring individual features in the dataset to understand their distribution, central tendency, and variability. This step is particularly important for identifying outliers and understanding the overall range of the data.

Price Distribution:



```
plt.figure(figsize=(12, 8))
ax = sns.histplot(df['price'], bins=50, kde=True, color='skyblue', edgecolor='black')

plt.title('Distribution of Price', fontsize=18, fontweight='bold')
plt.xlabel('Price', fontsize=14)
plt.ylabel('Frequency', fontsize=14)

mean_price = df['price'].mean()
median_price = df['price'].median()
ax.axvline(mean_price, color='red', linestyle=':', label=f'Mean: {mean_price:.2f}')
ax.axvline(median_price, color='green', linestyle='-.', label=f'Median: {median_price:.2f}')

ax.grid(True, linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```

Figure 15 - Price distribution

Price is one of the most important features in the Airbnb dataset. Understanding its distribution helps us identify typical pricing strategies and detect any anomalies or outliers.

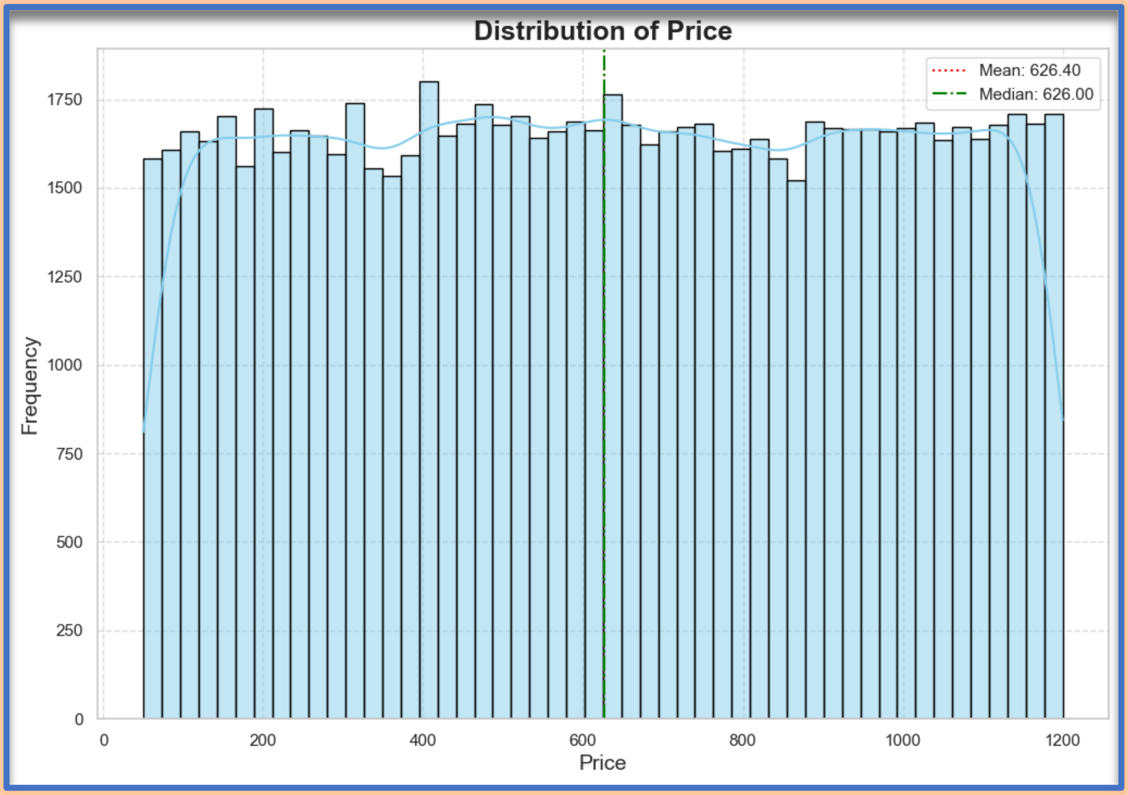
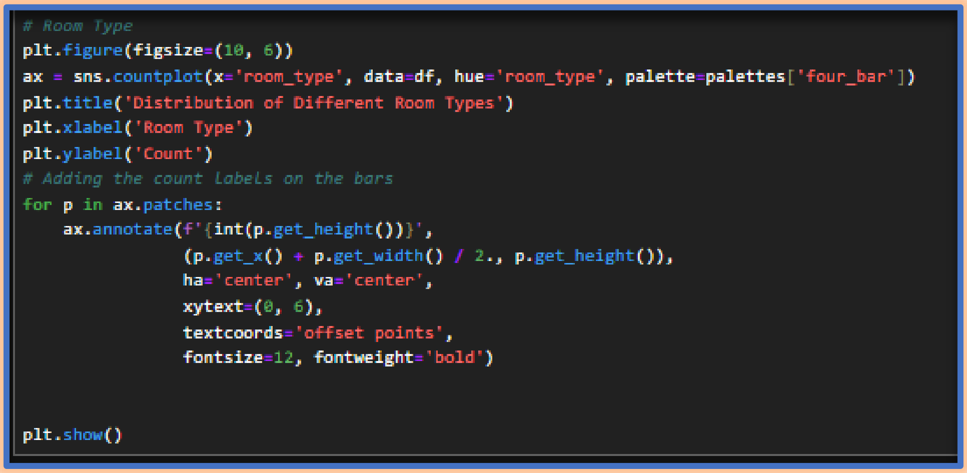


Figure 16 - Graph distribution of Prices

Room Type Distribution:

Room type is another critical feature, as it directly influences the price and availability of the listing. Analyzing the distribution of room types helps us understand the market composition.



```
# Room Type
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='room_type', data=df, hue='room_type', palette=palettes['four_bar'])
plt.title('Distribution of Different Room Types')
plt.xlabel('Room Type')
plt.ylabel('Count')
# Adding the count labels on the bars
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center',
                xytext=(0, 6),
                textcoords='offset points',
                fontsize=12, fontweight='bold')


plt.show()
```

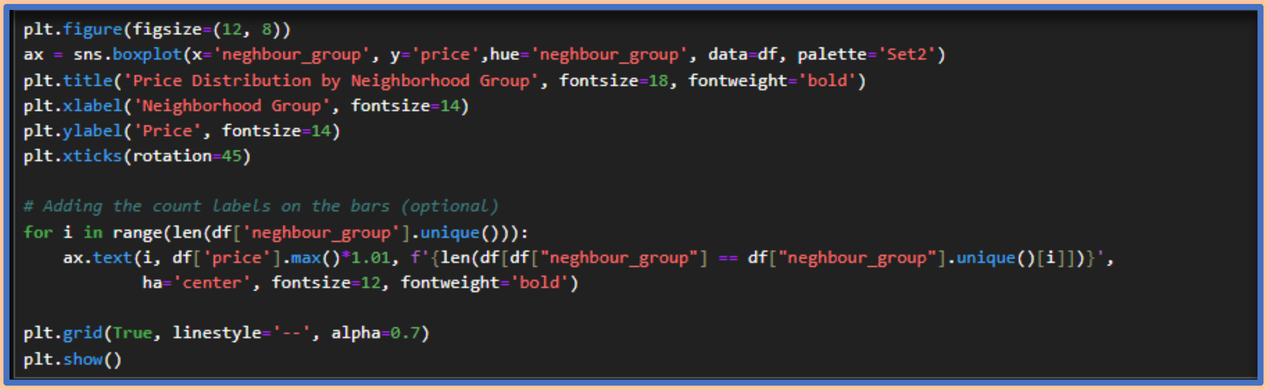
Figure 17 - Room type distribution

Bivariate Analysis

Bivariate analysis helps us explore the relationships between two features. This is where we can start understanding how different factors interact with each other, such as how the neighborhood affects pricing or how the number of reviews relates to the overall rating.

Price vs. Neighborhood:

Neighborhood is a significant factor that influences the price of an Airbnb listing. By examining the relationship between these two features, we can identify which areas are more expensive and which are more affordable.



```python
plt.figure(figsize=(12, 8))
ax = sns.boxplot(x='neghbour_group', y='price',hue='neghbour_group', data=df, palette='Set2')
plt.title('Price Distribution by Neighborhood Group', fontsize=18, fontweight='bold')
plt.xlabel('Neighborhood Group', fontsize=14)
plt.ylabel('Price', fontsize=14)
plt.xticks(rotation=45)

# Adding the count labels on the bars (optional)
for i in range(len(df['neghbour_group'].unique())):
    ax.text(i, df['price'].max()*1.01, f'{len(df[df["neghbour_group"] == df["neghbour_group"].unique()[i]])}',
            ha='center', fontsize=12, fontweight='bold')

plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

Figure 18 - Price By Neighbor Group

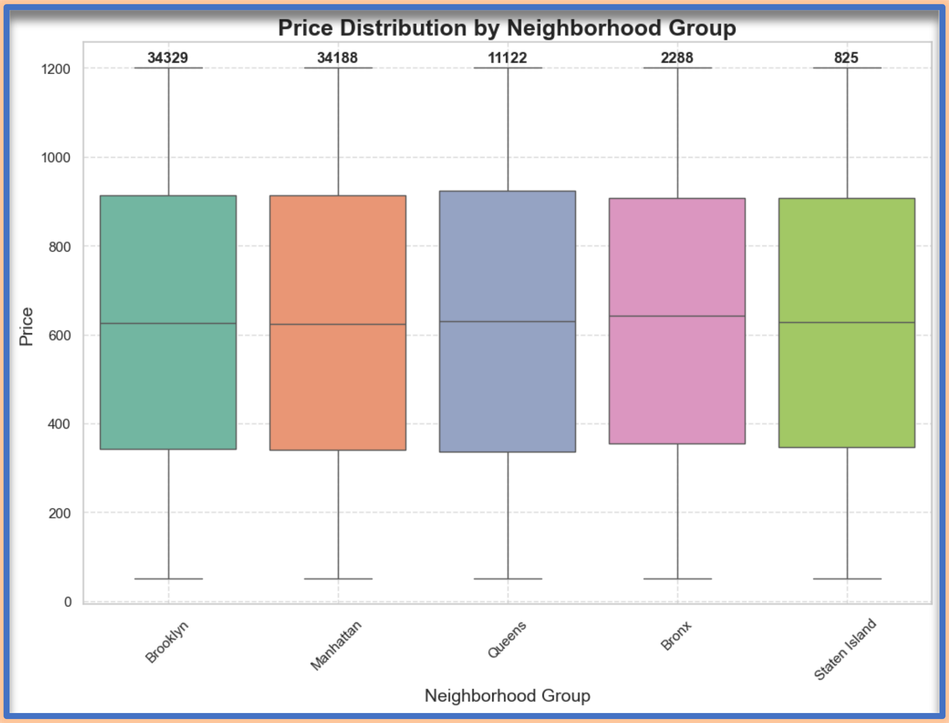


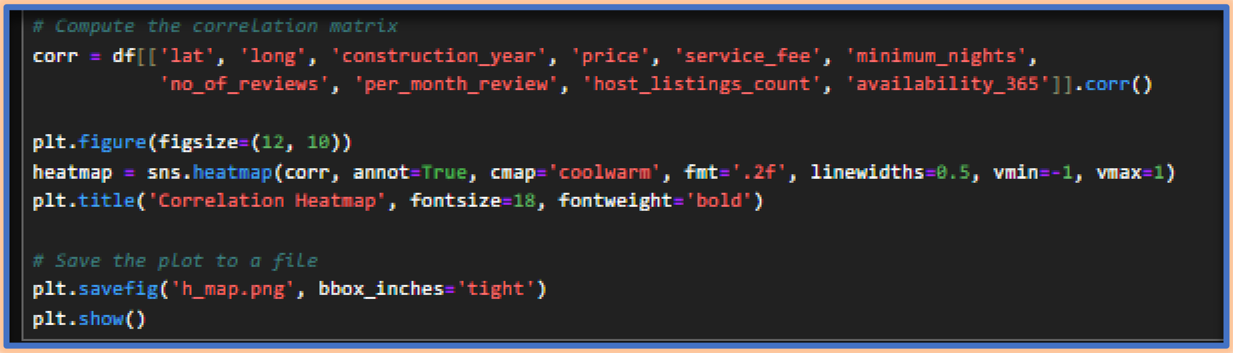
Figure 19 - Price by Neighbors Graph

Multivariate Analysis:

In multivariate analysis, we explore the interactions between more than two features. This type of analysis is useful for understanding more complex relationships and for building predictive models.

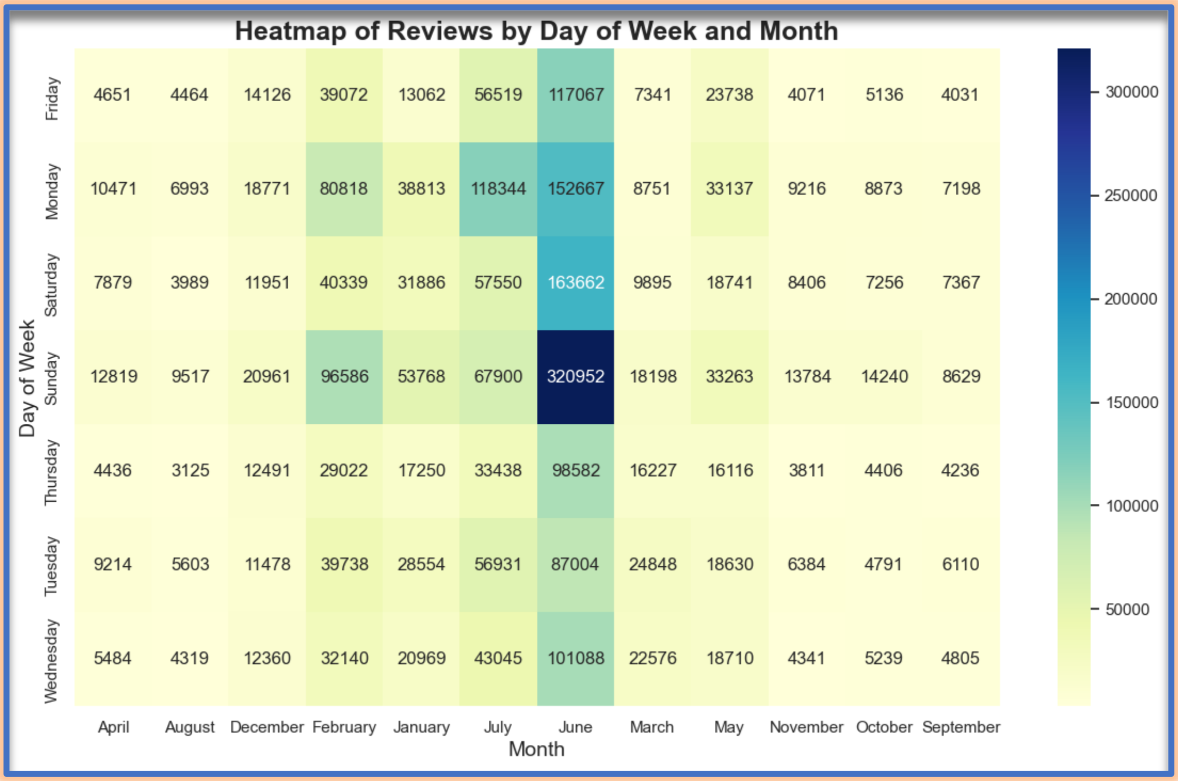
Correlation Heatmap:

A correlation heatmap allows us to visualize the strength of relationships between multiple features at once. This is useful for identifying which features are strongly correlated with each other.



```
# Compute the correlation matrix
corr = df[['lat', 'long', 'construction_year', 'price', 'service_fee', 'minimum_nights',
           'no_of_reviews', 'per_month_review', 'host_listings_count', 'availability_365']].corr()

plt.figure(figsize=(12, 10))
heatmap = sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5, vmin=-1, vmax=1)
plt.title('Correlation Heatmap', fontsize=18, fontweight='bold')

# Save the plot to a file
plt.savefig('h_map.png', bbox_inches='tight')
plt.show()
```

Figure 20 - Heatmap for multivariant



Day of Week	April	August	December	February	January	July	June	March	May	November	October	September
Friday	4651	4464	14126	39072	13062	56519	117067	7341	23738	4071	5136	4031
Monday	10471	6993	18771	80818	38813	118344	152667	8751	33137	9216	8873	7198
Saturday	7879	3989	11951	40339	31886	57550	163662	9895	18741	8406	7256	7367
Sunday	12819	9517	20961	96586	53768	67900	320952	18198	33263	13784	14240	8629
Thursday	4436	3125	12491	29022	17250	33438	98582	16227	16116	3811	4406	4236
Tuesday	9214	5603	11478	39738	28554	56931	87004	24848	18630	6384	4791	6110
Wednesday	5484	4319	12360	32140	20969	43045	101088	22576	18710	4341	5239	4805

Figure 21 - Heatmap

Conclusion:

The Data Exploration phase has provided us with a solid understanding of the Airbnb dataset. By analyzing the distributions of key features, examining relationships between them, and visualizing geographical trends, we have gained valuable insights that will inform further analysis and recommendations. This exploratory analysis sets the stage for more advanced modeling and decision-making processes.

Data Visualization

Introduction:

Data visualization is a crucial step in the Exploratory Data Analysis (EDA) process. It helps in understanding the underlying patterns, trends, and relationships in the dataset. By transforming raw data into graphical representations, we can identify insights that are not immediately apparent in tabular data. In this section, we will discuss various visualizations created to explore the Airbnb dataset, focusing on key aspects such as pricing, availability, neighborhood distribution, and guest reviews.

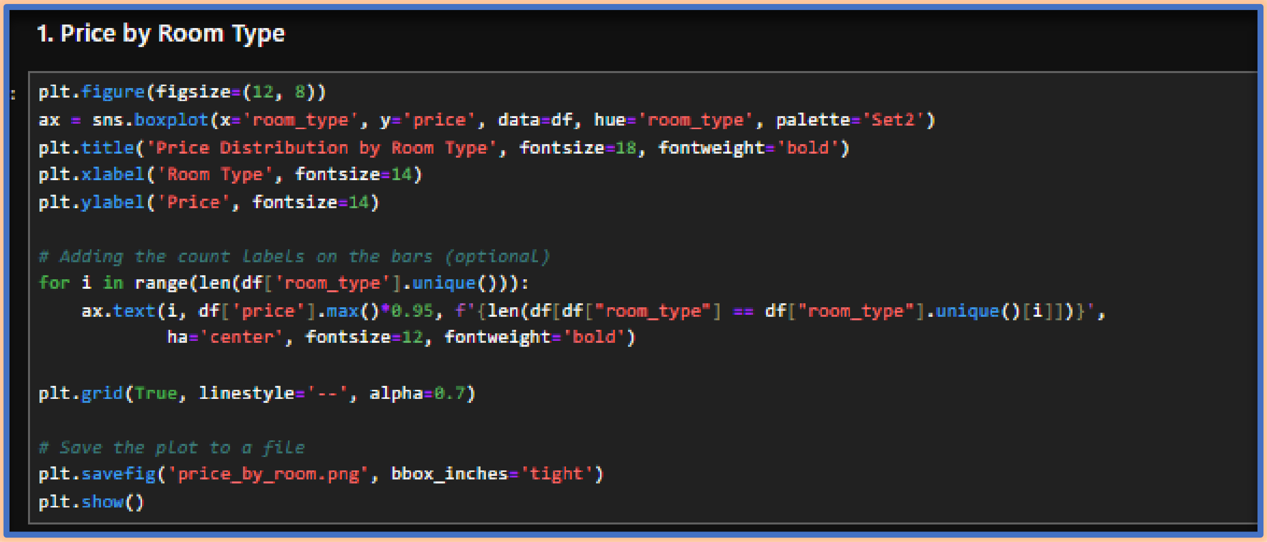
Objectives of Data Visualization:

- **Identify Patterns and Trends:** Visualization helps in identifying patterns and trends that might be difficult to detect in raw data.
- **Understand Relationships:** By plotting different variables against each other, we can explore relationships and correlations between them.
- **Communicate Findings:** Visualizations provide an intuitive way to present findings to stakeholders, making complex data more accessible.

Key Visualizations:

1. Distribution of Prices:

- **Objective:** To understand the distribution of listing prices across different neighborhoods and room types.
- **Visualization Used:** Histogram and Boxplot
- **Insights:** The histogram shows the frequency of different price ranges, helping to identify whether the majority of listings fall within a certain price bracket. The boxplot is used to compare price distributions across different neighborhoods or room types, highlighting any significant differences or outliers.



1. Price by Room Type

```
plt.figure(figsize=(12, 8))
ax = sns.boxplot(x='room_type', y='price', data=df, hue='room_type', palette='Set2')
plt.title('Price Distribution by Room Type', fontsize=18, fontweight='bold')
plt.xlabel('Room Type', fontsize=14)
plt.ylabel('Price', fontsize=14)

# Adding the count labels on the bars (optional)
for i in range(len(df['room_type'].unique())):
    ax.text(i, df['price'].max()*0.95, f'{len(df[df["room_type"] == df["room_type"].unique()[i]])}',
            ha='center', fontsize=12, fontweight='bold')

plt.grid(True, linestyle='--', alpha=0.7)

# Save the plot to a file
plt.savefig('price_by_room.png', bbox_inches='tight')
plt.show()
```

Figure 23 - Prices by Room Type

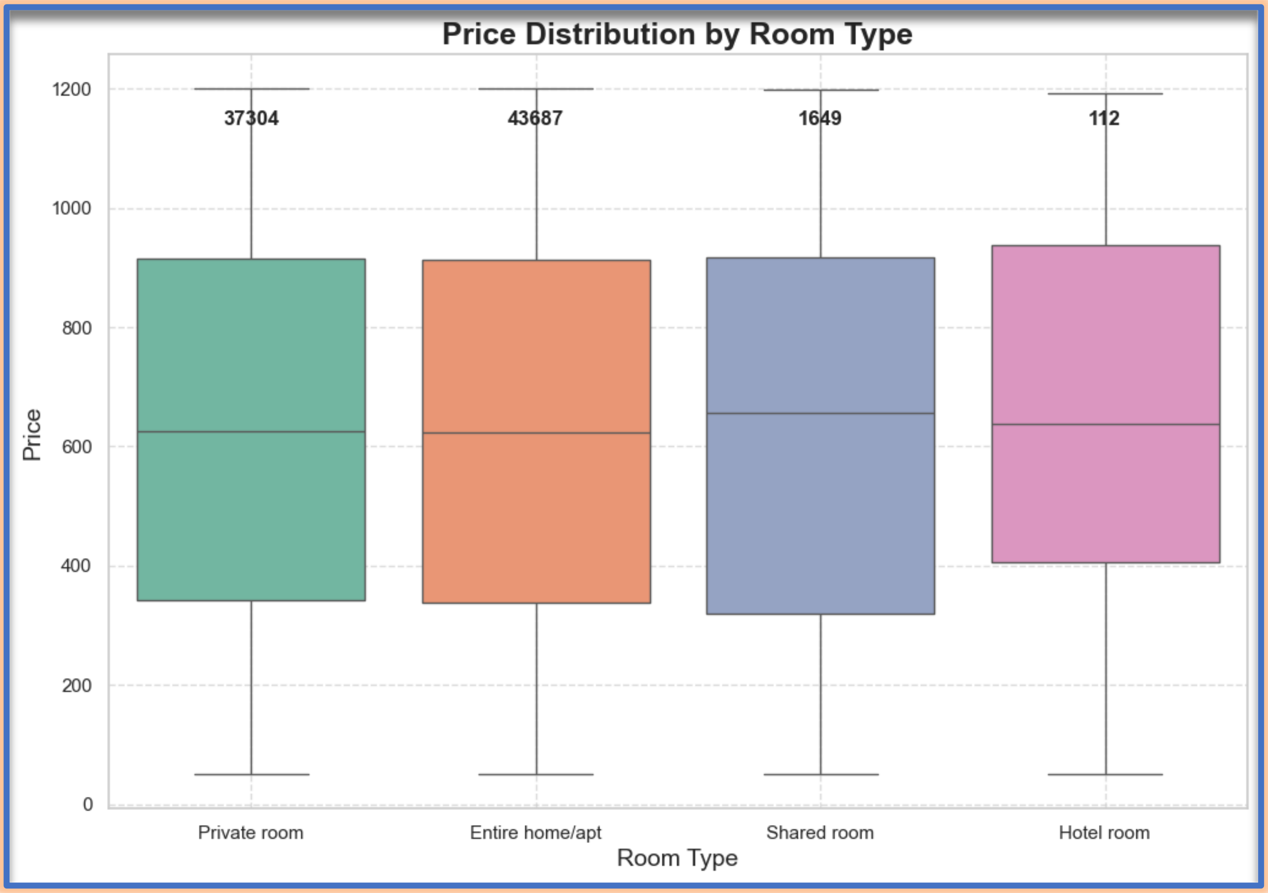
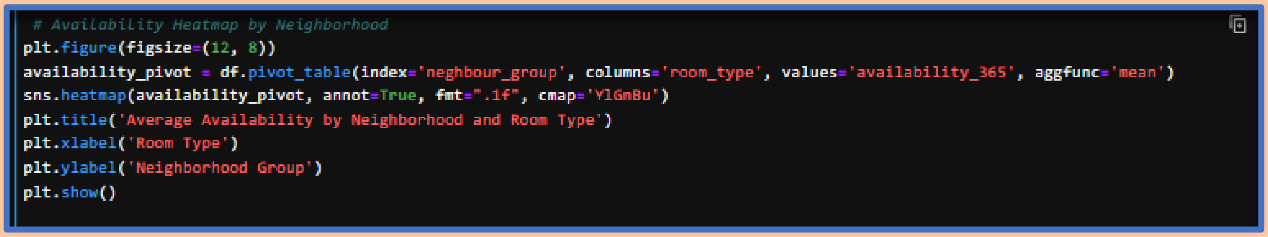


Figure 22 - Prices by Room Type Graph

2. Availability Across Neighborhoods:

- **Objective:** To examine how availability varies across different neighborhoods in New York City.
- **Visualization Used:** Heatmap
- **Insights:** The heatmap provides a visual representation of the availability of listings across different neighborhoods, indicating areas with high or low availability. This can help hosts and Airbnb management understand market saturation and demand in different areas.



```
 # Availability Heatmap by Neighborhood
plt.figure(figsize=(12, 8))
availability_pivot = df.pivot_table(index='neghbour_group', columns='room_type', values='availability_365', aggfunc='mean')
sns.heatmap(availability_pivot, annot=True, fmt=".1f", cmap='YlGnBu')
plt.title('Average Availability by Neighborhood and Room Type')
plt.xlabel('Room Type')
plt.ylabel('Neighborhood Group')
plt.show()
```

Figure 24 - Availability heatmap

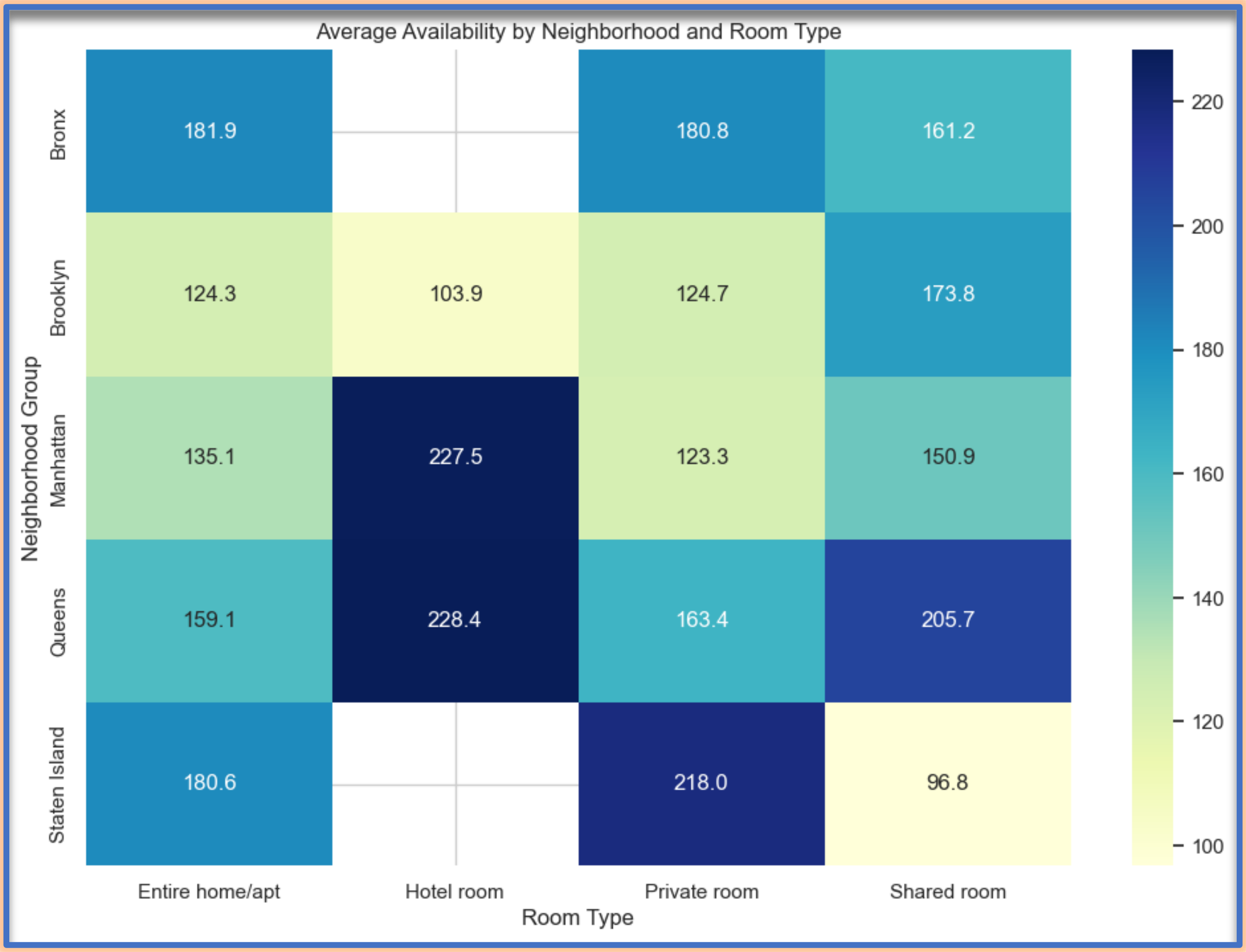
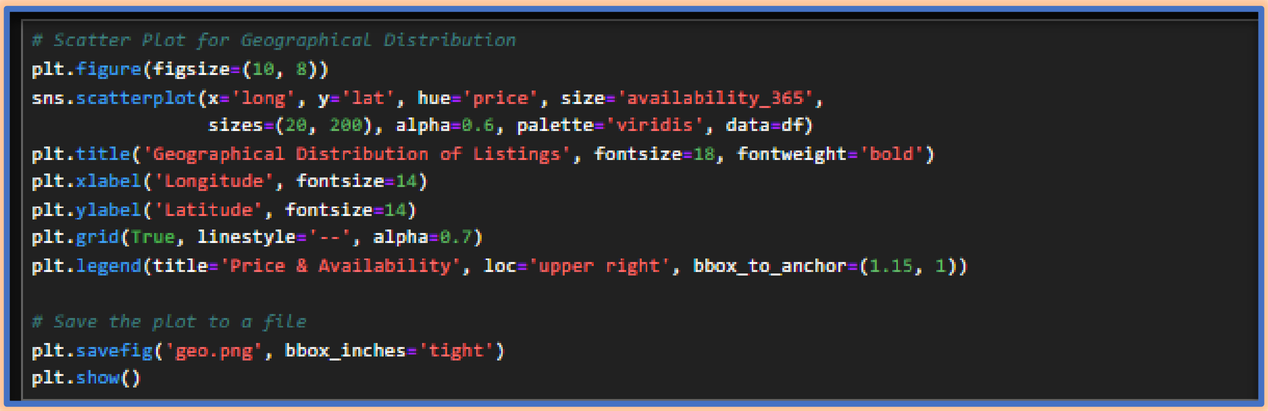


Figure 25 - Availability HeatMap

3. Geographical Distribution of Listings:

- **Objective:** To visualize the geographical distribution of Airbnb listings across New York City.
- **Visualization Used:** Scatter Plot on Map
- **Insights:** This visualization helps in understanding where the majority of listings are concentrated, which neighborhoods are popular, and how the distribution of listings aligns with pricing and availability trends.



```
# Scatter Plot for Geographical Distribution
plt.figure(figsize=(10, 8))
sns.scatterplot(x='long', y='lat', hue='price', size='availability_365',
                sizes=(20, 200), alpha=0.6, palette='viridis', data=df)
plt.title('Geographical Distribution of Listings', fontsize=18, fontweight='bold')
plt.xlabel('Longitude', fontsize=14)
plt.ylabel('Latitude', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.legend(title='Price & Availability', loc='upper right', bbox_to_anchor=(1.15, 1))

# Save the plot to a file
plt.savefig('geo.png', bbox_inches='tight')
plt.show()
```

Figure 26 - Geographical Distribution

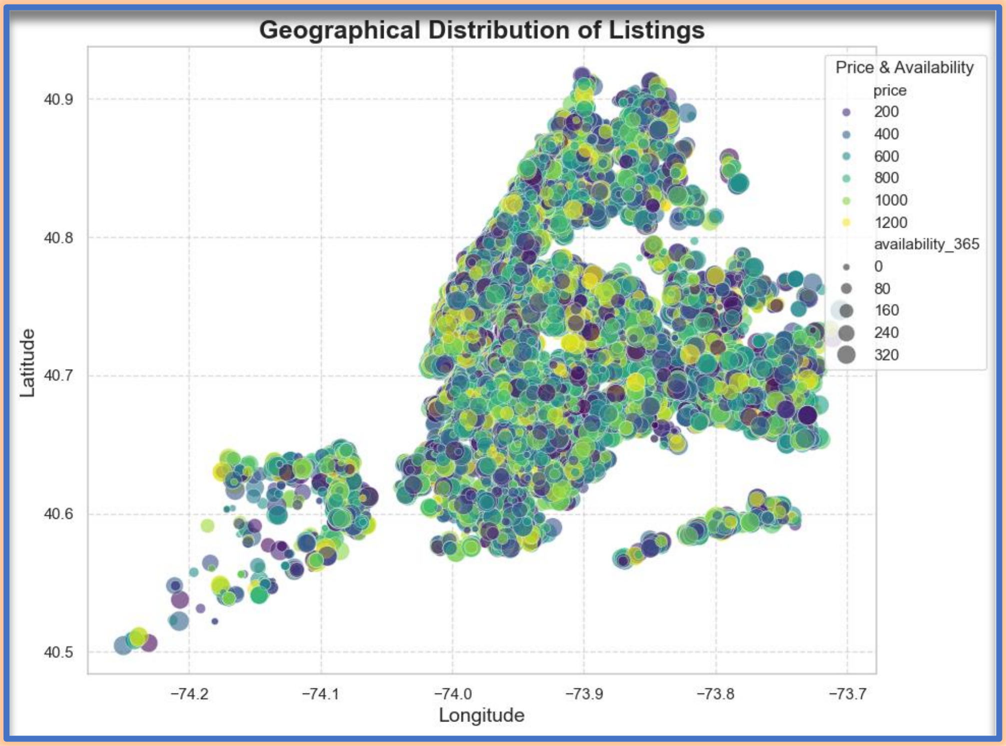
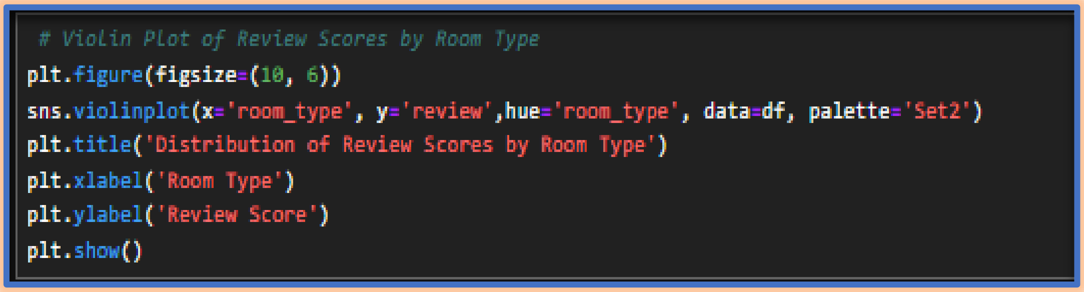


Figure 27 - Map for geographical distribution

4. Review Scores Distribution:

- **Objective:** To analyze the distribution of review scores across different room types and neighborhoods.
- **Visualization Used:** Violin Plot
- **Insights:** The violin plot combines elements of a boxplot and a density plot, showing the distribution of review scores across different categories. This helps in identifying whether certain room types or neighborhoods consistently receive higher or lower ratings.



```
 # Violin Plot of Review Scores by Room Type
plt.figure(figsize=(10, 6))
sns.violinplot(x='room_type', y='review',hue='room_type', data=df, palette='Set2')
plt.title('Distribution of Review Scores by Room Type')
plt.xlabel('Room Type')
plt.ylabel('Review Score')
plt.show()
```

Figure 28 - R Score

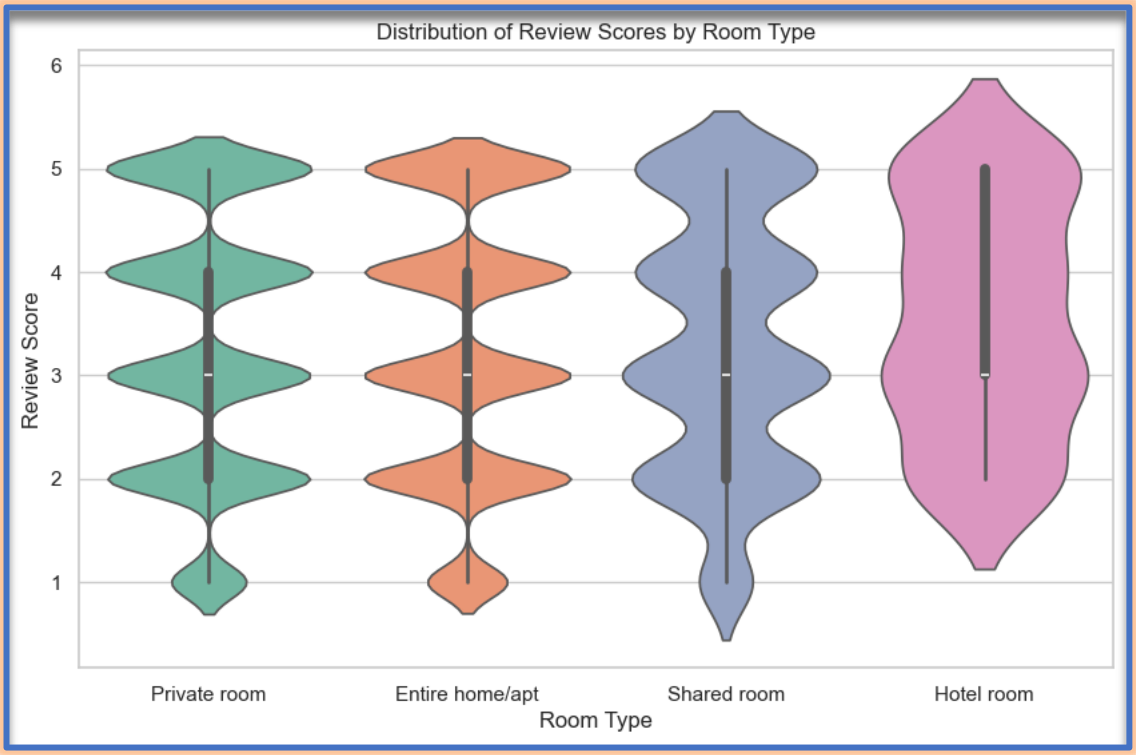


Figure 29 - Graph for review score

Conclusion:

Data visualization is an essential part of EDA as it allows us to see patterns, trends, and outliers that might not be evident in raw data. The visualizations presented in this section provide insights into various aspects of the Airbnb market in New York City, such as pricing, availability, and guest satisfaction. These insights can be used by hosts, guests, and Airbnb management to make informed decisions that enhance the overall experience on the platform.