AirBnb

Author - Sumanasri Valipireddy

Contents

Introduction

This report presents a detailed exploratory data analysis (EDA) of Airbnb listings in New York City using the AB_NYC_2019 dataset. The analysis focuses on identifying pricing patterns, the most and least expensive neighborhoods, and customer preferences. Key insights and actionable recommendations are provided to help Airbnb refine its pricing strategies and improve overall business performance.

Data Collection

The dataset for this exploratory analysis was obtained from Kaggle: "New York City Airbnb Open Data" (AB_NYC_2019).

```
# Load the dataset
airbnb_df = pd.read_csv('AB_NYC_2019.csv')

# Look at the dataset structure
airbnb_df.tail()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	0	NaN	
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	0	NaN	
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	0	NaN	
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	0	NaN	

```
# Dimensions of the dataset
airbnb_df.shape
```

```
(48895, 16)
```

```
# Information of the dataset
airbnb_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

```python
# Descriptive Statistics
airbnb_df.describe()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.000000	48895.000000
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.143982	112.781327
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.952519	131.622289
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.000000	0.000000
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.000000	45.000000
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.000000	227.000000
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

Data Cleaning

Duplicate Removal: The dataset was examined for duplicate rows, but none were found, ensuring data consistency.

Missing Data Handling: Columns such as name, host_name, last_review, and reviews_per_month had missing values. These were addressed by filling in default values. Specifically, name was filled with 'No Name', host_name with 'Unknown', last_review with '1970-01-01', and reviews_per_month with zero. This ensured no key fields remained incomplete.

Variable Identification: Numerical and categorical variables were identified for further analysis. The dataset had 10 numerical fields, including price and availability_365, and 6 categorical fields, including neighbourhood_group and room_type.

Checking missing Values

```python
# Check for duplicate rows in the datasets  s
duplicates = airbnb_df.duplicated()
print(f"Number of duplicate rows: {duplicates.sum()}")
```

```
Number of duplicate rows: 0
```

```python
# Check for missing values before imputation
missing_values_before = airbnb_df.isnull().sum()
print("Missing values before imputation:")
print(missing_values_before)
```

```
Missing values before imputation:
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

Filling missing Values

```
airbnb_df['name'].fillna('No Name', inplace=True)
airbnb_df['host_name'].fillna('Unknown', inplace=True)
```

```
airbnb_df['last_review'].fillna('1970-01-01', inplace=True)
airbnb_df['reviews_per_month'].fillna(0, inplace=True)
```

```
# Verify if missing values are handled
missing_values_after = airbnb_df.isnull().sum()
print("Missing values after imputation:")
print(missing_values_after)
```

```
Missing values after imputation:
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
dtype: int64
```

Identifying Numerical and Categorial Variables

```
# Identifying Numerical Variables
quantitative_vars = [col for col in airbnb_df.columns if airbnb_df[col].dtype in ['int64', 'float64']]
print("Numerical Variables:")
print(quantitative_vars)

# Count of Numerical Variables
num_quantitative_vars = len(quantitative_vars)
print("Count of Numerical Variables:")
print(num_quantitative_vars)
```

```
Numerical Variables:
['id', 'host_id', 'latitude', 'longitude', 'price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month', 'calculated_host_listings_count', 'availability_365']
Count of Numerical Variables:
10
```

```
# Identifying Categorical Variables
categorical_vars = [col for col in airbnb_df.columns if airbnb_df[col].dtype in ['object']]
print("Categorical Variables:")
print(categorical_vars)

# Count of Numerical Variables
num_categorical_vars = len(categorical_vars)
print("Count of Categorical Variables:")
print(num_categorical_vars)
```

```
Categorical Variables:
['name', 'host_name', 'neighbourhood_group', 'neighbourhood', 'room_type', 'last_review']
Count of Categorical Variables:
6
```

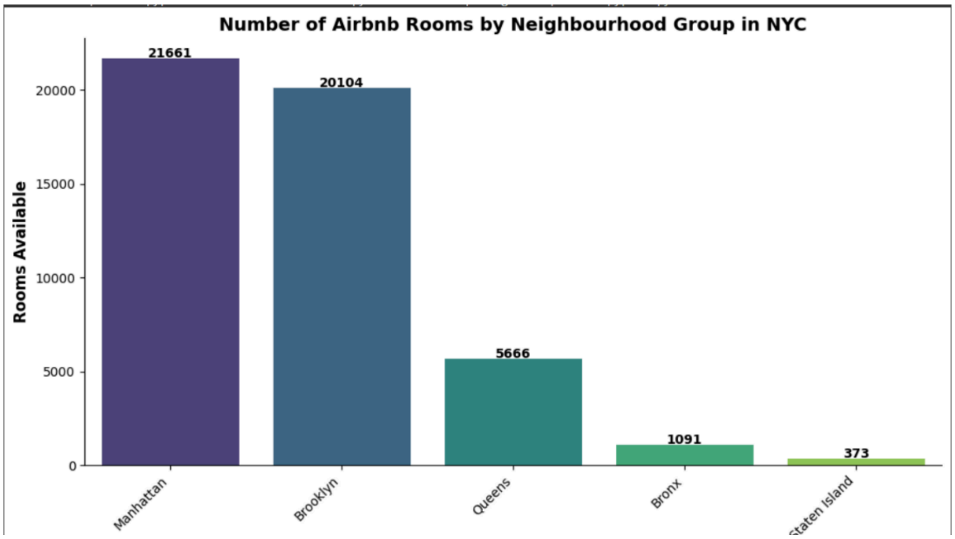
Descriptive Analysis

The main objective of conducting a descriptive analysis on the dataset is to summarize and explore the behaviour of the variables involved in it. The most important techniques for a descriptive analysis involve Frequency Distribution, Measures of Central Tendency and Measures of Dispersion; all of which we are going to explore in this project.

Visualizations:

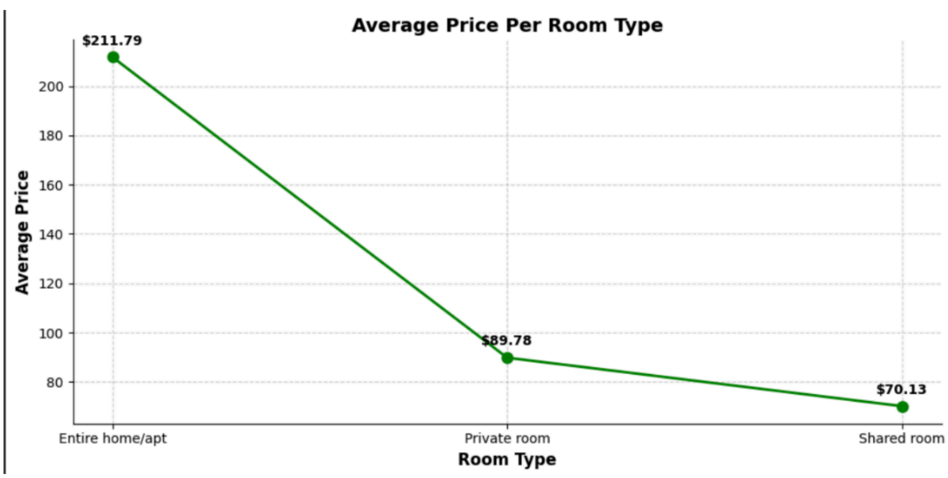
Number of Airbnb by Neighbourhood Groups in NYC

Manhattan -> 21661 (44.3%)
Brooklyn -> 20104 (41.1%)
Queens -> 5666 (11.6%)



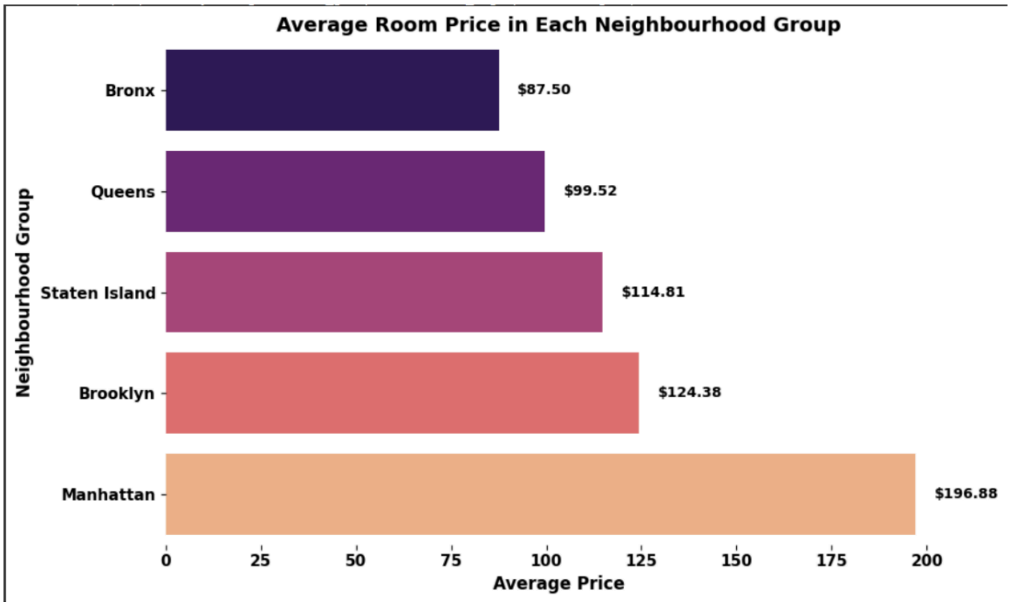
Average Price Per Room Type

- Entire home/apt -> 25409 (51.9%)
- Private Room -> 22326 (45.7%)
- Shared Room -> 1160 (2.4%



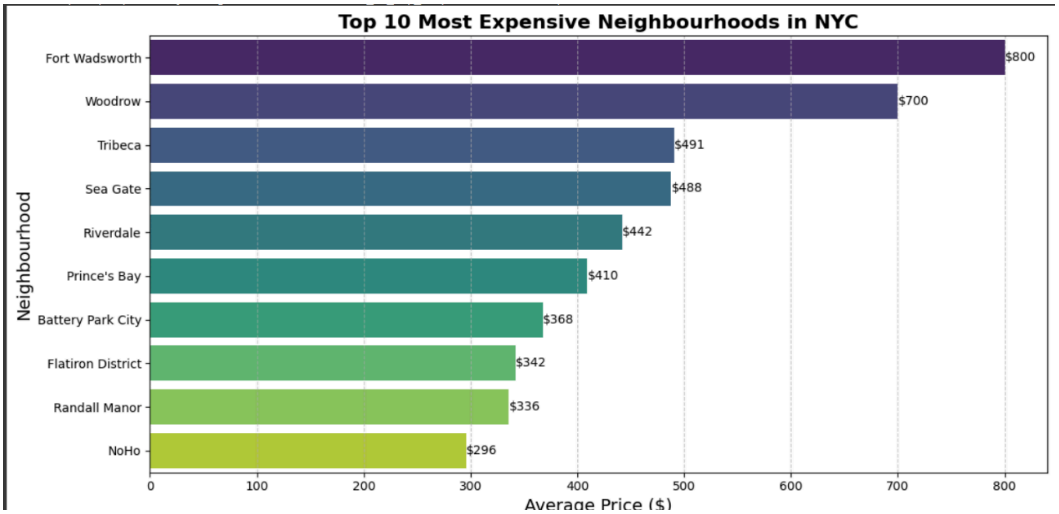
Average Room Price In Each Neighbourhood Group

- Manhattan -> $196.88 (33.7%)
- Brooklyn -> $124.38 (21.3%)
- Staten Island -> $114.81 (19.6%)
- Queens -> $99.52 (17.0%)
- Bronx -> $87.50 (15.0%)



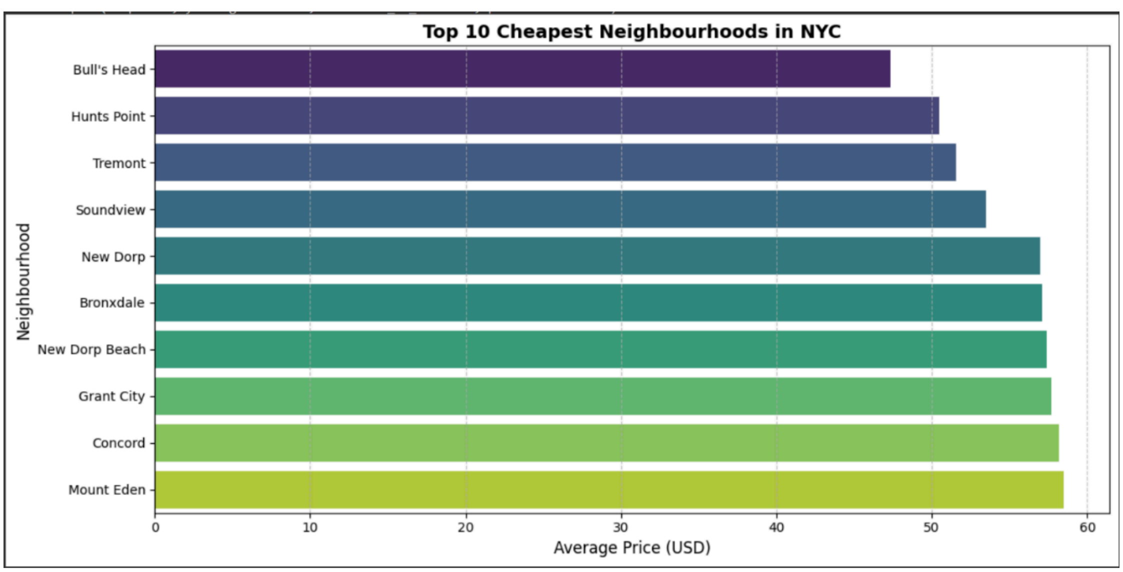
Top 10 Most Expensive Neighbourhoods In NYC

- Fort Wadsworth -> $800 (100%)
- Woodrow -> $700 (87.5%)
- Tribeca -> $491 (61.4%)
- Sea Gate -> $488 (61.0%)
- Riverdale -> $442 (55.3%)
- Prince's Bay -> $410 (51.3%)
- Battery Park City -> $368 (46.0%)
- Flatiron District -> $342 (42.8%)
- Randall Manor -> $336 (42.0%)
- NoHo -> $296 (37.0%)



Top 10 Cheapest Neighbourhoods In Nyc

- Fort Wadsworth -> $800 (100%)
- Woodrow -> $700 (87.5%)
- Tribeca -> $491 (61.4%)
- Sea Gate -> $488 (61.0%)
- Riverdale -> $442 (55.3%)
- Prince's Bay -> $410 (51.3%)
- Battery Park City -> $368 (46.0%)
- Flatiron District -> $342 (42.8%)
- Randall Manor -> $336 (42.0%)
- NoHo -> $296 (37.0%)



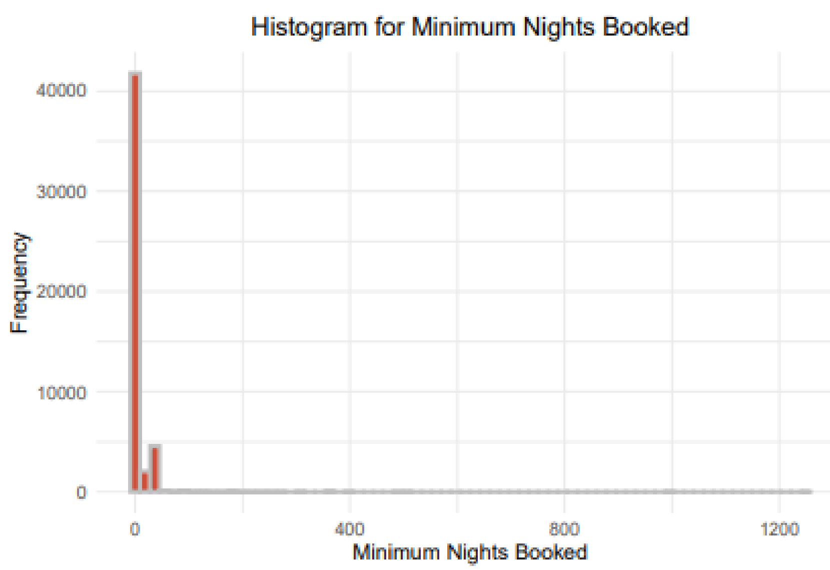
Quantitative Variables

Frequency distribution
Minimum Nights an Airbnb is booked for at a time

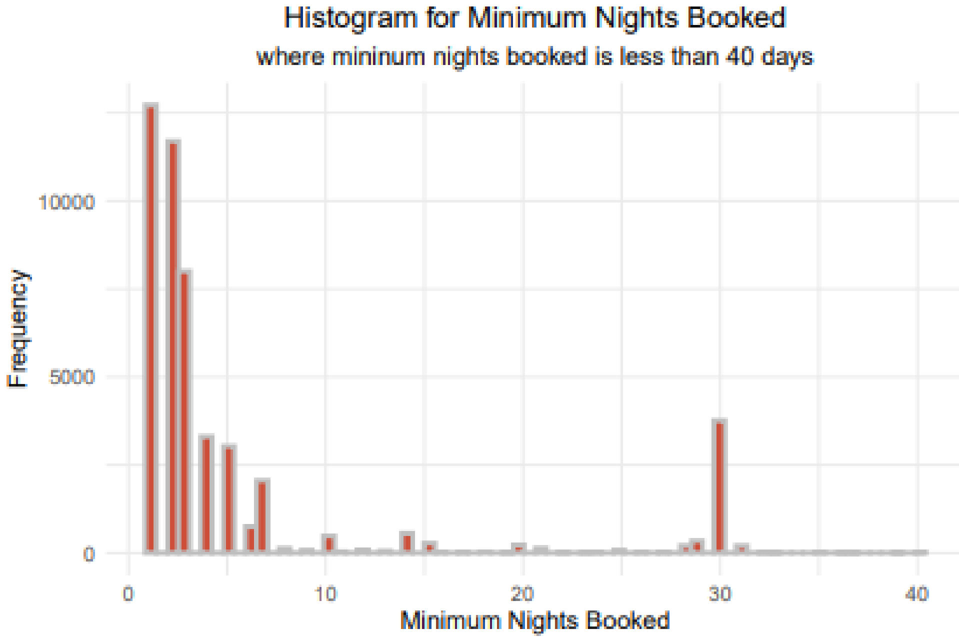
```
# Plot for all minimum_nights
plt.figure(figsize=(10, 6))
sns.histplot(data=AB_NYC_2019, x='minimum_nights', bins=70, color='red')
plt.title('Histogram for Minimum Nights Booked', fontsize=16, weight='bold')
plt.xlabel('Minimum Nights Booked')
plt.ylabel('Frequency')
plt.grid(False)
plt.show()

# Filter data where minimum_nights is less than or equal to 40
filtered_data = AB_NYC_2019[AB_NYC_2019['minimum_nights'] <= 40]

# Plot for minimum_nights <= 40
plt.figure(figsize=(10, 6))
sns.histplot(data=filtered_data, x='minimum_nights', bins=70, color='reds')
plt.title('Histogram for Minimum Nights Booked', fontsize=16, weight='bold')
plt.suptitle('Where minimum nights booked is less than 40 days', fontsize=12)
plt.xlabel('Minimum Nights Booked')
plt.ylabel('Frequency')
plt.grid(False)
plt.show()
```



The histogram shows that over 40,000 bookings require 0 to 50 minimum nights, indicating a strong concentration in this range. There are very few bookings beyond 400 nights, with an extremely small number extending up to 1200 nights. The data is heavily skewed toward shorter stays, with long-term bookings being rare outliers.



The histogram shows that the majority of bookings, over 12,000, require between 0 to 5 minimum nights. There are smaller but noticeable peaks around 10 nights and 30 nights, with about 1,000 to 5,000 bookings in those ranges. The frequency drops significantly after 5 nights, indicating fewer listings with longer minimum stay requirements. Overall, most bookings are for short stays, but some properties enforce specific requirements for stays of around 10 or 30 nights.

Exploratory Data Analysis

On an average, what price would you have to pay for each Room Type available?

Average Price per Room Type

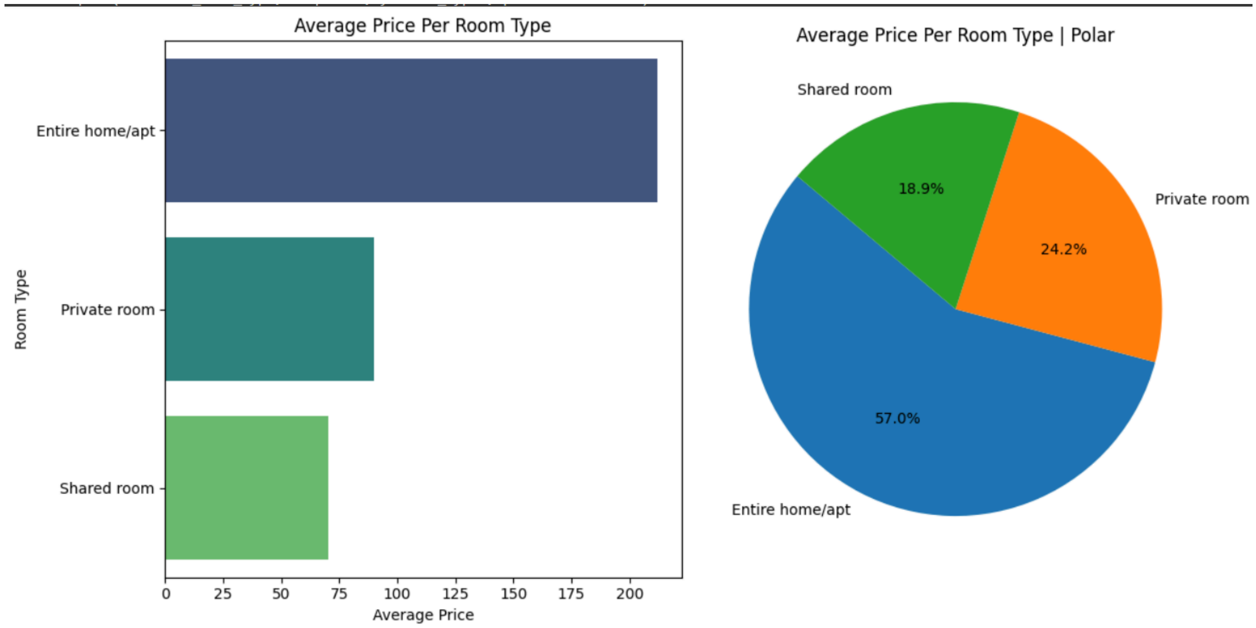
```
mean_room_type = airbnb_data.groupby('room_type')['price'].mean().reset_index()
mean_room_type['percent'] = (mean_room_type['price'] / mean_room_type['price'].sum()) * 100

# Plot Average Price Per Room Type
plt.figure(figsize=(12, 6))

# Plot A: Horizontal bar plot
plt.subplot(1, 2, 1)
sns.barplot(data=mean_room_type, x='price', y='room_type', palette='viridis')
plt.title('Average Price Per Room Type')
plt.xlabel('Average Price')
plt.ylabel('Room Type')

# Plot B: Polar plot (Pie chart representation)
plt.subplot(1, 2, 2)
plt.pie(mean_room_type['price'], labels=mean_room_type['room_type'], autopct='%1.1f%%', startangle=140)
plt.title('Average Price Per Room Type | Polar')

plt.tight_layout()
plt.show()
```



- Entire home/apartment listings are significantly more expensive than private or shared rooms, averaging around $200. This suggests that properties offering more space and privacy demand a higher price.
- Private rooms are moderately priced at $100 on average, offering a middle ground between cost and privacy.
- Shared rooms are the least expensive option at $50 on average, likely due to their shared nature, which implies reduced privacy and space.

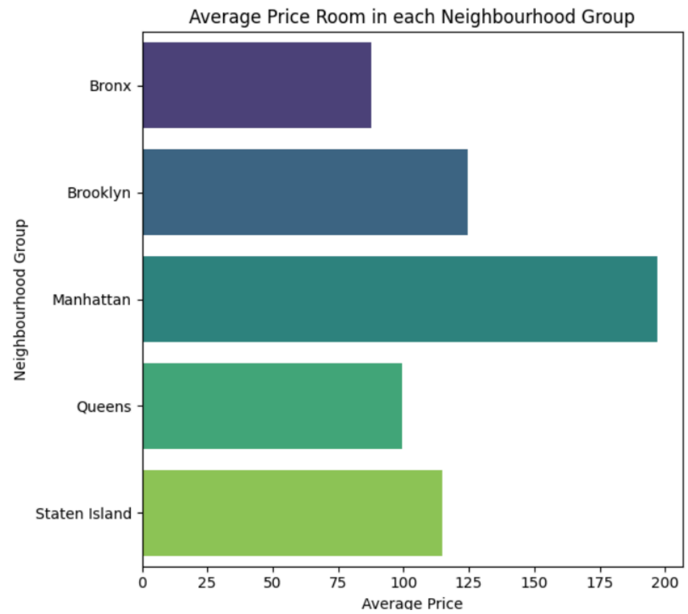
Average Price Room In Each Neighbourhood Group

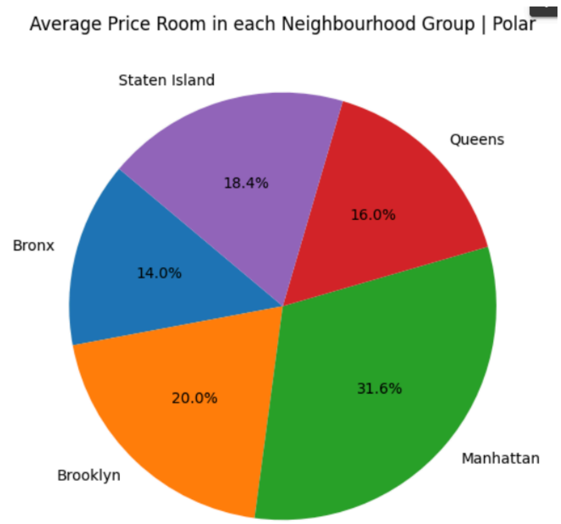
```
# Plot Average Price Room in each Neighbourhood Group
plt.figure(figsize=(12, 6))

# Plot A_2: Horizontal bar plot
plt.subplot(1, 2, 1)
sns.barplot(data=mean_nhg, x='price', y='neighbourhood_group', palette='viridis')
plt.title('Average Price Room in each Neighbourhood Group')
plt.xlabel('Average Price')
plt.ylabel('Neighbourhood Group')

# Plot B_2: Polar plot (Pie chart representation)
plt.subplot(1, 2, 2)
plt.pie(mean_nhg['price'], labels=mean_nhg['neighbourhood_group'], autopct='%1.1f%%', startangle=140)
plt.title('Average Price Room in each Neighbourhood Group | Polar')

plt.tight_layout()
plt.show()
```





Manhattan is the most expensive area, with nearly one-third of the total market by price share (31.6%) and an average price near $200. Its centrality and high demand likely contribute to this.

- Brooklyn follows with a significant share (20%) and a mid-range average price around $125, reflecting its growing popularity.
- Staten Island has a surprisingly large price share (18.4%) despite being the most affordable at $50, indicating that many lower-cost listings exist there.
- Bronx and Queens offer relatively affordable accommodation options, but their shares remain lower at 14% and 16%, respectively.

Business Recommendations for Airbnb

1. Expand Affordable Accommodation Options:

    - Increase Shared and Private Room Listings: Shared rooms account for only 18.9% of listings, while private rooms make up 24.2%. Boosting these options can attract budget-conscious travelers who currently find fewer choices.

    - Introduce Mid-Tier Listings: The average price gap between private rooms ($100) and entire homes ($200) indicates a need for mid-tier options like studio apartments. These can appeal to guests seeking privacy without the high cost.

2. Implement Dynamic Pricing Strategies:

    - Dynamic Pricing for Entire Homes: Entire homes represent 57% of listings and average around $200 per night. Using dynamic pricing during off-peak seasons could fill vacancies and increase occupancy rates.

    - Dynamic Pricing in Manhattan: Manhattan has the highest average price (~$200) and holds 31.6% of the market. Implementing dynamic pricing during slower periods can attract more guests.

3. Promote Listings in Budget-Friendly Neighborhoods:

- Increase Listings in Staten Island: With the lowest average price at $50 and 18.4% market share, promoting Staten Island can draw budget travelers and increase occupancy.
- Encourage Listings in the Bronx and Queens: Average prices in these areas range from $75-$100, but they represent only 14% and 16% of the market, respectively. More listings here can balance the distribution and attract mid-range customers.

4. Highlight Amenities in Private Rooms:
    - Emphasize Amenities: Private rooms average $100. Highlighting unique features (like private bathrooms and kitchens) can justify their price and make them more appealing compared to entire homes.
5. Enhance Promotion of Shared Rooms:
    - Target Group Travelers and Students: Shared rooms, priced around $50, represent only 18.9% of listings. Promoting these options can increase demand among budget-conscious groups.
6. Market Brooklyn as an Affordable Alternative:
    - Highlight Brooklyn's Value: With an average price of $125 and 20% market share, Brooklyn is becoming a popular choice. Promoting it as a trendy yet affordable option compared to Manhattan can attract more visitors.

By following these recommendations, Airbnb can better serve diverse travelers, increase bookings, and enhance overall business performance.