In [1]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
pd.set_option('display.max_columns', None)
```

In [2]:
```
df=pd.read_csv(r"C:\Users\ASUS\Studia\Python dla analityków\Airbnb.csv")
```

Step 1: Exploring the DataFrame

In [4]:
```
df.head()
```

Out[4]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood
0	1312228	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382.0	Walter	Brooklyn	Clinton Hil
1	45277537	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835.0	Jeniffer	Manhattan	Hell's Kitchen
2	9.71E+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354.0	Joshua	Manhattan	Chelsea
3	3857863	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271.0	John And Catherine	Manhattan	Washington Heights
4	40896611	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963.0	Stay With Vibe	Manhattan	Murray Hil

In [5]:
```
df.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                            20735 non-null  object
 2   host_id                         20735 non-null  float64
 3   host_name                       20735 non-null  object
 4   neighbourhood_group             20735 non-null  object
 5   neighbourhood                   20728 non-null  object
 6   latitude                        20728 non-null  float64
 7   longitude                       20728 non-null  float64
 8   room_type                       20728 non-null  object
 9   price                           20701 non-null  float64
 10  minimum_nights                  20728 non-null  float64
 11  number_of_reviews               20728 non-null  float64
 12  last_review                     20728 non-null  object
 13  reviews_per_month               20728 non-null  float64
 14  calculated_host_listings_count  20728 non-null  float64
 15  availability_365                20728 non-null  float64
 16  number_of_reviews_ltm           20728 non-null  float64
 17  license                         20735 non-null  object
 18  rating                          20735 non-null  object
 19  bedrooms                        20735 non-null  object
 20  beds                            20735 non-null  float64
 21  baths;                          20735 non-null  object
dtypes: float64(11), object(11)
memory usage: 3.5+ MB
```

In [6]:
```
df.describe()
```

Out[6]:

	host_id	latitude	longitude	price	minimum_nights	num
count	2.073500e+04	20728.000000	20728.000000	20701.000000	20728.000000	
mean	1.750325e+08	40.726842	-73.939162	187.438288	28.558954	
std	1.726028e+08	0.060282	0.061358	1022.830883	33.540051	
min	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	
25%	2.044260e+07	40.684177	-73.980710	80.000000	30.000000	
50%	1.089827e+08	40.722876	-73.949593	125.000000	30.000000	
75%	3.144677e+08	40.763120	-73.917477	199.000000	30.000000	
max	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	

Step 2: Data Cleaning

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
id                                0
name                             35
host_id                          35
host_name                        35
neighbourhood_group              35
neighbourhood                    42
latitude                         42
longitude                        42
room_type                        42
price                            69
minimum_nights                   42
number_of_reviews                42
last_review                      42
reviews_per_month                42
calculated_host_listings_count   42
availability_365                 42
number_of_reviews_ltm            42
license                          35
rating                           35
bedrooms                         35
beds                             35
baths;                           35
dtype: int64
```

In [9]:
```
df.dropna(inplace=True)
df.isnull().sum()
```

Out[9]:
```
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
```

In [10]:
```
df.shape
```

Out[10]:
```
(20701, 22)
```

In [11]:
```
df.duplicated().sum()
```

Out[11]:
```
12
```

In [12]:
```
duplicates = df[df.duplicated()]
duplicates
```

Out[12]:

	id	name	host_id	host_name	neighbourhood_group	neighbour
6	45277537	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835.0	Jeniffer	Manhattan	Hell's Ki
7	9.71E+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354.0	Joshua	Manhattan	Ch
8	3857863	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271.0	John And Catherine	Manhattan	Washi He
9	40896611	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963.0	Stay With Vibe	Manhattan	Murra
10	49584983	Rental unit in New York · ★5.0 · 1 bedroom · 1...	51501835.0	Jeniffer	Manhattan	Hell's Ki
20736	7.99E+17	Rental unit in New York · 2 bedrooms · 2 beds ...	224733902.0	CozySuites Copake	Manhattan	Upper Eas
20737	5.93E+17	Rental unit in New York · ★4.79 · 2 bedrooms ·...	23219783.0	Rob	Manhattan	West V
20738	9.23E+17	Loft in New York · ★4.33 · 1 bedroom · 2 beds ...	520265731.0	Rodrigo	Manhattan	Gree V

	id	name	host_id	host_name	neighbourhood_group	neighbour
20739	13361613	Rental unit in New York · ★4.89 · 2 bedrooms ·...	8961407.0	Jamie	Manhattan	H
20740	51195659	Rental unit in New York · Studio · 1 bed · 1 bath	51501835.0	Jeniffer	Manhattan	China
20741	25234732	Rental unit in New York · ★4.41 · 1 bedroom · ...	1497427.0	Mara	Manhattan	Upper Eas
20742	3339399	Rental unit in New York · ★4.73 · 1 bedroom · ...	2119276.0	Urban Furnished	Manhattan	West V

In [13]:
```
df.drop_duplicates(inplace=True)
df.duplicated().sum()
```

Out[13]:
```
0
```

In [14]:
```
df['host_id']=df['host_id'].astype(object)
df.dtypes
```

Out[14]:
```
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                              float64
baths;                             object
dtype: object
```

In [15]:
```
df['rating'] = pd.to_numeric(df['rating'], errors='coerce')
```

Step 3: Exploratory Data Analysis (EDA)

In [17]:
```
df.columns = df.columns.str.strip().str.rstrip(';')
print(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')
```

In [18]:
```
df_room_type_count=pd.DataFrame(df["room_type"].value_counts())
df_room_type_count
```

Out[18]:

room_type	count
Entire home/apt	11517
Private room	8769
Shared room	291
Hotel room	112

Room Types - Homes/apartments - 11,517, indicating the dominance of entire property rentals on the platform. - Private rooms = 8,769 listings. - Shared rooms - 291 listings. - Hotel rooms - 112 listings, highlighting the dominance of private owners over hotels. ---

In [19]:
```
listings_count =pd.DataFrame(df['neighbourhood_group'].value_counts())
listings_count
```

Out[19]:

neighbourhood_group	count
Manhattan	8025
Brooklyn	7680
Queens	3748
Bronx	947
Staten Island	289

Neighborhood Groups

- Manhattan and Brooklyn dominate in terms of the number of listings. Manhattan leads slightly with 8,025 listings compared to Brooklyn's 7,680. This may reflect Manhattan's prestige and tourist appeal.
- Queens ranks third with 3,748 listings, likely offering more affordable options compared to Manhattan and Brooklyn.
- Bronx (947) and Staten Island (289) have significantly fewer listings. This could be due to lower popularity among tourists or fewer properties available for rent.

In [21]:
```
relevant_columns = ['price', 'minimum_nights', 'number_of_reviews',
                    'reviews_per_month', 'calculated_host_listings_count',
                    'availability_365', 'bedrooms', 'beds', 'baths', 'rating']

df[relevant_columns].describe()
```

Out[21]:

	price	minimum_nights	number_of_reviews	reviews_per_month	calculat
count	20689.000000	20689.000000	20689.000000	20689.000000	
mean	187.455411	28.566871	42.541834	1.257530	
std	1023.125534	33.567680	73.499696	1.906069	
min	10.000000	1.000000	1.000000	0.010000	
25%	80.000000	30.000000	4.000000	0.210000	
50%	125.000000	30.000000	14.000000	0.650000	
75%	199.000000	30.000000	49.000000	1.800000	
max	100000.000000	1250.000000	1865.000000	75.490000	

Price:

The typical price for an Airbnb rental in New York City is about **$187** *per night*. However, prices vary greatly, with some listings priced as low as **$10**, while others can go up to **$100,000**, likely reflecting luxury or one-of-a-kind properties. The largest standard deviation (1022.80) shows that most listings are priced lower, but a few high-end listings are pulling the average upwards.

Minimum Nights:

Hosts generally require a minimum stay of about 28 nights, which is relatively long. This suggests that some properties are aimed at guests looking for extended stays. While a few listings allow just 1-night stays, others have minimums as long as 1250 nights, indicating that some properties are geared toward long-term rentals.

Number of Reviews:

On average, an Airbnb listing has approximately **42** reviews.
However, the number of reviews can vary widely, with some properties having just 1 review, while others have as many as 1865 reviews, indicating a difference in popularity and how often they're booked.

Reviews Per Month:

Listings typically receive an average of **1.25** reviews per month, which suggests that bookings are fairly consistent.
Some listings, however, can get up to 75 reviews per month, pointing to very high booking rates and popularity.

Host Listings:

On average, hosts manage **18.8** listings, indicating that many are professional property managers.
Some hosts have only one listing, typical for individual hosts, while the busiest hosts manage up to 713 listings, suggesting a more commercial operation.

Availability (365 Days):

On average, listings are available for about **206 days** a year, meaning many are booked or unavailable for part of the year.
While some listings are available all year round, others are fully booked or unavailable for the entire year.

Beds:

The typical listing offers **1.72 beds**, indicating that many are smaller properties, likely 1-bedroom apartments or studios.
The highest number of beds in a listing is **42**, showing that there are also larger properties designed for group stays.

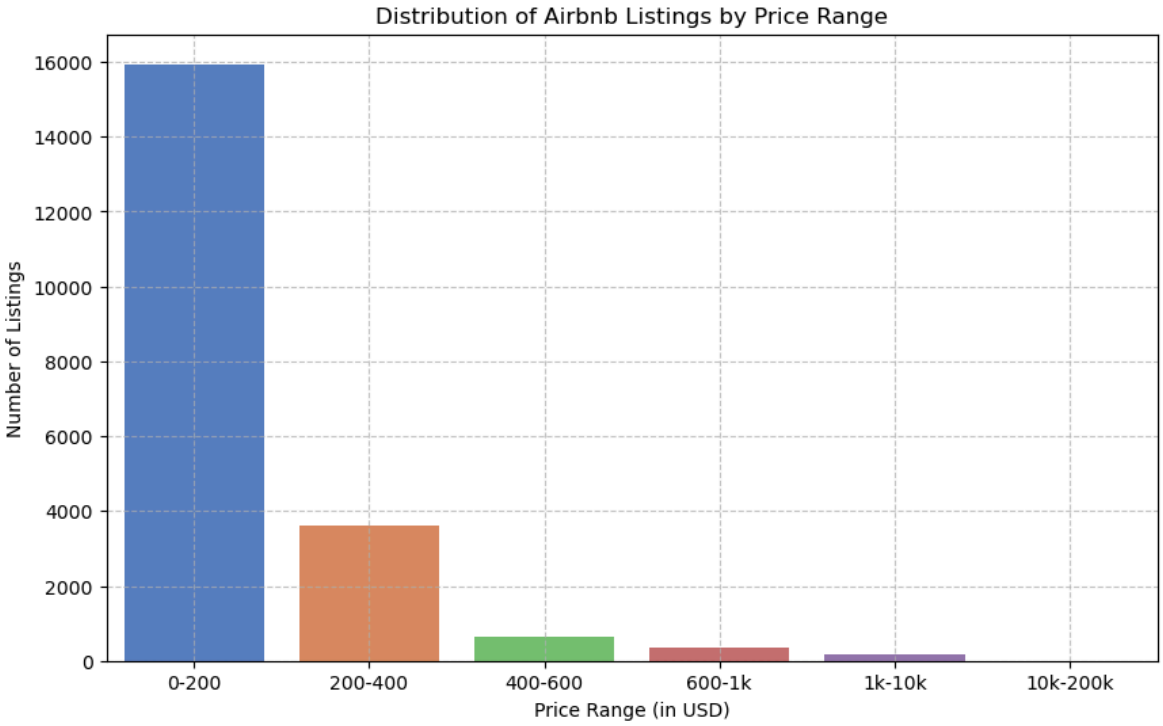
In [23]:
```
plt.figure(figsize=(10,5))
sns.histplot(df['price'],bins=50,kde=True)
plt.title("Price Distribuition of Airbnb Listings")
plt.xlabel('Price (in USD)')
plt.ylabel("Frequency")
plt.show()
```

Price Distribution:

An attempt was made to display the full distribution of Airbnb listing prices. However, it's clear that most prices are concentrated around **$0**, with the x-axis stretching all the way up to **$100,000**.
This happens because a few extremely high-priced listings (outliers) distort the distribution, making it harder to accurately represent the prices of most listings.
As a result, the majority of listings, which are priced much lower, appear squeezed into a small section of the plot, making it challenging to identify any meaningful trends

In [25]:
```
bins = [0, 200, 400, 600, 1000, 10000,200000]

bin_labels = ['0-200', '200-400', '400-600', '600-1k', '1k-10k','10k-200k']

df['price_bin'] = pd.cut(df['price'], bins=bins, labels=bin_labels, include_lowe

plt.figure(figsize=(10,6))
sns.countplot(x='price_bin', data=df, palette='muted')
plt.title('Distribution of Airbnb Listings by Price Range')
plt.xlabel('Price Range (in USD)')
plt.ylabel('Number of Listings')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13872\1338153536.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x='price_bin', data=df, palette='muted')
```



Price Distribution:

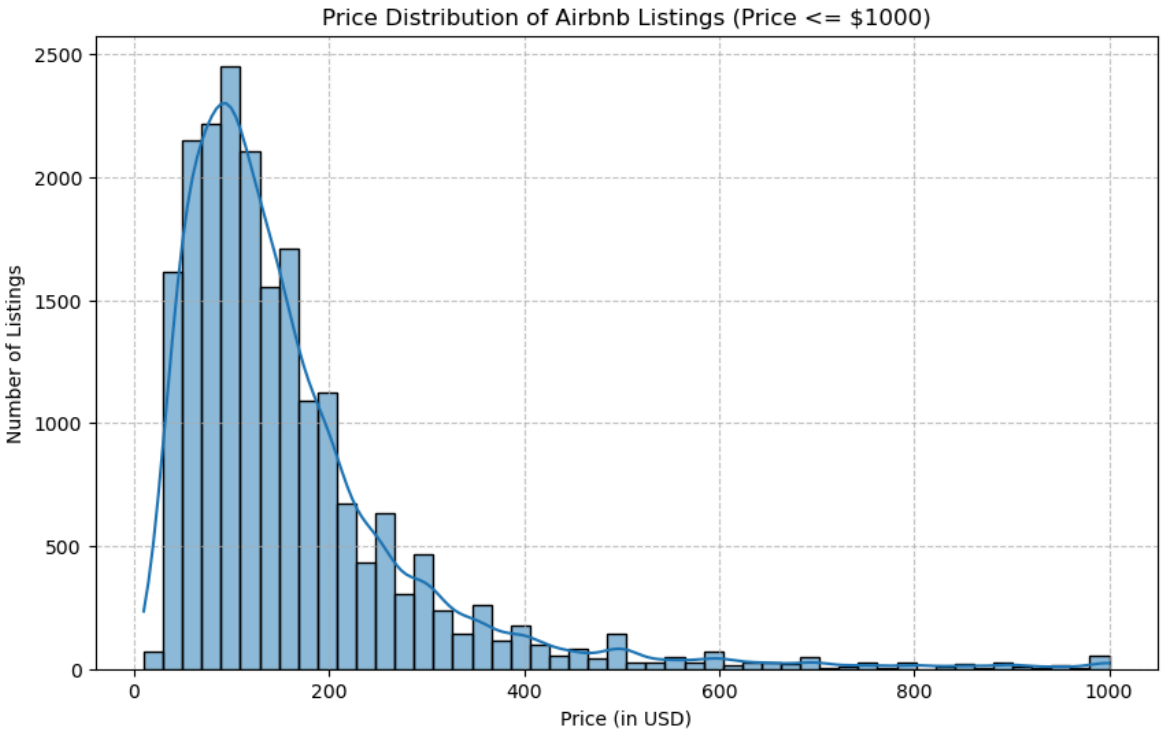
It's clear that most Airbnb listings are priced between **$0** and **$200**, with a significant number falling in the **$200**-**$400** range.
There are fewer listings in the **$400**-**$600** price range, and the number of listings drops sharply beyond $600.

Only a small fraction of listings are priced above **$1,000**, and extremely high-priced listings (e.g., above **$10,000**) make up a very small part of the data.
These outliers can skew visualizations, making it harder to focus on the majority of listings that are priced more typically.

To enhance clarity, outliers will be addressed by applying a price cap and filtering the data, allowing for a more focused analysis of the majority of Airbnb listings

In [27]:
```
filtered_df = df[df['price'] <= 1000]
```

In [28]:
```
plt.figure(figsize=(10,6))
sns.histplot(filtered_df['price'], bins=50, kde=True)
plt.title('Price Distribution of Airbnb Listings (Price <= $1000)')
plt.xlabel('Price (in USD)')
plt.ylabel('Number of Listings')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



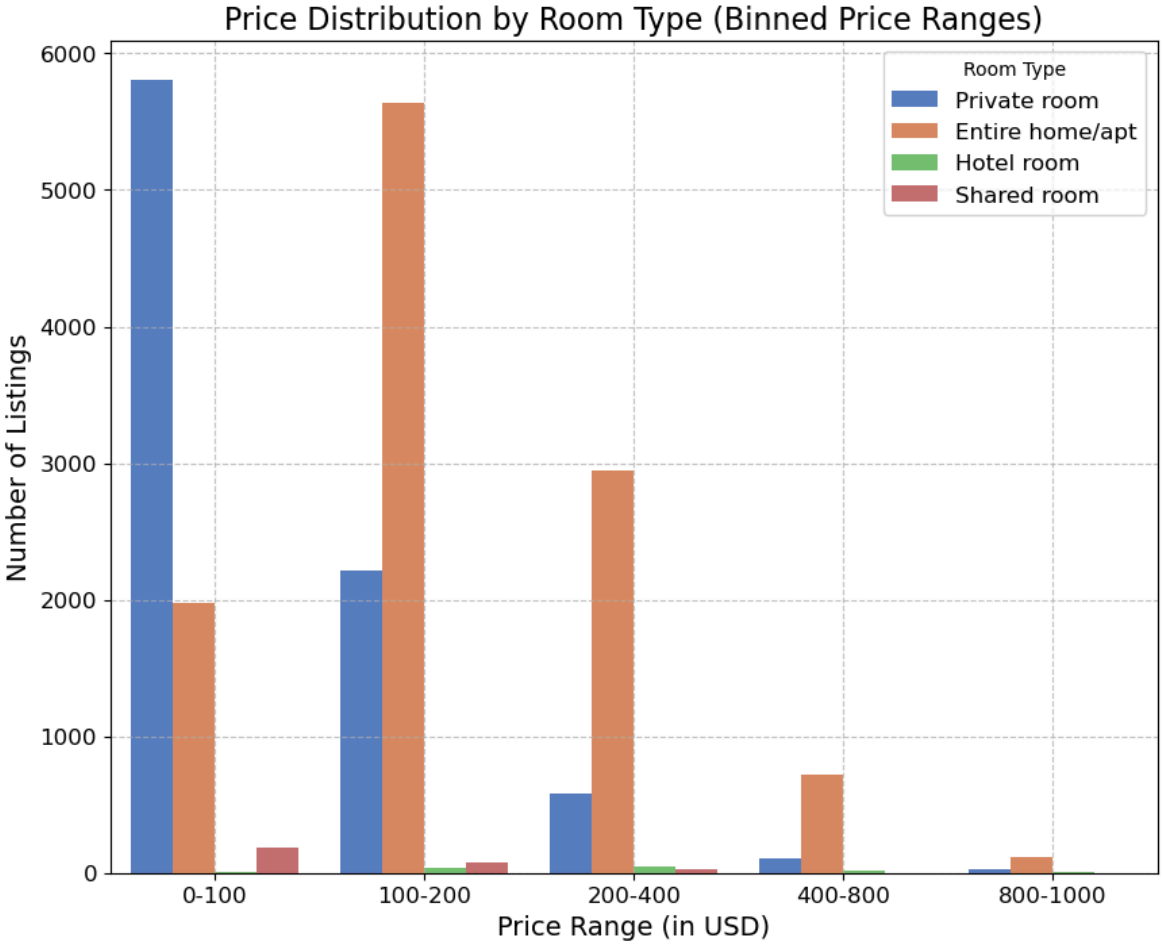
Adjusted Price Distribution:

In this second plot, the dataset was limited to include only listings priced at $1,000 or below, which covers the vast majority of listings.
As a result, the distribution is now much clearer:

Most listings are priced between **$50** and **$300**, with a peak around **$100** per night.
This plot provides a much clearer picture of the typical prices for Airbnb listings in New York City, free from the distortion caused by extreme outliers.

In [30]:
```
bins = [0, 100, 200, 400, 800, 1000]
bin_labels = ['0-100', '100-200', '200-400', '400-800', '800-1000']

filtered_df['price_bin'] = pd.cut(filtered_df['price'], bins=bins, labels=bin_la

plt.figure(figsize=(10,8))
sns.countplot(x='price_bin', hue='room_type', data=filtered_df, palette='muted')

plt.title('Price Distribution by Room Type (Binned Price Ranges)', fontsize=16)
plt.xlabel('Price Range (in USD)', fontsize=14)
plt.ylabel('Number of Listings', fontsize=14)

plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

plt.legend(title='Room Type', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7)

plt.show()
```

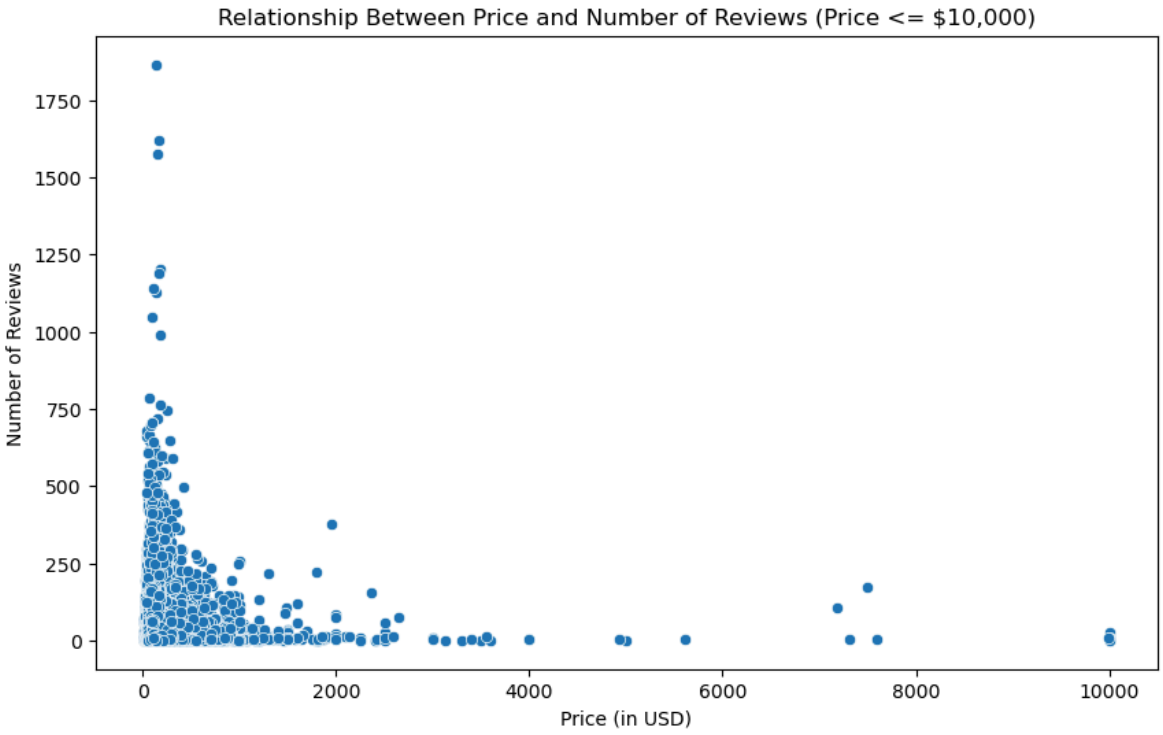
```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13872\2968815239.py:4: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['price_bin'] = pd.cut(filtered_df['price'], bins=bins, labels=bin_l
abels, include_lowest=True)
```



Price Concentration:

Most Airbnb listings are concentrated in the lower price ranges, especially for private rooms and entire homes/apartments.
This supports the idea that Airbnb is primarily a platform for more affordable, short-term stays.
As prices rise, the number of listings drops sharply, with only a small number of high-end listings exceeding $1,000 USD.

In [74]:
```
filtered_df = df[df['price'] <= 10000]
plt.figure(figsize=(10,6))
sns.scatterplot(x='price', y='number_of_reviews', data=filtered_df)
plt.title('Relationship Between Price and Number of Reviews (Price <= $10,000)')
plt.xlabel('Price (in USD)')
plt.ylabel('Number of Reviews')
plt.show()
```



Price vs. Number of Reviews:

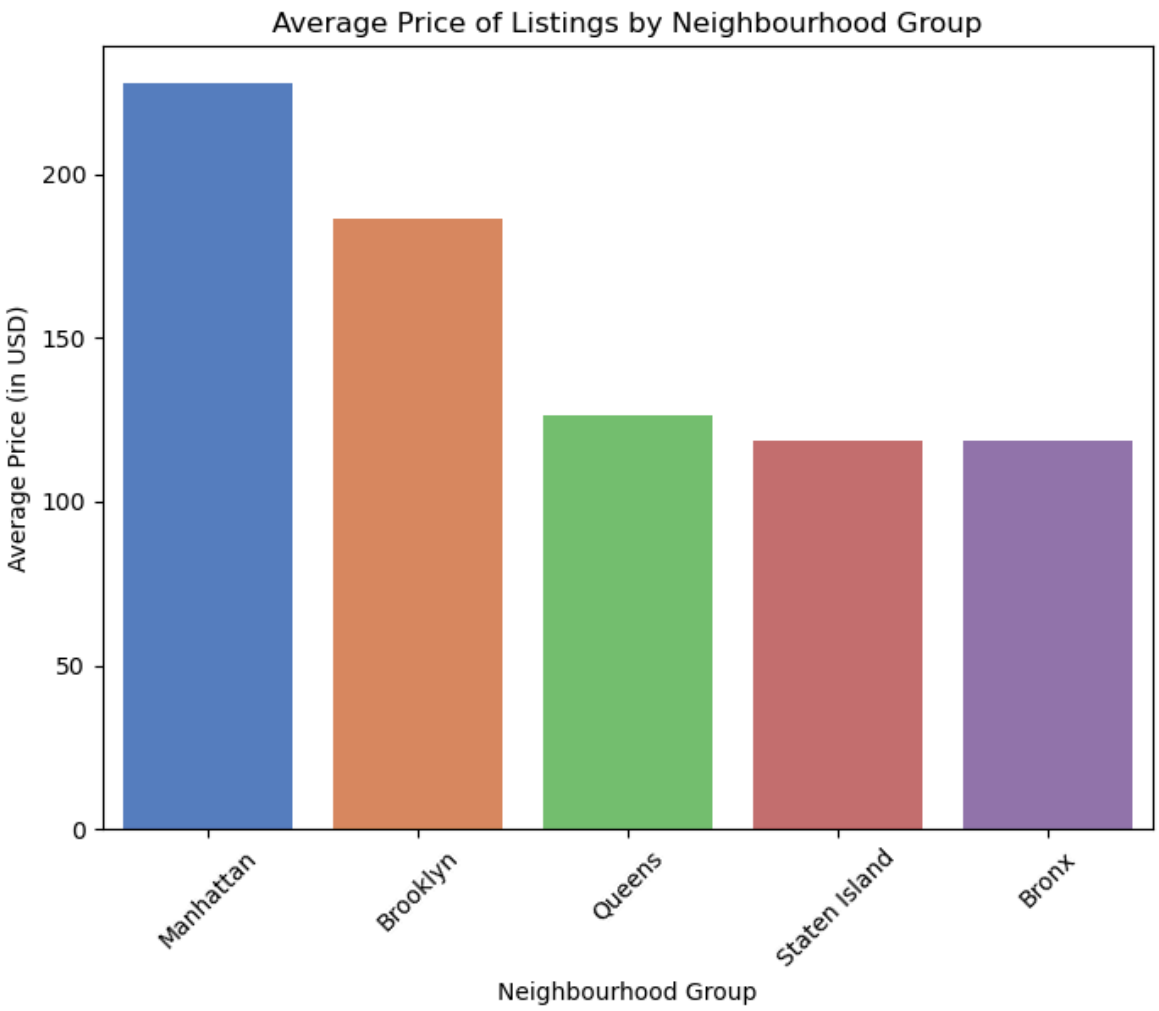
Lower-priced listings (below $1,000 USD) tend to have more reviews, with many surpassing 500 reviews.
This indicates that affordable listings are booked more frequently, leading to a higher number of reviews.

Higher-priced listings (above $2,000 USD) typically have fewer reviews, suggesting that luxury or high-cost properties are booked less often, likely catering to a more selective audience.

This shows an inverse relationship between price and the number of reviews: cheaper listings receive more reviews, while expensive listings generally have fewer.

In [34]:
```
average_price_neighbourhood=df.groupby('neighbourhood_group')['price'].mean().so

plt.figure(figsize=(8,6))
sns.barplot(x='neighbourhood_group', y='price', data=average_price_neighbourhood
plt.title('Average Price of Listings by Neighbourhood Group')
plt.xlabel('Neighbourhood Group')
plt.ylabel('Average Price (in USD)')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13872\2714812310.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x='neighbourhood_group', y='price', data=average_price_neighbourhoo
d, palette='muted')
```



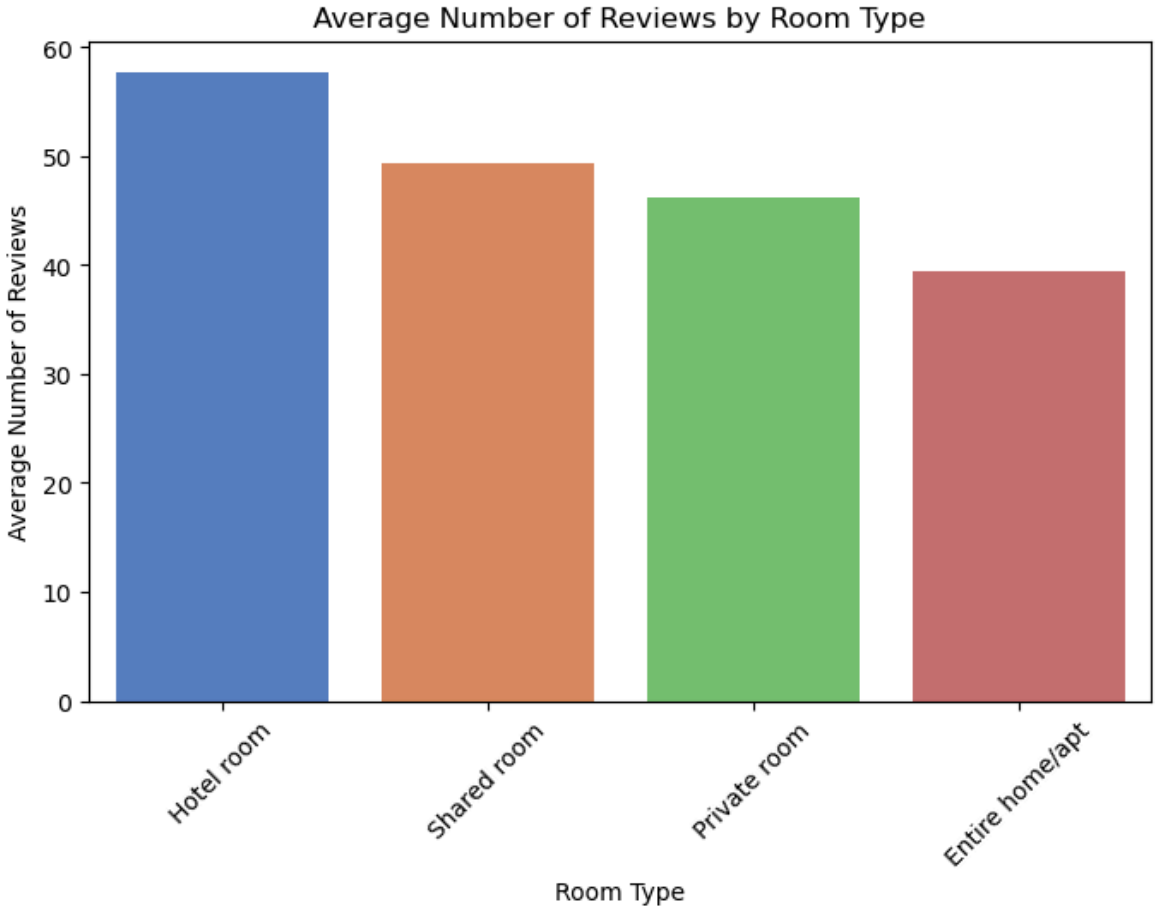
Price Variation Across Neighborhoods:

The bar plot shows how Airbnb prices differ across the neighborhoods of NYC:

- **Manhattan** leads the chart, with average prices exceeding $200 USD per night. This is expected, as it's a prime location with high demand.
- **Brooklyn** follows, with listings averaging around $150 USD per night-offering a popular, more affordable alternative to Manhattan.
- **Queens, Staten Island, and the Bronx** offer the most budget-friendly options, all averaging below $150 USD per night, making them ideal for travelers seeking affordability.

In [40]:
```
room_reviews = df.groupby('room_type')['number_of_reviews'].mean().sort_values(a

plt.figure(figsize=(8, 5))
sns.barplot(x='room_type', y='number_of_reviews', data=room_reviews,palette='mut
plt.xlabel('Room Type')
plt.ylabel('Average Number of Reviews')
plt.title('Average Number of Reviews by Room Type')
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_13872\428948131.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x='room_type', y='number_of_reviews', data=room_reviews,palette='mu
ted')
```



Insights

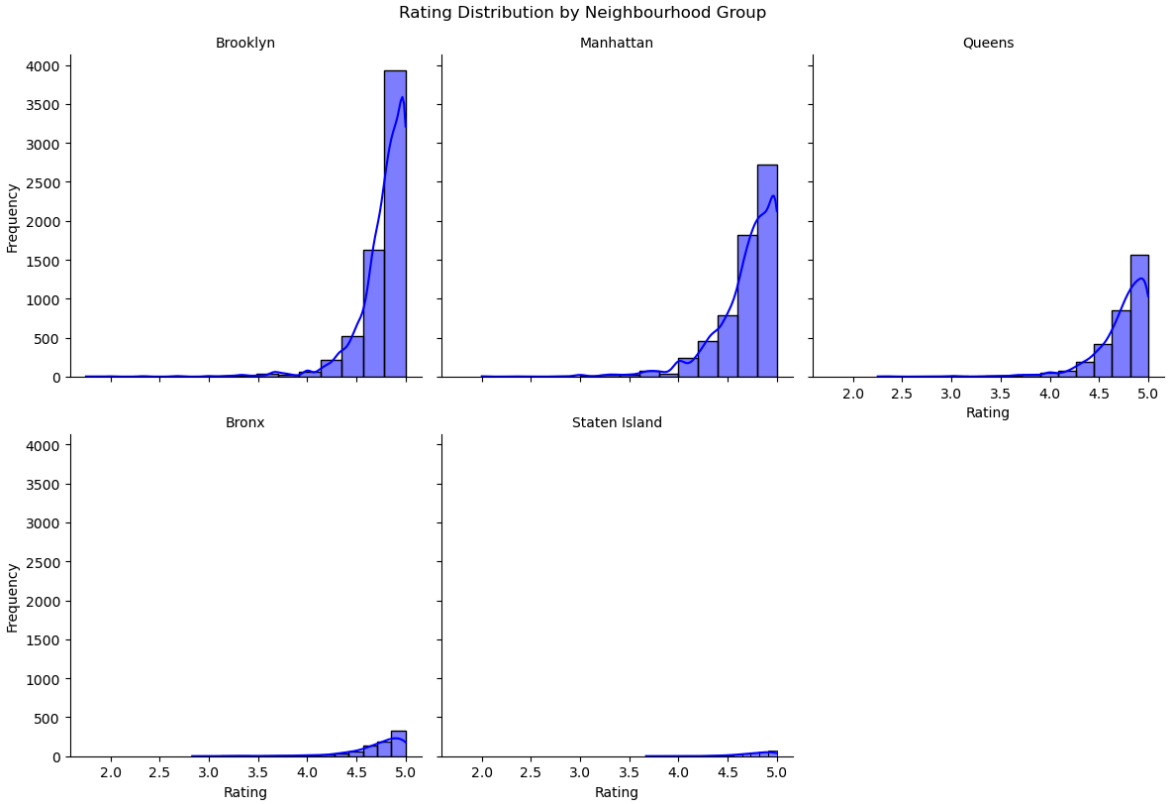
- **Hotel room**: This category has the highest average number of reviews per listing, around **55**. Despite having relatively few listings (only **112**), they tend to receive high engagement from guests.
- **Shared room**: The second-highest average, around **50** reviews per listing. Shared rooms are likely a niche but engaging choice for guests who actively leave reviews.
- **Private room**: Falls in third place with an average of about **45** reviews per listing, indicating a popular choice with steady engagement.
- **Entire home/apt**: This category has the lowest average, around **40** reviews per listing. Despite dominating with the largest number of listings (**11,517**), the average engagement per listing is comparatively lower.

Observations

- **High engagement for low-volume categories**:

    - Hotel rooms and shared rooms, which have significantly fewer listings, attract guests who are more likely to leave reviews.
- **Lower engagement for high-volume category**:
    - Entire homes/apartments, despite their large listing count, show relatively lower average engagement per listing. This might result from diluted guest experience or less incentivized review behavior in this category.

In [42]:
```
facet = sns.FacetGrid(data=df, col="neighbourhood_group", col_wrap=3, height=4,
facet.map(sns.histplot, "rating", kde=True, bins=15, color='b')
facet.set_titles('{col_name}')
facet.set_axis_labels('Rating', 'Frequency')
plt.suptitle('Rating Distribution by Neighbourhood Group ', y=1.02)
plt.show()
```



Rating Distribution Across Neighbourhoods:

This visualization shows how ratings are distributed across different neighbourhood groups. Each subplot represents a neighbourhood:

- **Brooklyn**
- **Manhattan**
- **Bronx**
- **Queens**
- **Staten Island**

The data is displayed using histograms along with Kernel Density Estimation (KDE) curves for a smoother view of the distribution.

Key Takeaways:

1. **Most Listings Have High Ratings**
    - The majority of ratings fall between **4.5 and 5.0**, indicating that most Airbnb hosts receive positive reviews.
    - Very few listings have low ratings, suggesting that negative experiences are either rare or underreported.
2. **Differences Between Neighbourhoods**
    - **Brooklyn** and **Manhattan** have the largest number of listings, with similar rating distributions.
    - **Queens** follows a similar trend but has fewer listings.
    - **Bronx** and **Staten Island** have the fewest listings, and their rating distributions are less pronounced, though they still cluster around the **4.5–5.0** range.

Insights:

- **High ratings dominate**, suggesting either good service quality or a tendency for guests to leave positive feedback.
- **Few low ratings** might indicate a bias—guests may be less inclined to leave negative reviews, or poorly rated listings are removed over time.

Summary of Airbnb Listings Analysis in NYC

1. Price Distribution and Types of Listings

- **Price variations:**
  The analysis revealed significant price differences based on neighborhood and listing type (entire apartment vs. private room). Listings in central or prestigious neighborhoods (e.g., Manhattan) tend to have higher rates.

- **Different types of rentals:**
  The dataset covers various types of accommodations, ranging from entire apartments to private rooms. Analyzing these categories provided insights into the most popular options and their average prices.

---

2. Popularity and User Engagement Analysis

- **Reviews as a popularity indicator:**
  The "number_of_reviews" variable, along with "last_review" date, helps identify which listings attract the most interest. A high number of reviews suggests greater credibility and appeal.

3. Insights into NYC's Short-Term Rental Market

- **Geographical differences:**
  The analysis categorized listings by borough and neighborhood groups (e.g., Manhattan, Brooklyn). The results suggest that listings in central areas or trendy neighborhoods command higher prices, aligning with tourist demand and shorter availability periods.

- **Customer preferences:**
  The popularity of certain listing types (entire apartments vs. private rooms) and high review counts indicate that guests prioritize both location and accommodation quality. This suggests that the market is responsive to specific needs—such as a preference for privacy and comfort in more exclusive areas.

In [ ]: