In [1]:

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

In [2]:

```python
df=pd.read_csv("AB_NYC_2019.csv")
```

In [3]:

```python
df
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48895 rows × 16 columns

In [4]:

```python
df.shape
```

Out[4]:

```
(48895, 16)
```

In [7]:

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

Out[7]:

```
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                     0
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

In [6]:

```
# Fill missing 'reviews_per_month' with the median
median_reviews_per_month = df['reviews_per_month'].median()
df['reviews_per_month'].fillna(median_reviews_per_month, inplace=True)
```

In [12]:

```
df['last_review'] = pd.to_datetime(df['last_review'], errors='coerce')
```

In [20]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 18 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                     48895 non-null  datetime64[ns]
 13  reviews_per_month               48895 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64
 15  availability_365                48895 non-null  int64
 16  review_year                     48895 non-null  int32
 17  review_month                    48895 non-null  int32
dtypes: datetime64[ns](1), float64(3), int32(2), int64(7), object(5)
memory usage: 6.3+ MB
```

In [15]:

```
df['last_review'].fillna('01/01/2025', inplace=True)
```

In [16]:

```
df.drop(columns=['review_year','review_month'],inplace=True)
```

In [17]:

```
df
```

Out[17]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73

3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48895 rows × 16 columns

In [18]:

```
df['review_year'] = df['last_review'].dt.year
df['review_month'] = df['last_review'].dt.month
```

In [19]:

```
df
```

Out[19]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73

48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48895 rows × 18 columns

In [24]:

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

Out[24]:

```
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                        0
reviews_per_month                  0
calculated_host_listings_count     0
availability_365                   0
review_year                        0
review_month                       0
dtype: int64
```

In [33]:

```
df=df.dropna(subset=['name','host_name'])
```

In [34]:

```
df
```

Out[34]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73

48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48858 rows × 18 columns

In [35]:

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

Out[35]:

```
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
review_year                       0
review_month                      0
dtype: int64
```

In [36]:

```
duplicates=df.duplicated().sum()
```

In [37]:

```
duplicates
```

Out[37]:

```
0
```

In [38]:

```
unique_values=df.nunique()
```

In [39]:

```
unique_values
```

Out[39]:

```
id                                48858
name                              47884
host_id                           37425
host_name                         11450
neighbourhood_group                   5
neighbourhood                       221
latitude                          19039
longitude                         14716
room_type                             3
price                               674
minimum_nights                      108
number_of_reviews                   394
last_review                        1765
reviews_per_month                   937
calculated_host_listings_count       47
```

```
availability_365                  366
review_year                        10
review_month                       12
dtype: int64
```

In [41]:
```
total_rows=df.shape[0]
```

In [42]:
```
unique_check=(unique_values==total_rows)
```

In [44]:
```
print(unique_check)
```

```
id                                 True
name                              False
host_id                           False
host_name                         False
neighbourhood_group               False
neighbourhood                     False
latitude                          False
longitude                         False
room_type                         False
price                             False
minimum_nights                    False
number_of_reviews                 False
last_review                       False
reviews_per_month                 False
calculated_host_listings_count    False
availability_365                  False
review_year                       False
review_month                      False
dtype: bool
```

In [45]:
```
print(unique_check[unique_check])
```

```
id    True
dtype: bool
```

In [46]:
```
df.head()
```

Out[46]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	roo
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	h
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	h
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	h

In [47]:
```
df.tail()
```

Out[47]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitu
48890	36484665	Charming one bedroom - newly	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.949

		renovated rowhouse						
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.933
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.948
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.991
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.989

In [106…:

```
df.describe()
```

Out[106]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_re
count	4.885800e+04	4.885800e+04	48858.000000	48858.000000	48858.000000	48858.000000	48858.0
mean	1.902335e+07	6.763169e+07	40.728941	-73.952170	152.740309	7.012444	23.2
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.0
25%	9.475980e+06	7.818669e+06	40.690090	-73.983070	69.000000	1.000000	1.0
50%	1.969114e+07	3.079133e+07	40.723070	-73.955680	106.000000	3.000000	5.0
75%	2.915765e+07	1.074344e+08	40.763107	-73.936280	175.000000	5.000000	24.0
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.0
std	1.098289e+07	7.862389e+07	0.054528	0.046159	240.232386	20.019757	44.5

In [50]:

```
df.shape
```

Out[50]:

```
(48858, 18)
```

In [117…:

```
df=pd.DataFrame(data=df)
```

In [118…:

```
df
```

Out[118]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lo
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-7
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-7
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-7
3	3831	Cozy Entire Floor of	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-7

		Brownstone						
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-7
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-7
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-7
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-7
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-7
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-7

48858 rows × 21 columns

In [123…

```python
# Filter out non-numeric columns
numeric_df = df.select_dtypes(include=[np.number])
# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()
# Display the correlation matrix
print(correlation_matrix)
```

```
                                      id   host_id  latitude  longitude  \
id                              1.000000  0.588221 -0.003116   0.091076   
host_id                         0.588221  1.000000  0.020193   0.127198   
latitude                       -0.003116  0.020193  1.000000   0.084819   
longitude                       0.091076  0.127198  0.084819   1.000000   
price                           0.010564  0.015328  0.033944  -0.149954   
minimum_nights                 -0.012038 -0.017027  0.025893  -0.062893   
number_of_reviews              -0.320020 -0.140273 -0.015198   0.059151   
reviews_per_month               0.220736  0.236371 -0.014039   0.137404   
calculated_host_listings_count  0.133224  0.154954  0.019548  -0.114746   
availability_365                0.085616  0.203743 -0.010775   0.082754   
review_year                     0.281886  0.161740  0.020623  -0.007427   
review_month                   -0.145110 -0.076717 -0.021387   0.029956   
price_per_night                 0.015898  0.043009  0.017673  -0.075108   
price_increased                 0.010564  0.015328  0.033944  -0.149954   
cumulative_sum                  0.997905  0.592025 -0.001909   0.088801   

                                   price  minimum_nights  number_of_reviews  \
id                              0.010564       -0.012038          -0.320020   
host_id                         0.015328       -0.017027          -0.140273   
latitude                        0.033944        0.025893          -0.015198   
longitude                      -0.149954       -0.062893           0.059151   
price                           1.000000        0.042804          -0.047949   
minimum_nights                  0.042804        1.000000          -0.081610   
number_of_reviews              -0.047949       -0.081610           1.000000   
reviews_per_month              -0.036829       -0.112294           0.568005   
calculated_host_listings_count  0.057460        0.131313          -0.072408   
availability_365                0.081817        0.145953           0.171855   
review_year                     0.075023        0.091112          -0.152750   
review_month                   -0.055910       -0.083885           0.176783   
```

```
price_per_night                 0.690993       -0.107503          -0.003786
price_increased                 1.000000        0.042804          -0.047949
cumulative_sum                  0.013106       -0.011756          -0.322082

                                reviews_per_month  \
id                                       0.220736   
host_id                                  0.236371   
latitude                                -0.014039   
longitude                                0.137404   
price                                   -0.036829   
minimum_nights                          -0.112294   
number_of_reviews                        0.568005   
reviews_per_month                        1.000000   
calculated_host_listings_count          -0.027130   
availability_365                         0.166016   
review_year                             -0.010731   
review_month                             0.120260   
price_per_night                          0.041027   
price_increased                         -0.036829   
cumulative_sum                           0.217516   

                                calculated_host_listings_count  \
id                                                    0.133224   
host_id                                               0.154954   
latitude                                              0.019548   
longitude                                            -0.114746   
price                                                 0.057460   
minimum_nights                                        0.131313   
number_of_reviews                                    -0.072408   
reviews_per_month                                    -0.027130   
calculated_host_listings_count                        1.000000   
availability_365                                      0.225784   
review_year                                           0.123874   
review_month                                         -0.094251   
price_per_night                                      -0.026356   
price_increased                                       0.057460   
cumulative_sum                                        0.133152   

                                availability_365  review_year  review_month  \
id                                      0.085616     0.281886     -0.145110   
host_id                                 0.203743     0.161740     -0.076717   
latitude                               -0.010775     0.020623     -0.021387   
longitude                               0.082754    -0.007427      0.029956   
price                                   0.081817     0.075023     -0.055910   
minimum_nights                          0.145953     0.091112     -0.083885   
number_of_reviews                       0.171855    -0.152750      0.176783   
reviews_per_month                       0.166016    -0.010731      0.120260   
calculated_host_listings_count          0.225784     0.123874     -0.094251   
availability_365                        1.000000     0.070330     -0.001127   
review_year                             0.070330     1.000000     -0.715980   
review_month                           -0.001127    -0.715980      1.000000   
price_per_night                         0.034470     0.050998     -0.026242   
price_increased                         0.081817     0.075023     -0.055910   
cumulative_sum                          0.090327     0.293704     -0.155351   

                                price_per_night  price_increased  \
id                                     0.015898         0.010564   
host_id                                0.043009         0.015328   
latitude                               0.017673         0.033944   
longitude                             -0.075108        -0.149954   
price                                  0.690993         1.000000   
minimum_nights                        -0.107503         0.042804   
number_of_reviews                     -0.003786        -0.047949   
reviews_per_month                      0.041027        -0.036829   
calculated_host_listings_count        -0.026356         0.057460   
availability_365                       0.034470         0.081817   
```

```
review_year                              0.050998        0.075023
review_month                            -0.026242       -0.055910
price_per_night                          1.000000        0.690993
price_increased                          0.690993        1.000000
cumulative_sum                           0.018387        0.013106

                                cumulative_sum
id                                    0.997905
host_id                               0.592025
latitude                             -0.001909
longitude                             0.088801
price                                 0.013106
minimum_nights                       -0.011756
number_of_reviews                    -0.322082
reviews_per_month                     0.217516
calculated_host_listings_count        0.133152
availability_365                      0.090327
review_year                           0.293704
review_month                         -0.155351
price_per_night                       0.018387
price_increased                       0.013106
cumulative_sum                        1.000000
```

In [51]:

```
df
```

Out[51]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48858 rows × 18 columns

In [53]:

```
df['price_per_night'] = df['price'] / df['minimum_nights']
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\3140661797.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df['price_per_night'] = df['price'] / df['minimum_nights']
```

In [54]:

```
# Group by 'neighbourhood_group' and 'room_type', calculate mean price per night and tot
grouped_df = df.groupby(['neighbourhood_group', 'room_type']).agg({
    'price_per_night': 'mean',
    'number_of_reviews': 'sum',
    'availability_365': 'mean'
}).reset_index()
```

In [55]:

```
grouped_df
```

Out[55]:

	neighbourhood_group	room_type	price_per_night	number_of_reviews	availability_365
0	Bronx	Entire home/apt	68.501386	11590	158.349206
1	Bronx	Private room	40.861001	16312	171.331288
2	Bronx	Shared room	44.643933	432	150.644068
3	Brooklyn	Entire home/apt	75.408559	266734	97.161311
4	Brooklyn	Private room	41.516116	213647	99.964240
5	Brooklyn	Shared room	31.558991	5793	178.007264
6	Manhattan	Entire home/apt	100.299537	235031	117.151175
7	Manhattan	Private room	66.217700	208823	101.914963
8	Manhattan	Shared room	64.023421	10272	138.572917
9	Queens	Entire home/apt	72.649460	60644	132.267176
10	Queens	Private room	44.911398	93513	149.285163
11	Queens	Shared room	48.742919	2745	192.186869
12	Staten Island	Entire home/apt	94.775155	5857	178.073864
13	Staten Island	Private room	40.769991	5670	226.361702
14	Staten Island	Shared room	27.907407	14	64.777778

In [56]:

```
df['neighbourhood_group'].value_counts()
```

Out[56]:

```
neighbourhood_group
Manhattan        21643
Brooklyn         20089
Queens            5664
Bronx             1089
Staten Island      373
Name: count, dtype: int64
```

In [57]:

```
# pivot table for better visualization
pivot_table = pd.pivot_table(df, values='price_per_night', index='neighbourhood_group',
```

In [58]:

```
pivot_table
```

Out[58]:

room_type	Entire home/apt	Private room	Shared room

neighbourhood_group			
Bronx	68.501386	40.861001	44.643933
Brooklyn	75.408559	41.516116	31.558991
Manhattan	100.299537	66.217700	64.023421
Queens	72.649460	44.911398	48.742919
Staten Island	94.775155	40.769991	27.907407

In [148…

```python
# Calculate total listings count per host
host_listings_count = df.groupby('host_id')['id'].count().reset_index(name='total_listin
```

In [149…

```python
# Calculate average reviews per month per neighbourhood
avg_reviews_per_neighbourhood = df.groupby('neighbourhood')['reviews_per_month'].mean().
```

In [150…

```python
print(host_listings_count)
print(avg_reviews_per_neighbourhood)
```

```
         host_id  total_listings
34615  219517861             327
29379  107434423             232
19557   30283594             121
31050  137358866             103
12796   12243051              96
...          ...             ...
13347   13538150               1
13346   13535952               1
13345   13533446               1
13344   13532838               1
37424  274321313               1

[37425 rows x 2 columns]
                  neighbourhood  avg_reviews_per_month
59                East Elmhurst               4.512486
177                 Silver Lake               4.340000
183          Springfield Gardens               4.235529
170                    Rosebank               3.812857
101                    Huguenot               3.760000
..                          ...                    ...
116                 Little Neck               0.488000
9     Bay Terrace, Staten Island               0.455000
208                 West Farms               0.395000
21                 Breezy Point               0.386667
42                     Co-op City               0.245000

[221 rows x 2 columns]
```

In [69]:

```python
# Group by neighbourhood and room type, and calculate the mean price per night
grouped_df = df.groupby(['neighbourhood', 'room_type']).agg({
    'price_per_night': 'mean',
    'number_of_reviews': 'sum'
})
```

In [70]:

```python
grouped_df
```

Out[70]:

		price_per_night	number_of_reviews
neighbourhood	**room_type**		
Allerton	**Entire home/apt**	62.317708	864
	Private room	45.265110	939
Arden Heights	**Entire home/apt**	30.588889	24

	Private room	20.500000	7
Arrochar	**Entire home/apt**	95.025000	169
...	**...**	...	...
Woodlawn	**Shared room**	35.000000	2
Woodrow	**Entire home/apt**	100.000000	0
Woodside	**Entire home/apt**	90.078345	2649
	Private room	31.911387	2378
	Shared room	32.500000	8

540 rows × 2 columns

In [132…

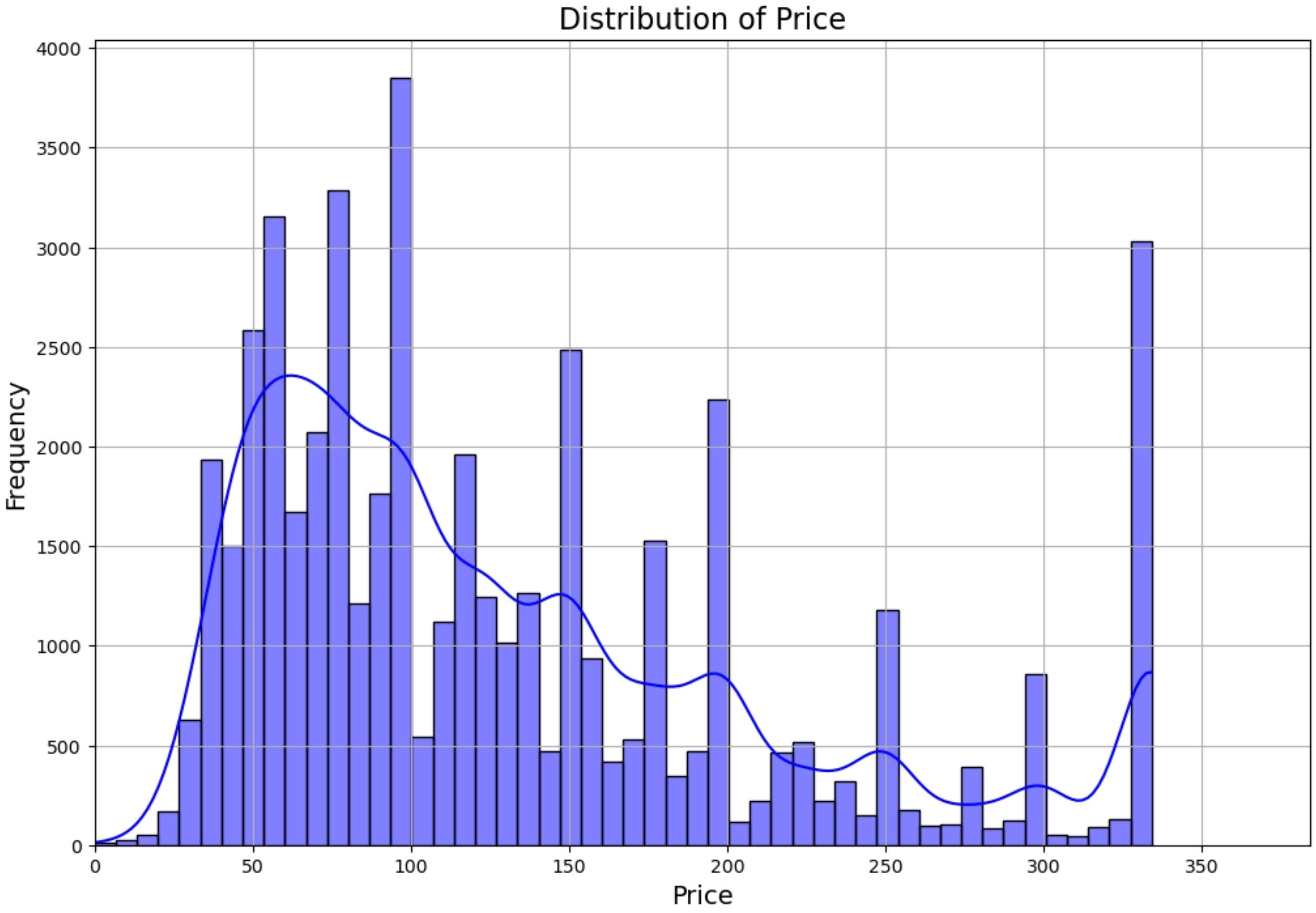
```python
# Calculate Q1 (25th percentile) and Q3 (75th percentile)
Q1 = df['price'].quantile(0.25)
Q3 = df['price'].quantile(0.75)
IQR = Q3 - Q1

# Define the range for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers
outliers_iqr = df[(df['price'] < lower_bound) | (df['price'] > upper_bound)]

print("Outliers based on IQR:")
print(outliers_iqr[['id', 'price']])
```

```
Outliers based on IQR:
Empty DataFrame
Columns: [id, price]
Index: []
```

In [127…

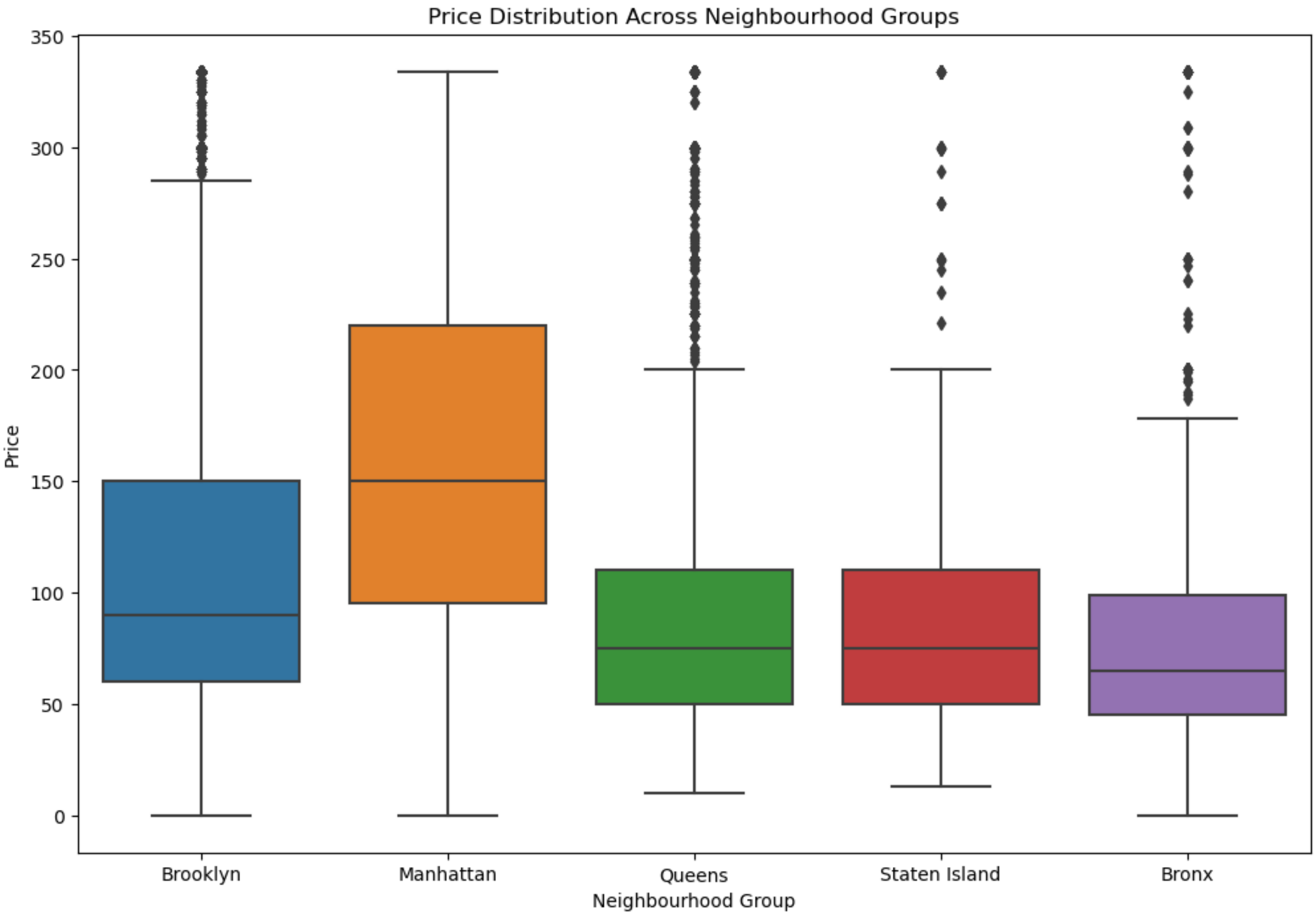
```python
# Distribution of the Target Variable (Price)
plt.figure(figsize=(12, 8))
sns.histplot(df['price'], kde=True, bins=50, color='blue', edgecolor='black')
plt.title('Distribution of Price', fontsize=16)
plt.xlabel('Price', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.xlim(0, df['price'].max() + 50)
plt.grid(True)
plt.show()
```

```
C:\Users\HP\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf
_as_na option is deprecated and will be removed in a future version. Convert inf values
to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



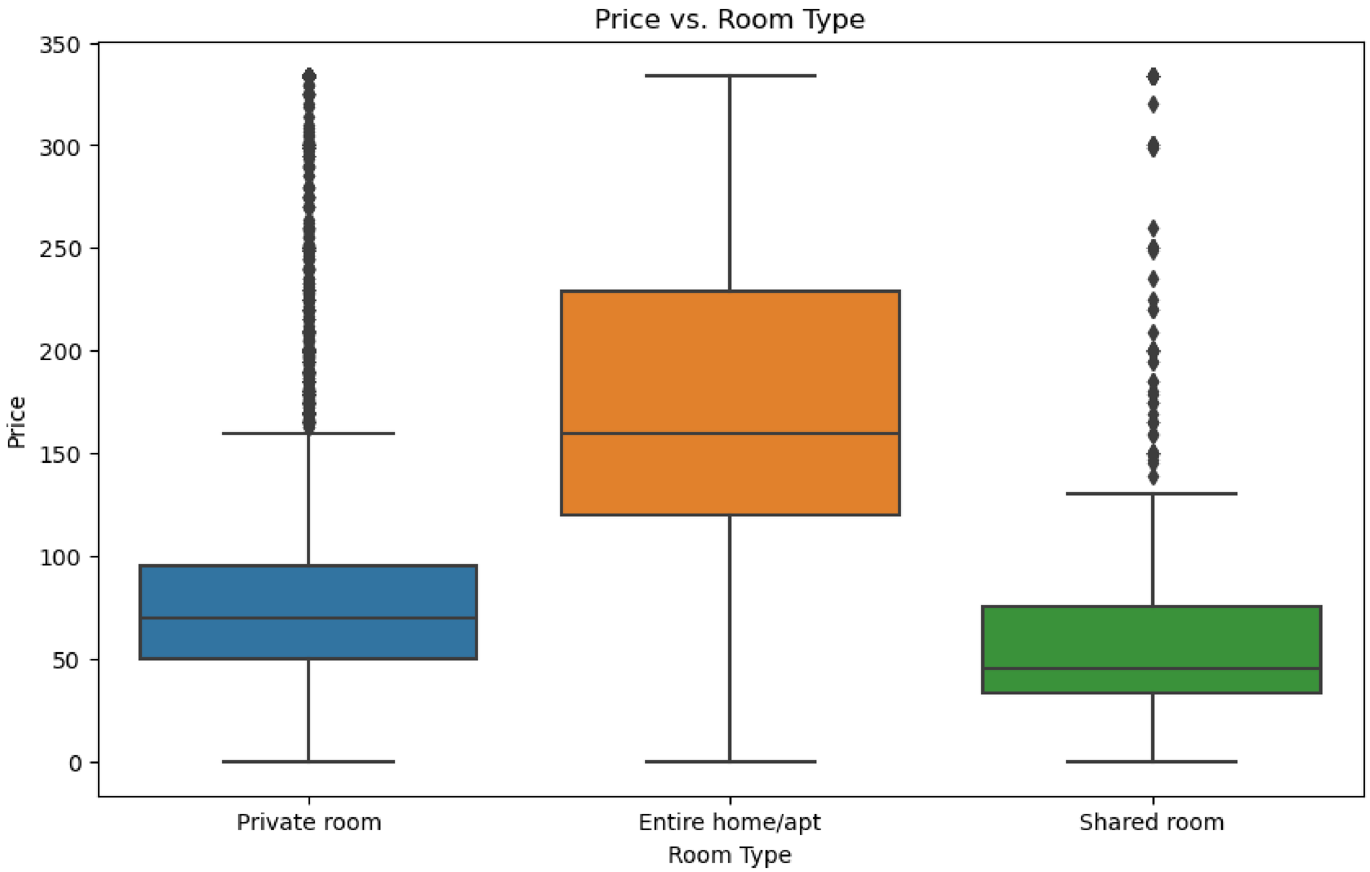
In [128…

```
plt.figure(figsize=(12, 8))
sns.boxplot(x='neighbourhood_group', y='price', data=df)
plt.title('Price Distribution Across Neighbourhood Groups')
plt.xlabel('Neighbourhood Group')
plt.ylabel('Price')
plt.show()
```



In [129…

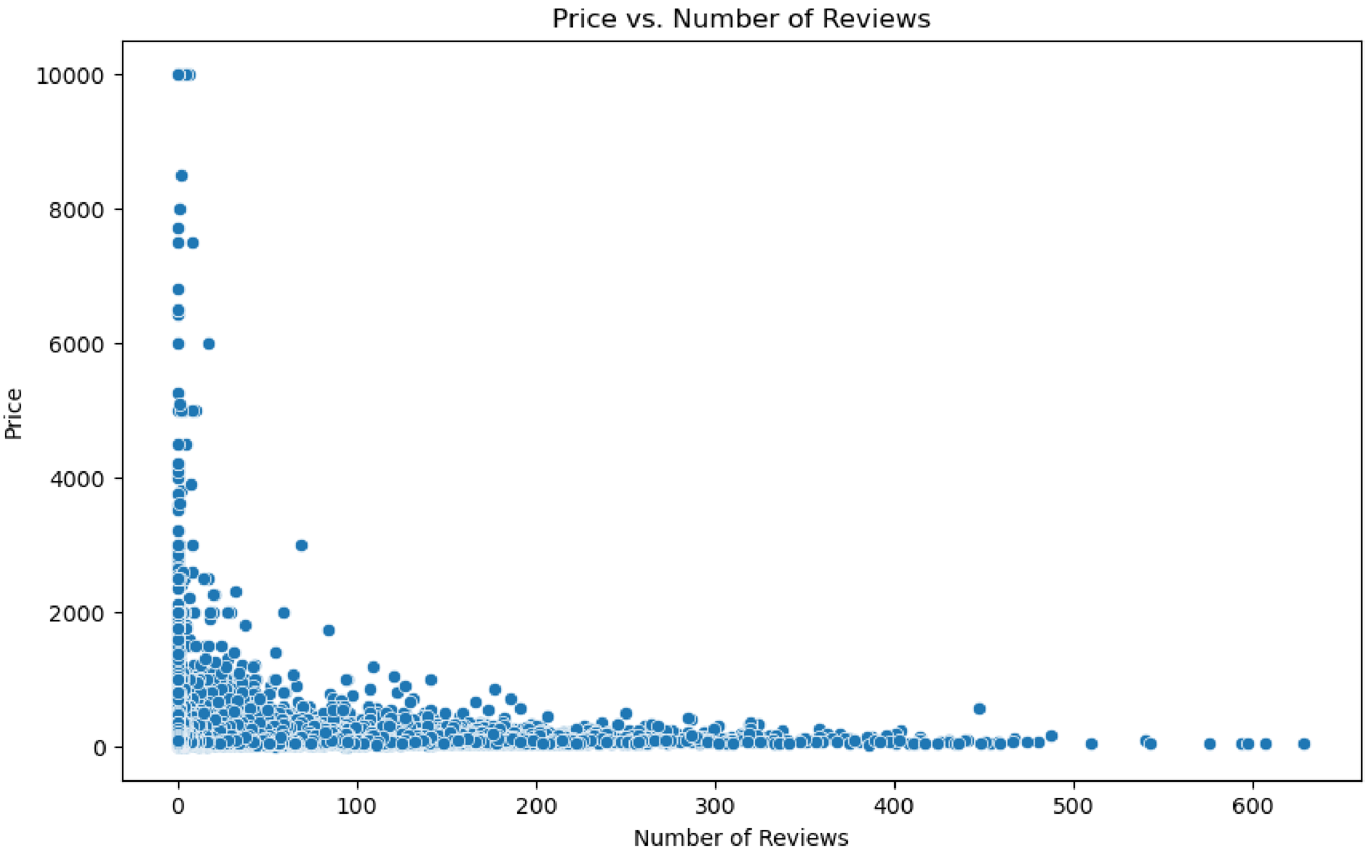
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='room_type', y='price', data=df)
plt.title('Price vs. Room Type')
plt.xlabel('Room Type')
plt.ylabel('Price')
plt.show()
```



```
In [78]: df['price'] = pd.to_numeric(df['price'], errors='coerce')
```

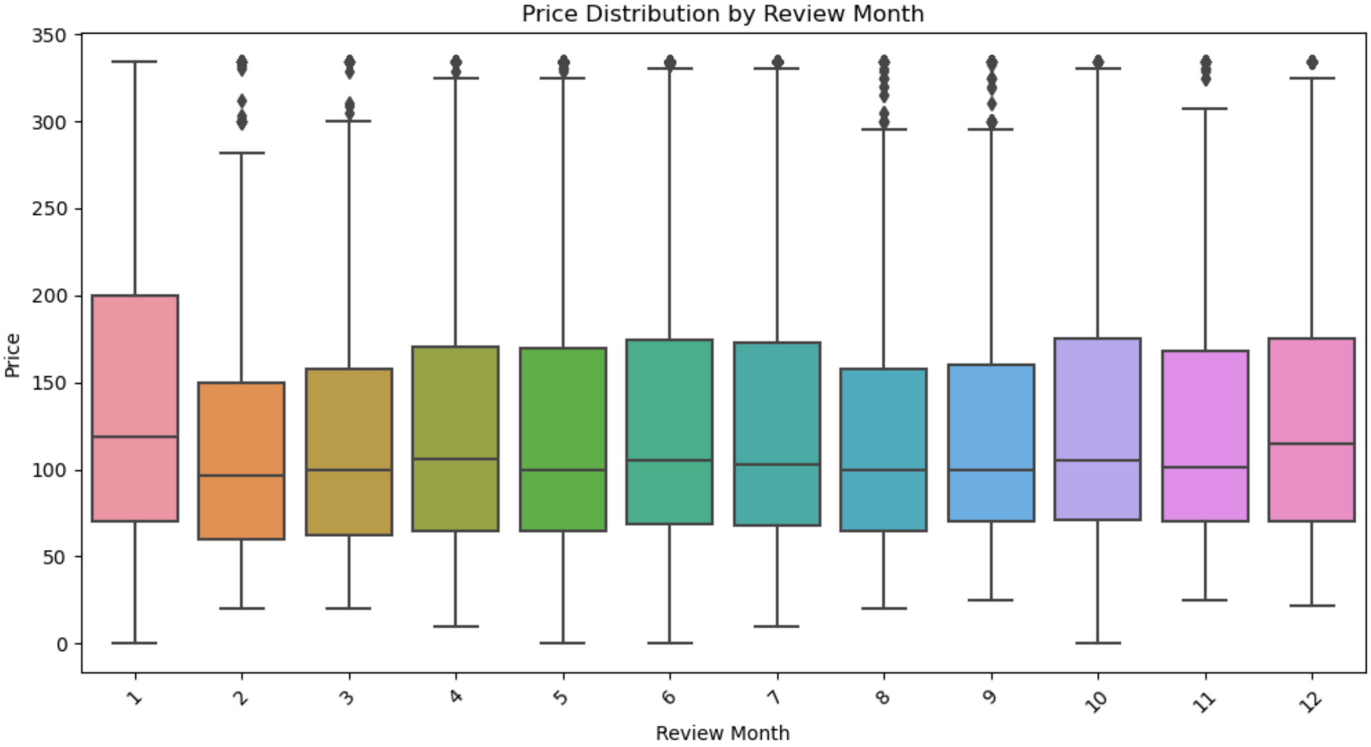
```
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\3828005640.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df['price'] = pd.to_numeric(df['price'], errors='coerce')
```

```
In [84]: plt.figure(figsize=(10, 6))
         sns.scatterplot(x='number_of_reviews', y='price', data=df)
         plt.title('Price vs. Number of Reviews')
         plt.xlabel('Number of Reviews')
         plt.ylabel('Price')
         plt.show()
```



In [140…

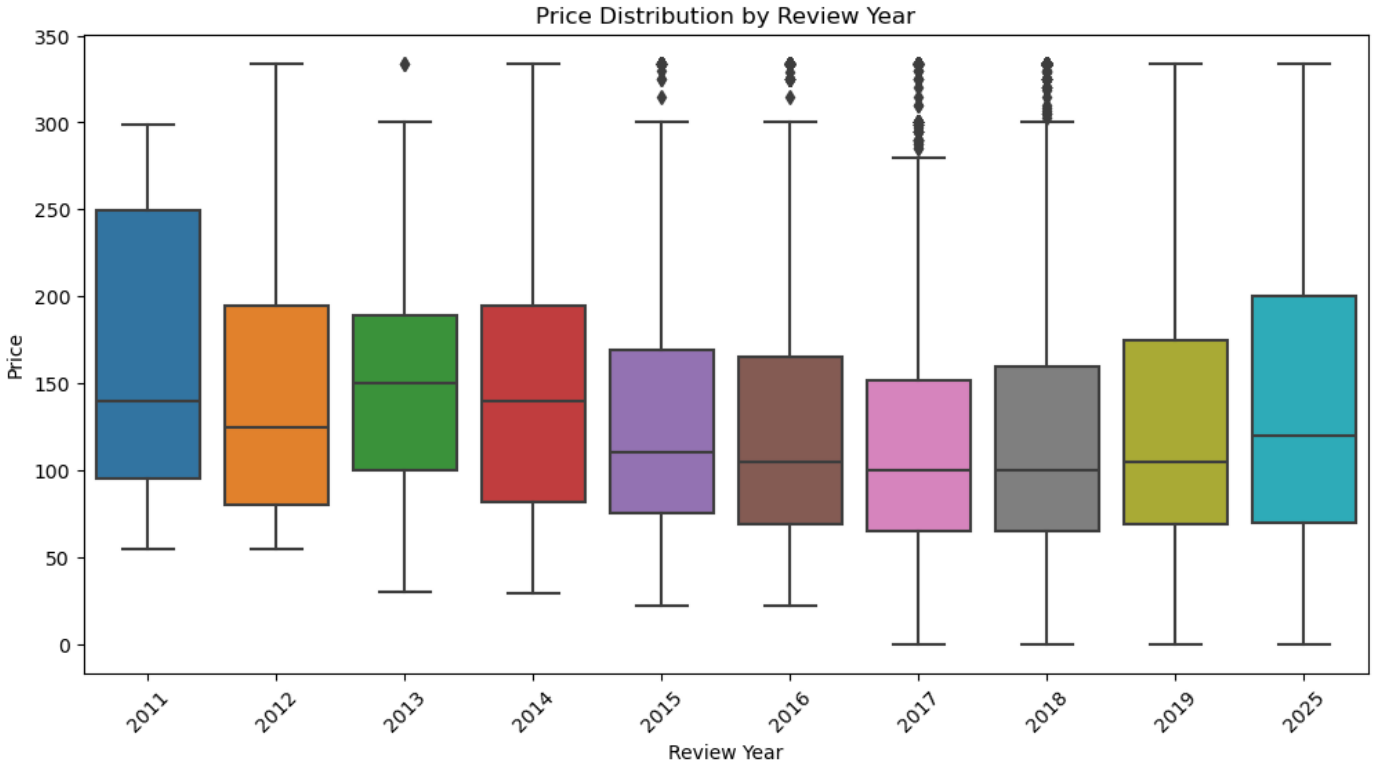
```
# Create a box plot
plt.figure(figsize=(12, 6))
sns.boxplot(x='review_month', y='price', data=df)
plt.title('Price Distribution by Review Month')
plt.xlabel('Review Month')
plt.ylabel('Price')
plt.xticks(rotation=45)  # Rotate x-axis labels for readability
plt.show()
```



In [141…

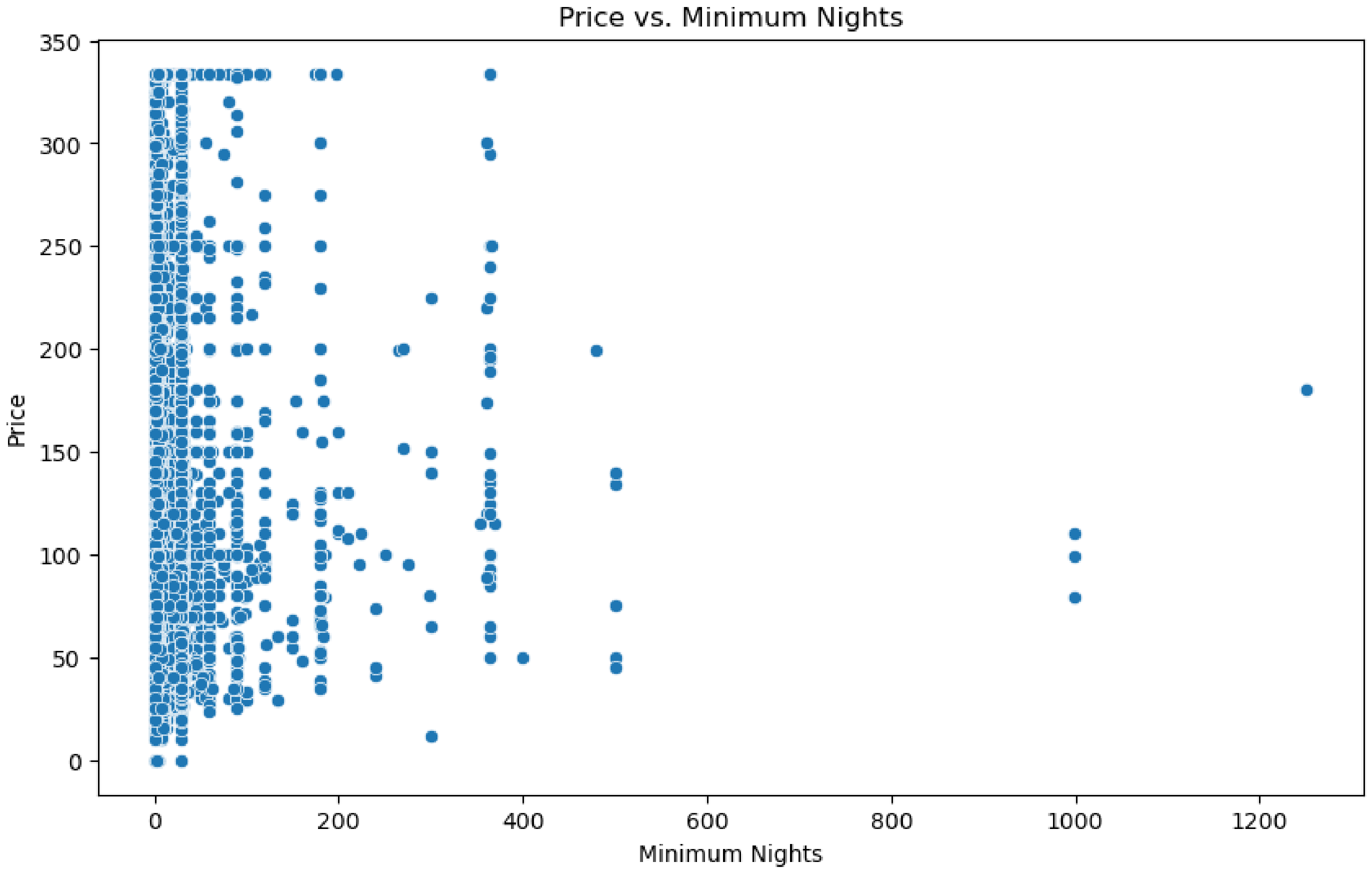
```
# Create a box plot
plt.figure(figsize=(12, 6))
sns.boxplot(x='review_year', y='price', data=df)
```

```python
plt.title('Price Distribution by Review Year')
plt.xlabel('Review Year')
plt.ylabel('Price')
plt.xticks(rotation=45)  # Rotate x-axis labels for readability
plt.show()
```



In [134…

```python
plt.figure(figsize=(10, 6))
sns.scatterplot(x='minimum_nights', y='price', data=df)
plt.title('Price vs. Minimum Nights')
plt.xlabel('Minimum Nights')
plt.ylabel('Price')
plt.show()
```



In [94]:

```
#Basic Statistical Metrics
import numpy as np

# Convert column to NumPy array
prices = df['price'].to_numpy()

# Calculate basic statistics
mean_price = np.mean(prices)
median_price = np.median(prices)
std_dev_price = np.std(prices)

print("Mean price:", mean_price)
print("Median price:", median_price)
print("Standard deviation of price:", std_dev_price)
```

```
Mean price: 152.74030864955586
Median price: 106.0
Standard deviation of price: 240.22992703717497
```

In [95]:

```
#Perform arithmetic operations on prices.
# Increase all prices by 10%
increased_prices = prices * 1.10
# Add the new column to DataFrame
df['price_increased'] = increased_prices
df['price_increased']
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\941692084.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df['price_increased'] = increased_prices
```

Out[95]:

```
0         163.9
1         247.5
2         165.0
3          97.9
```

```
4         88.0
         ...
48890     77.0
48891     44.0
48892    126.5
48893     60.5
48894     99.0
Name: price_increased, Length: 48858, dtype: float64
```

In [97]:
```
df
```

Out[97]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48858 rows × 20 columns

In [98]:
```
#Aggregation Functions
# Calculate sum of prices
total_price = np.sum(prices)
print("Total price:", total_price)

# Calculate cumulative sum
cumulative_sum = np.cumsum(prices)

# Add cumulative sum to DataFrame
df['cumulative_sum'] = cumulative_sum
```

```
Total price: 7462586
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\2672193380.py:10: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df['cumulative_sum'] = cumulative_sum
```

In [99]:

```
df
```

Out[99]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	lon
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73
...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73

48858 rows × 21 columns

In [102…

```
df.dtypes
```

Out[102]:

```
id                                           int64
name                                        object
host_id                                      int64
host_name                                   object
neighbourhood_group                         object
neighbourhood                               object
latitude                                   float64
longitude                                  float64
room_type                                   object
price                                        int64
```

```
 minimum_nights                                int64
 number_of_reviews                             int64
 last_review                          datetime64[ns]
 reviews_per_month                           float64
 calculated_host_listings_count                int64
 availability_365                              int64
 review_year                                   int32
 review_month                                  int32
 price_per_night                             float64
 price_increased                             float64
 cumulative_sum                                int64
 dtype: object
```

In [107…

```python
# Inspect the problematic columns
for col in ['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month', 'calcul
    # Check for non-numeric values
    non_numeric = df[~df[col].apply(lambda x: isinstance(x, (int, float)))]
    if not non_numeric.empty:
        print(f"Non-numeric values in column {col}:\n", non_numeric)
```

In [108…

```python
print(f"Non-numeric values in column {col}:\n", non_numeric)
```

```
Non-numeric values in column availability_365:
 Empty DataFrame
Columns: [id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, lo
ngitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_m
onth, calculated_host_listings_count, availability_365, review_year, review_month, price
_per_night, price_increased, cumulative_sum]
Index: []

[0 rows x 21 columns]
```

In [110…

```python
# Filter listings with prices greater than the median price
median_price = np.median(prices)
high_price_listings = df[prices > median_price]

print(high_price_listings.head())
```

```
     id                                              name  host_id  host_name  \
0  2539                Clean & quiet apt home by the park     2787       John   
1  2595                             Skylit Midtown Castle     2845   Jennifer   
2  3647               THE VILLAGE OF HARLEM....NEW YORK !     4632  Elisabeth   
5  5099  Large Cozy 1 BR Apartment In Midtown East     7322      Chris   
9  5238                Cute & Cozy Lower East Side 1 bdrm     7549        Ben   

  neighbourhood_group neighbourhood  latitude  longitude        room_type  \
0            Brooklyn    Kensington  40.64749  -73.97237     Private room   
1           Manhattan       Midtown  40.75362  -73.98377  Entire home/apt   
2           Manhattan        Harlem  40.80902  -73.94190     Private room   
5           Manhattan   Murray Hill  40.74767  -73.97500  Entire home/apt   
9           Manhattan     Chinatown  40.71344  -73.99037  Entire home/apt   

   price  ...  number_of_reviews  last_review reviews_per_month  \
0    149  ...                  9   2018-10-19              0.21   
1    225  ...                 45   2019-05-21              0.38   
2    150  ...                  0   2025-01-01              0.72   
5    200  ...                 74   2019-06-22              0.59   
9    150  ...                160   2019-06-09              1.33   

   calculated_host_listings_count  availability_365  review_year  \
0                               6               365         2018   
1                               2               355         2019   
2                               1               365         2025   
5                               1               129         2019   
9                               4               188         2019   
```

```
   review_month  price_per_night  price_increased  cumulative_sum
0            10       149.000000            163.9             149
1             5       225.000000            247.5             374
2             1        50.000000            165.0             524
5             6        66.666667            220.0             893
9             6       150.000000            165.0            1261

[5 rows x 21 columns]
```

In [112…

```python
import numpy as np

# Convert relevant columns to NumPy arrays
prices = df['price'].to_numpy()
minimum_nights = df['minimum_nights'].to_numpy()
number_of_reviews = df['number_of_reviews'].to_numpy()
reviews_per_month = df['reviews_per_month'].to_numpy()
calculated_host_listings_count = df['calculated_host_listings_count'].to_numpy()
availability_365 = df['availability_365'].to_numpy()

# Calculate basic statistics
mean_price = np.mean(prices)
median_price = np.median(prices)
std_dev_price = np.std(prices)

print("Mean price:", mean_price)
print("Median price:", median_price)
print("Standard deviation of price:", std_dev_price)
```

```
Mean price: 152.74030864955586
Median price: 106.0
Standard deviation of price: 240.22992703717497
```

In [113…

```python
# Calculate total reviews and average reviews per month
total_reviews = np.sum(number_of_reviews)
average_reviews_per_month = np.nanmean(reviews_per_month)

print("Total reviews:", total_reviews)
print("Average reviews per month:", average_reviews_per_month)
```

```
Total reviews: 1137077
Average reviews per month: 1.2390349584510212
```

In [115…

```python
import pandas as pd
import numpy as np


# Ensure the necessary columns are numeric
columns_to_convert = ['price', 'review_year', 'review_month']
for col in columns_to_convert:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Drop rows with NaN values in the necessary columns
df.dropna(subset=columns_to_convert, inplace=True)

# Convert necessary columns to NumPy arrays
prices = df['price'].to_numpy()
review_year = df['review_year'].to_numpy()
review_month = df['review_month'].to_numpy()

# Create a unique identifier for each month
months = review_year * 100 + review_month

# Get unique months
unique_months = np.unique(months)

# Initialize an array to hold the total price for each month
```

```
monthly_total_price = np.zeros(unique_months.shape)

# Calculate total price for each unique month
for i, month in enumerate(unique_months):
    monthly_total_price[i] = np.sum(prices[months == month])

# Create a DataFrame to display the results
monthly_totals_df = pd.DataFrame({
    'YearMonth': unique_months,
    'TotalPrice': monthly_total_price
}).sort_values(by="TotalPrice",ascending=False)

# Display the results
print(monthly_totals_df)
```

```
    YearMonth  TotalPrice
92     202501   1937095.0
90     201906   1784286.0
91     201907    682360.0
89     201905    515743.0
88     201904    208516.0
..        ...         ...
20     201308       175.0
26     201402        95.0
27     201403        90.0
6      201205        65.0
0      201103        55.0

[93 rows x 2 columns]
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\2500310539.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col], errors='coerce')
C:\Users\HP\AppData\Local\Temp\ipykernel_13192\2500310539.py:11: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df.dropna(subset=columns_to_convert, inplace=True)
```

Using NumPy for these operations ensures computational efficiency and can significantly speed up data manipulation tasks compared to traditional loops or list comprehensions. By leveraging NumPy’s array operations, wecan handle large datasets more effectively and perform complex calculations with ease.