AirBnB Data Analysis

Data Fetching & Loading

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv('AirBnB.csv')
```

```
C:\Users\SAGAR\AppData\Local\Temp\ipykernel_9712\3083142198.py:1: DtypeWarning:
Columns (25) have mixed types. Specify dtype option on import or set low_memor
y=False.
  df=pd.read_csv('AirBnB.csv')
```

In [4]:
```
df.head(5)
```

Out[4]:

	id	NAME	host id	host_identity_verified	host name	neighb
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	M
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	M
3	1002755	NaN	85098326012	unconfirmed	Garry	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	M

5 rows × 26 columns

Check the column names in the Dataset

In [5]:
```
df.columns
```

Out[5]:
```
Index(['id', 'NAME', 'host id', 'host_identity_verified', 'host name',
       'neighbourhood group', 'neighbourhood', 'lat', 'long', 'country',
       'country code', 'instant_bookable', 'cancellation_policy', 'room typ
e',
       'Construction year', 'price', 'service fee', 'minimum nights',
       'number of reviews', 'last review', 'reviews per month',
       'review rate number', 'calculated host listings count',
       'availability 365', 'house_rules', 'license'],
      dtype='object')
```

Data Cleaning & Preprocessing Check for Missing Values

In [7]:
```
print(df.isnull().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
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   id                              102599 non-null  int64  
 1   NAME                            102349 non-null  object 
 2   host id                         102599 non-null  int64  
 3   host_identity_verified          102310 non-null  object 
 4   host name                       102193 non-null  object 
 5   neighbourhood group             102570 non-null  object 
 6   neighbourhood                   102583 non-null  object 
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object 
 10  country code                    102468 non-null  object 
 11  instant_bookable                102494 non-null  object 
 12  cancellation_policy             102523 non-null  object 
 13  room type                       102599 non-null  object 
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object 
 16  service fee                     102326 non-null  object 
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   object 
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object 
 25  license                         2 non-null       object 
dtypes: float64(9), int64(2), object(15)
memory usage: 20.4+ MB
```

Handle the Missing Values

In [11]:
```
# To check any particular field top 10 values
df['last review'].head(10)
```

Out[11]:
```
0    10/19/2021
1     5/21/2022
2           NaN
3     7/5/2019
4    11/19/2018
5     6/22/2019
6     10/5/2017
7     10/5/2017
8     6/24/2019
9     7/21/2017
Name: last review, dtype: object
```

In [13]:
```
# Convert 'last review' to datetime and handle errors, errors='coerce' it mean
df['last review']=pd.to_datetime(df['last review'],errors='coerce')
```

```
#Fill missing values
df.fillna({'reviews per month':0, 'last review':df['last review'].min()}, inpl

# Drop records with missing 'name' or 'host name'
df.dropna(subset=['NAME', 'host name'], inplace=True)
```

In [15]:
```
df.head(20)
```

Out[15]:

	id	NAME	host id	host_identity_verified	host name	neigh
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	
5	1004098	Large Cozy 1 BR Apartment In Midtown East	45498551794	verified	Michelle	
6	1004650	BlissArtsSpace!	61300605564	NaN	Alberta	
7	1005202	BlissArtsSpace!	90821839709	unconfirmed	Emma	
8	1005754	Large Furnished Room Near B'way	79384379533	verified	Evelyn	
9	1006307	Cozy Clean Guest Room - Family Apt	75527839483	unconfirmed	Carl	
10	1006859	Cute & Cozy Lower East	1280143094	verified	Miranda	

	id	NAME	host id	host_identity_verified	host name	neigh
		Side 1 bdrm				
11	1007411	Beautiful 1br on Upper West Side	18824631834	verified	Alan	
13	1008516	Lovely Room 1, Garden, Best Area, Legal rental	26802410424	verified	Darcy	
14	1009068	Wonderful Guest Bedroom in Manhattan for SINGLES	88920244552	verified	Leonardo	
15	1009621	West Village Nest - Superhost	46551725984	verified	Daniel	
16	1010173	Only 2 stops to Manhattan studio	62566345680	unconfirmed	Heather	
17	1010725	Perfect for Your Parents + Garden	80380130347	verified	Ryan	
18	1011277	Chelsea Perfect	73862528370	verified	Alberta	
19	1011830	Hip Historic Brownstone Apartment with Backyard	72145018858	NaN	Martin	
20	1012382	Huge 2 BR Upper East Cental Park	79805143117	verified	Audrey	
21	1012934	Sweet and Spacious Brooklyn Loft	86554611512	verified	Alissa	

20 rows × 26 columns

In [16]:
```
print(df.isnull().sum())
```

```
id                                    0
NAME                                  0
host id                               0
host_identity_verified              276
host name                             0
neighbourhood group                  26
neighbourhood                        16
lat                                   8
long                                  8
country                             526
country code                        122
instant_bookable                     96
cancellation_policy                  70
room type                             0
Construction year                   200
price                               239
service fee                         268
minimum nights                      403
number of reviews                   182
last review                           0
reviews per month                     0
review rate number                  314
calculated host listings count      318
availability 365                    420
house_rules                       51867
license                          101947
dtype: int64
```

To remove unnecessary columns like house rules and ignore which are not required for analysis

In [17]:
```
df = df.drop(columns=['house_rules','license'], errors='ignore')
```

In [18]:
```
df.columns
```

Out[18]:
```
Index(['id', 'NAME', 'host id', 'host_identity_verified', 'host name',
       'neighbourhood group', 'neighbourhood', 'lat', 'long', 'country',
       'country code', 'instant_bookable', 'cancellation_policy', 'room typ
e',
       'Construction year', 'price', 'service fee', 'minimum nights',
       'number of reviews', 'last review', 'reviews per month',
       'review rate number', 'calculated host listings count',
       'availability 365'],
      dtype='object')
```

Here price and service fee are included with $ sign, so we remove that sign and change its dtype

Correct Data Types

In [19]:
```
df['price'] =df['price'].replace('[\$,]','',regex=True).astype(float)
df['service fee'] =df['service fee'].replace('[\$,]','',regex=True).astype(flo
```

```
<>:1: SyntaxWarning: invalid escape sequence '\$'
<>:2: SyntaxWarning: invalid escape sequence '\$'
<>:1: SyntaxWarning: invalid escape sequence '\$'
<>:2: SyntaxWarning: invalid escape sequence '\$'
C:\Users\SAGAR\AppData\Local\Temp\ipykernel_9712\836703720.py:1: SyntaxWarning:
invalid escape sequence '\$'
  df['price'] =df['price'].replace('[\$,]','',regex=True).astype(float)
C:\Users\SAGAR\AppData\Local\Temp\ipykernel_9712\836703720.py:2: SyntaxWarning:
invalid escape sequence '\$'
  df['service fee'] =df['service fee'].replace('[\$,]','',regex=True).astype(fl
oat)
```

In [20]:
```
df['price'].head(5)
```

Out[20]:
```
0    966.0
1    142.0
2    620.0
4    204.0
5    577.0
Name: price, dtype: float64
```

In [21]:
```
df['service fee'].head(5)
```

Out[21]:
```
0    193.0
1     28.0
2    124.0
4     41.0
5    115.0
Name: service fee, dtype: float64
```

Remove Duplicates

In [22]:
```
df.drop_duplicates(inplace=True)
```

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 101410 entries, 0 to 102057
Data columns (total 24 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              101410 non-null  int64
 1   NAME                            101410 non-null  object
 2   host id                         101410 non-null  int64
 3   host_identity_verified          101134 non-null  object
 4   host name                       101410 non-null  object
 5   neighbourhood group             101384 non-null  object
 6   neighbourhood                   101394 non-null  object
 7   lat                             101402 non-null  float64
 8   long                            101402 non-null  float64
 9   country                         100884 non-null  object
 10  country code                    101288 non-null  object
 11  instant_bookable                101314 non-null  object
 12  cancellation_policy             101340 non-null  object
 13  room type                       101410 non-null  object
 14  Construction year               101210 non-null  float64
 15  price                           101171 non-null  float64
 16  service fee                     101142 non-null  float64
 17  minimum nights                  101016 non-null  float64
 18  number of reviews               101228 non-null  float64
 19  last review                     101410 non-null  datetime64[ns]
 20  reviews per month               101410 non-null  float64
 21  review rate number              101103 non-null  float64
 22  calculated host listings count  101092 non-null  float64
 23  availability 365                100990 non-null  float64
dtypes: datetime64[ns](1), float64(11), int64(2), object(10)
memory usage: 19.3+ MB
```

Descriptive Statistics

In [25]:
```
df.describe()
```

Out[25]:

	id	host id	lat	long	Constructior yea
count	1.014100e+05	1.014100e+05	101402.000000	101402.000000	101210.00000
mean	2.920959e+07	4.926155e+10	40.728082	-73.949663	2012.48690
min	1.001254e+06	1.236005e+08	40.499790	-74.249840	2003.00000
25%	1.507574e+07	2.459183e+10	40.688730	-73.982570	2007.00000
50%	2.922911e+07	4.912069e+10	40.722300	-73.954440	2012.00000
75%	4.328308e+07	7.399747e+10	40.762750	-73.932340	2017.00000
max	5.736742e+07	9.876313e+10	40.916970	-73.705220	2022.00000
std	1.626820e+07	2.853703e+10	0.055850	0.049474	5.76513

Actual Problem statements

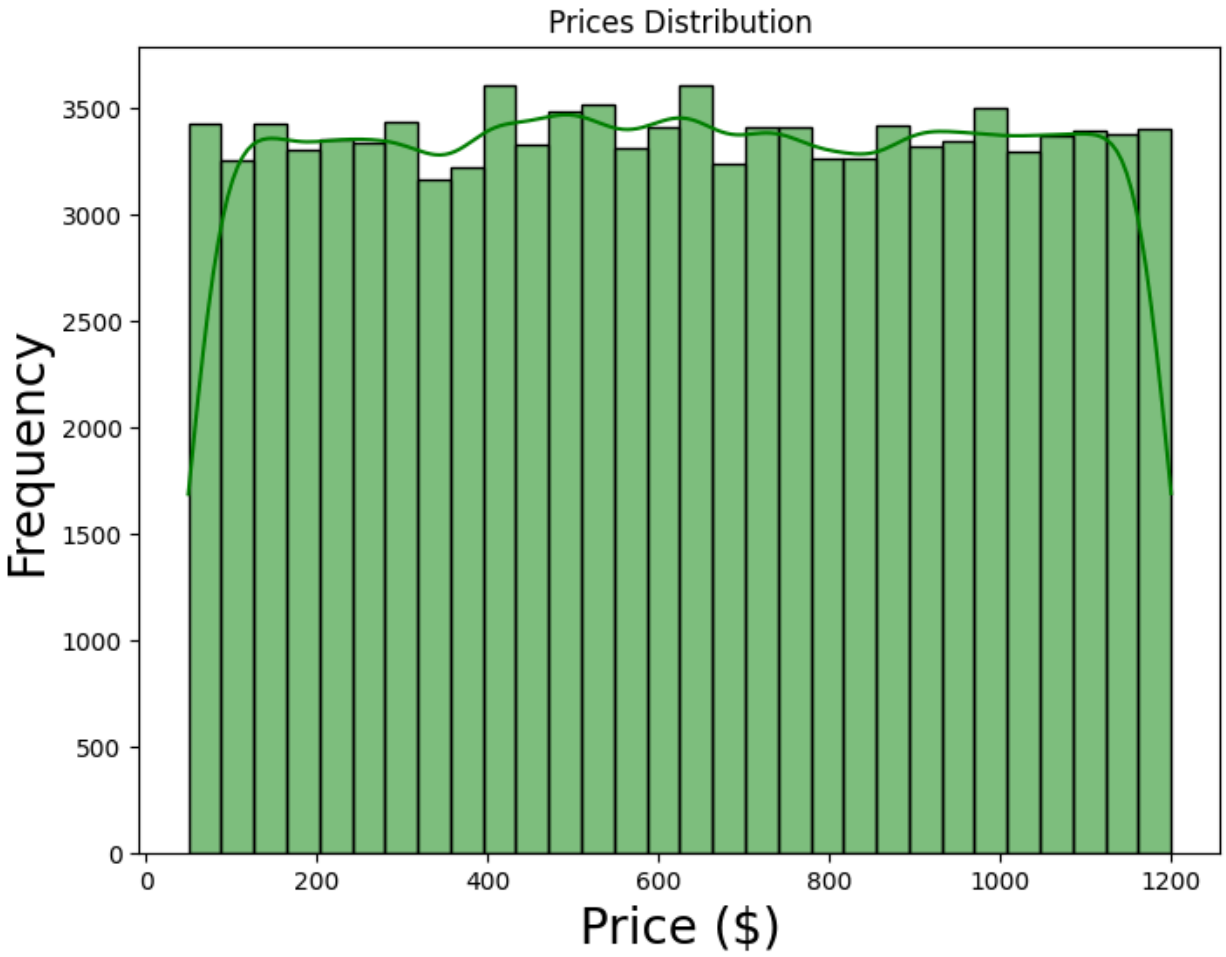
1. Distribution of Prices

In [57]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,6))
sns.histplot(df['price'], bins=30, kde=True, color='green')
plt.title('Prices Distribution')
plt.xlabel('Price ($)',size=20)
plt.ylabel('Frequency',size=20)
plt.show()

#kde is for curve variation
```



Insight Generation: Even distribution of price range, there is no particular spike observed for particular price

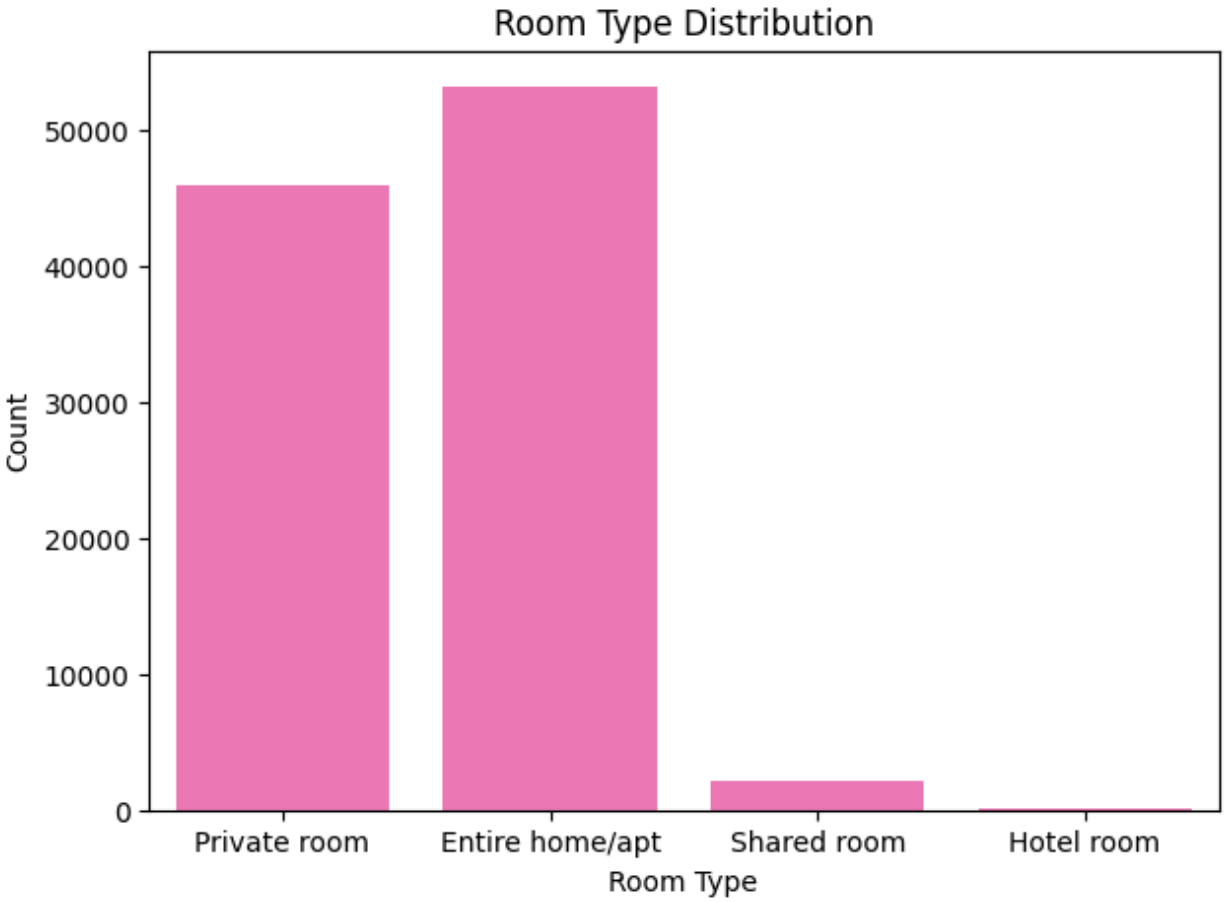
2. How different room types are distributed

In [32]:
```
df['room type'].head(10)
```

Out[32]:
```
0        Private room
1     Entire home/apt
2        Private room
4     Entire home/apt
5     Entire home/apt
6        Private room
7        Private room
8        Private room
9        Private room
10    Entire home/apt
Name: room type, dtype: object
```

In [42]:
```
plt.figure(figsize=(7,5))
sns.countplot(x='room type', data=df, color='hotpink')
plt.title('Room Type Distribution')
```

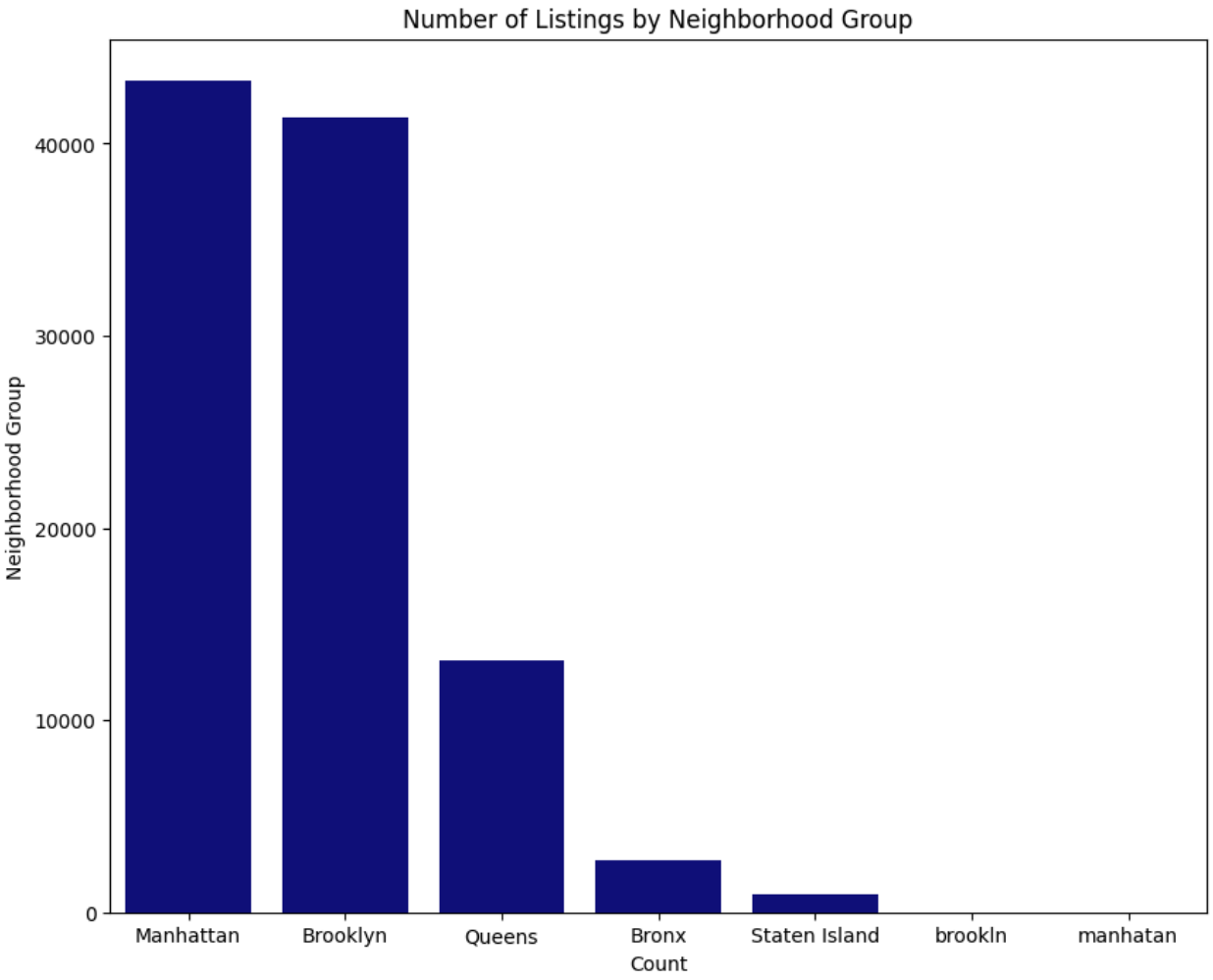
```
plt.xlabel('Room Type',size=10)
plt.ylabel('Count',size=10)
plt.show()
```



Insight Generation: Entire home/Apt shows the max count/most preferred booking type, second one is the private room.

3.Examine how listings are distributed across different neighborhoods.

In [48]:
```
plt.figure(figsize=(10, 8))
sns.countplot(x='neighbourhood group', data=df,color="darkblue" , order=df['ne
plt.title('Number of Listings by Neighborhood Group')
plt.xlabel('Count')
plt.ylabel('Neighborhood Group')
plt.show()
```

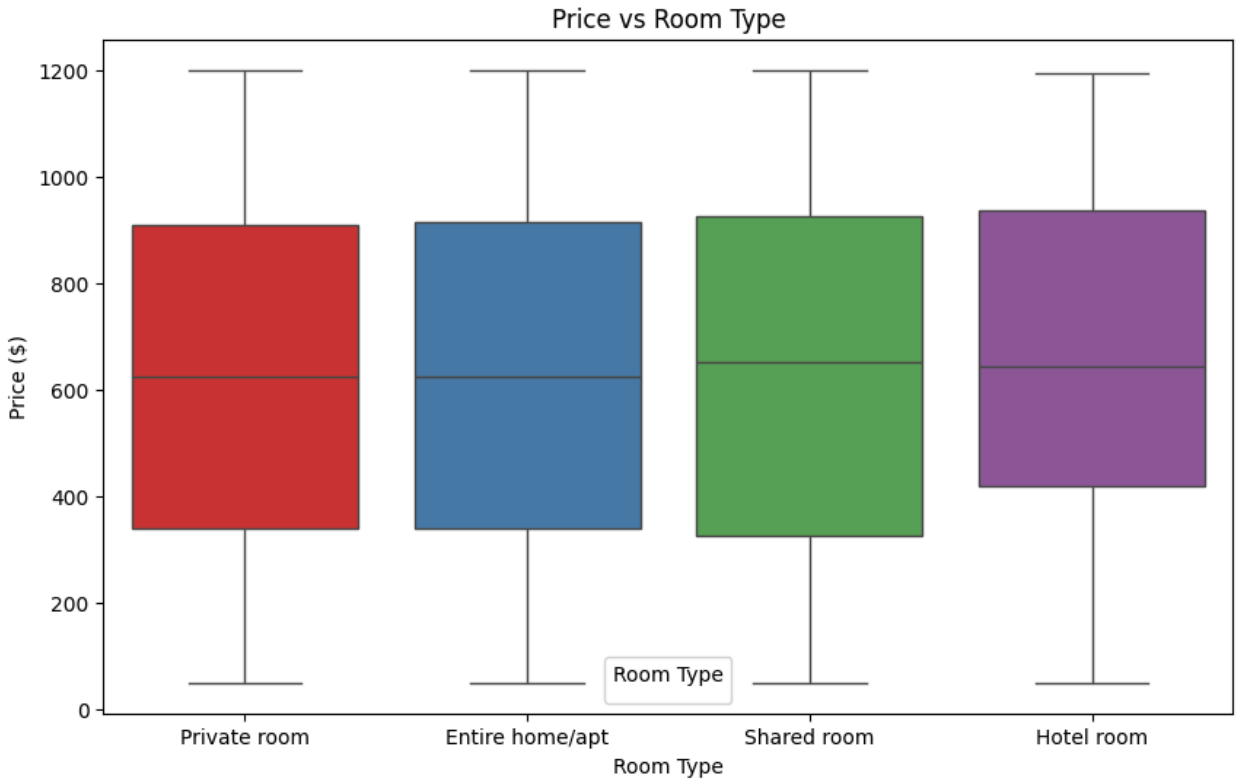


Insight Generation: Most popular Neighbourhood groups are Manhattan, Brooklyn, Queens

4. Visualize the relationship between price and room type

In [53]:
```
plt.figure(figsize=(10,6))
sns.boxplot(x='room type', y='price',data=df, hue='room type',palette='Set1')
plt.title('Price vs Room Type')
plt.xlabel('Room Type',size=10)
plt.ylabel('Price ($)',size=10)
plt.legend(title='Room Type')
plt.show()
```

```
C:\Users\SAGAR\AppData\Local\Temp\ipykernel_9712\3748806386.py:6: UserWarning:
No artists with labels found to put in legend.  Note that artists whose label s
tart with an underscore are ignored when legend() is called with no argument.
  plt.legend(title='Room Type')
```

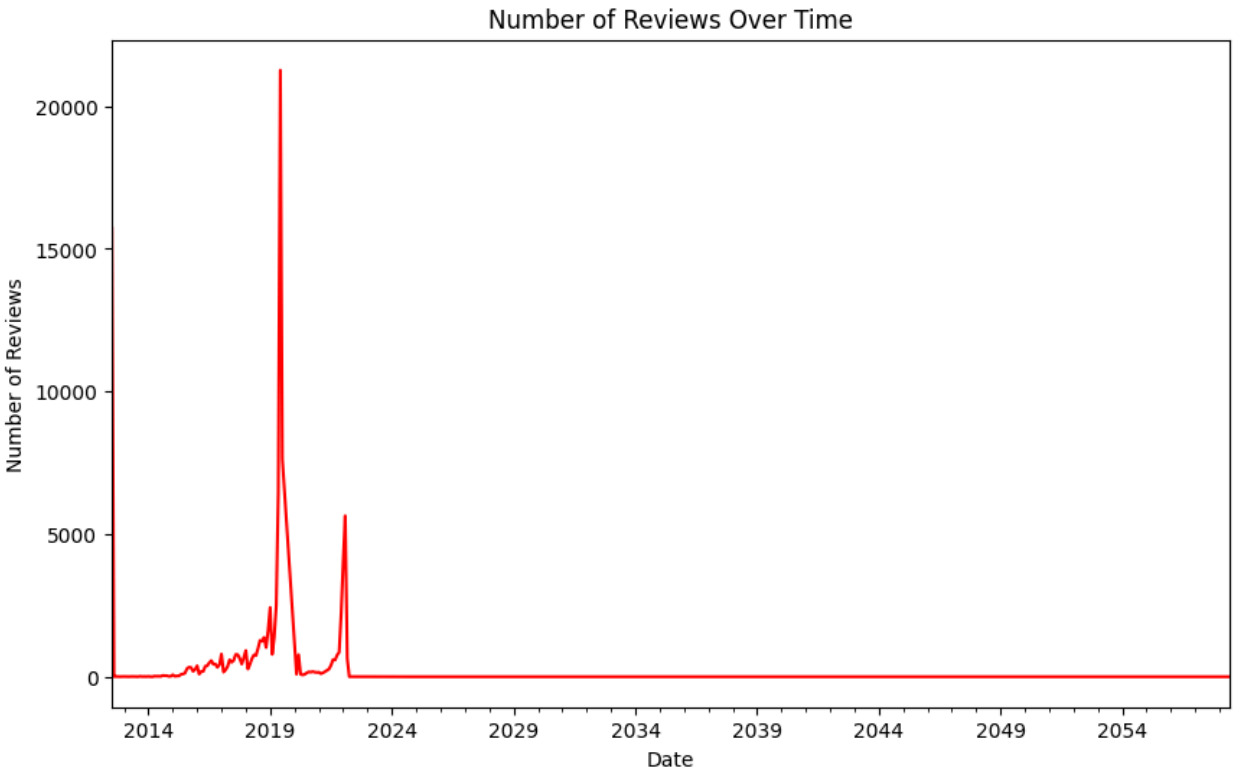


Insight Generation: Price vs. Room Type The box plot provides a detailed view of how prices vary across different room types in the Airbnb dataset. It shows that while 'Shared room' tends to have lower prices, 'Private room', 'Entire home/apt', and 'Hotel room' have higher and more varied price ranges. This visualization helps in understanding the pricing dynamics for different types of accommodations on Airbnb.

5. Reviews over Time

In [55]:
```
df['last review'] =pd.to_datetime(df['last review'])
reviews_over_time=df.groupby(df['last review'].dt.to_period('M')).size()

plt.figure(figsize=(10,6))
reviews_over_time.plot(kind='line',color='red')
plt.title('Number of Reviews Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Reviews')
plt.show()
```



Insight Generation: We can see the number of reviews during 2019 was tremendously high, and all time it was good between 2019 to near 2024.

In [ ]: