Air BNB Data Analysis Project

In [ ]:

In [21]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [23]:
```
df = pd.read_csv("Airbnb_Open_Data.csv")
```

```
C:\Users\swati\AppData\Local\Temp\ipykernel_1952\3424017332.py:1: DtypeWarning: C
olumns (25) have mixed types. Specify dtype option on import or set low_memory=Fa
lse.
  df = pd.read_csv("Airbnb_Open_Data.csv")
```

In [24]:
```
df
```

Out[24]:

	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	
3	1002755	NaN	85098326012	unconfirmed	Garry	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	
...	...	...	...	...	...	
102594	6092437	Spare room in Williamsburg	12312296767	verified	Krik	
102595	6092990	Best Location near Columbia U	77864383453	unconfirmed	Mifan	
102596	6093542	Comfy, bright room in Brooklyn	69050334417	unconfirmed	Megan	
102597	6094094	Big Studio-One Stop from Midtown	11160591270	unconfirmed	Christopher	
102598	6094647	585 sf Luxury Studio	68170633372	unconfirmed	Rebecca	

102599 rows × 26 columns

Check the column names in the Dataset

In [27]:
```
df.columns
```

Out[27]:
```
Index(['id', 'NAME', 'host id', 'host_identity_verified', 'host name',
       'neighbourhood group', 'neighbourhood', 'lat', 'long', 'country',
       'country code', 'instant_bookable', 'cancellation_policy', 'room type',
       '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')
```

Check for Missing Values

In [31]:
```
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
```

Handle Missing Values

In [ ]:
```
This code ensures that the 'last review' column is properly
formatted as datetime, missing values in key columns
are appropriately handled, and incomplete records are removed,
preparing the dataset for further analysis or visualization.
```

In [34]:
```
# Convert 'last review' to datetime and handle errors
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 [36]:
```
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
```

Correct Data Types

In [ ]:
```
Ensure that all columns have the correct data types.
```

In [39]:
```
# Remove dollar signs and convert to float
df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)
df['service fee'] = df['service fee'].replace('[\$,]', '', regex=True).astype(fl
```

```
<>:2: SyntaxWarning: invalid escape sequence '\$'
<>:3: SyntaxWarning: invalid escape sequence '\$'
<>:2: SyntaxWarning: invalid escape sequence '\$'
<>:3: SyntaxWarning: invalid escape sequence '\$'
C:\Users\swati\AppData\Local\Temp\ipykernel_1952\136629131.py:2: SyntaxWarning: i
nvalid escape sequence '\$'
  df['price'] = df['price'].replace('[\$,]', '', regex=True).astype(float)
C:\Users\swati\AppData\Local\Temp\ipykernel_1952\136629131.py:3: SyntaxWarning: i
nvalid escape sequence '\$'
  df['service fee'] = df['service fee'].replace('[\$,]', '', regex=True).astype(f
loat)
```

Remove Duplicates

In [42]:
```
df.drop_duplicates(inplace=True)
```

Confirm Data Cleaning

In [45]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 101410 entries, 0 to 102057
Data columns (total 26 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
 24  house_rules                     49831 non-null   object
 25  license                         2 non-null       object
dtypes: datetime64[ns](1), float64(11), int64(2), object(12)
memory usage: 20.9+ MB
None
```

In [47]:
```
df = df.drop(columns=["license", "house_rules"], errors='ignore')  # Permanently
```

In [49]:
```
df
```

Out[49]:

	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
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	M
5	1004098	Large Cozy 1 BR Apartment In Midtown East	45498551794	verified	Michelle	M
...	...	...	...	...	...	
102053	57365208	Cozy bright room near Prospect Park	77326652202	unconfirmed	Mariam	
102054	57365760	Private Bedroom with Amazing Rooftop View	45936254757	verified	Trey	
102055	57366313	Pretty Brooklyn One-Bedroom for 2 to 4 people	23801060917	verified	Michael	
102056	57366865	Room & private bathroom in historic Harlem	15593031571	unconfirmed	Shireen	M
102057	57367417	Rosalee Stewart	93578954226	verified	Stanley	M

101410 rows × 24 columns

Descriptive Statistics

In [54]:
```
df.describe()
```

Out[54]:

	id	host id	lat	long	Construction year	
count	1.014100e+05	1.014100e+05	101402.000000	101402.000000	101210.000000	10117
mean	2.920959e+07	4.926155e+10	40.728082	-73.949663	2012.486908	62
min	1.001254e+06	1.236005e+08	40.499790	-74.249840	2003.000000	5
25%	1.507574e+07	2.459183e+10	40.688730	-73.982570	2007.000000	34
50%	2.922911e+07	4.912069e+10	40.722300	-73.954440	2012.000000	62
75%	4.328308e+07	7.399747e+10	40.762750	-73.932340	2017.000000	91
max	5.736742e+07	9.876313e+10	40.916970	-73.705220	2022.000000	120
std	1.626820e+07	2.853703e+10	0.055850	0.049474	5.765130	33

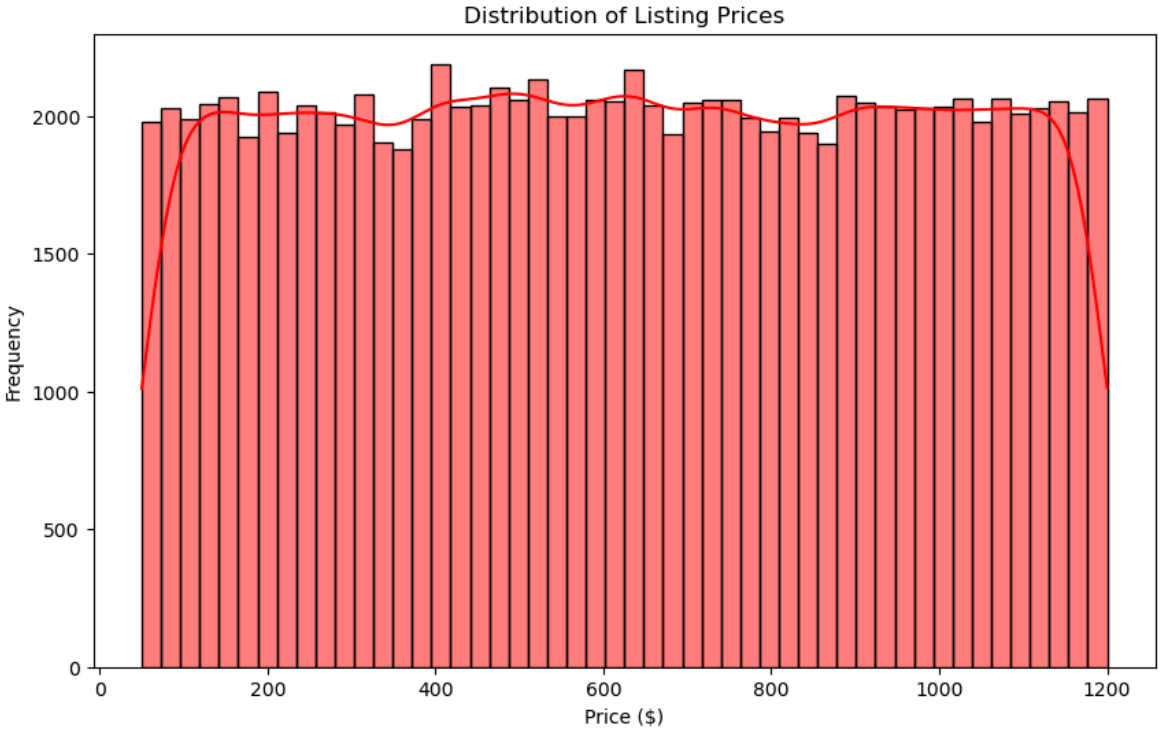
Visualization

Distribution of Prices

Plot the distribution of listing prices.

In [62]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.histplot(df['price'], bins=50, kde=True, color='red')  # Set histogram color
plt.title('Distribution of Listing Prices')
plt.xlabel('Price ($)')
plt.ylabel('Frequency')
plt.show()
```

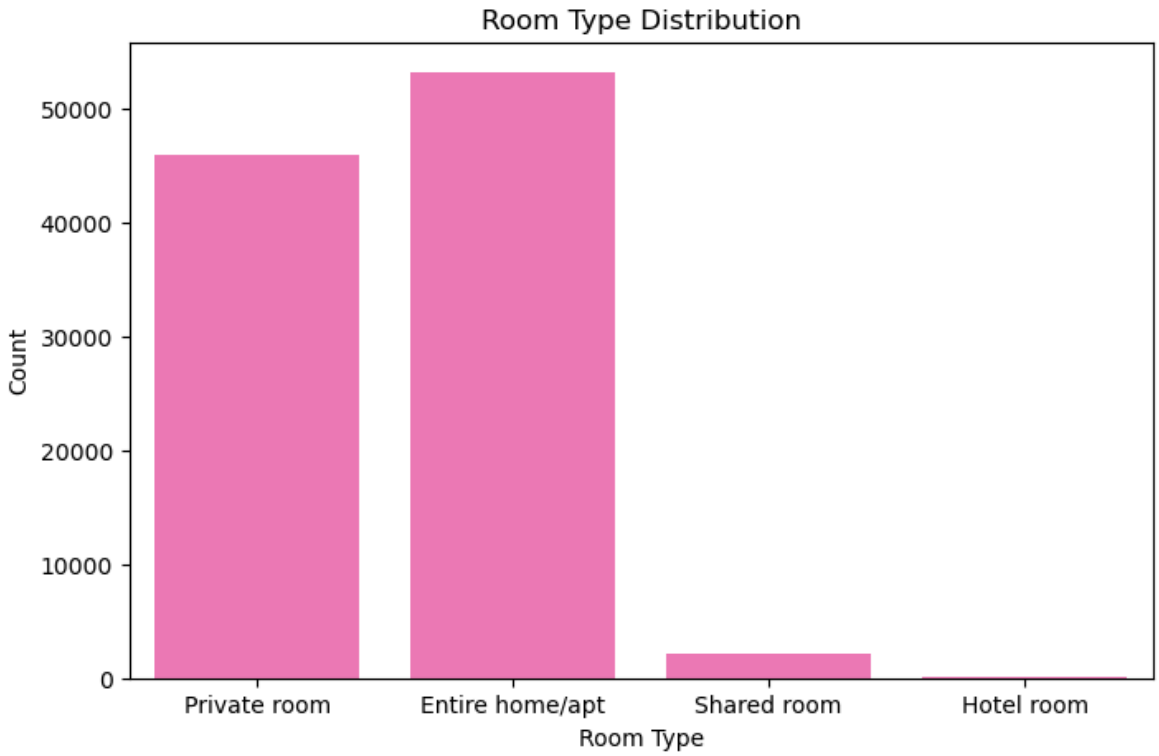


In [ ]:
```
The histogram shows a fairly even distribution of
listing prices across different price ranges, indicating no
particular concentration of listings in any specific price range.
The KDE line helps visualize this even spread more clearly,
confirming that the dataset contains listings with a wide variety of prices.
```

Room Type Analysis

Analyze the distribution of different room types.

In [65]:
```
plt.figure(figsize=(8, 5))
sns.countplot(x='room type', data=df , color='hotpink')
plt.title('Room Type Distribution')
plt.xlabel('Room Type')
plt.ylabel('Count')
plt.show()
```

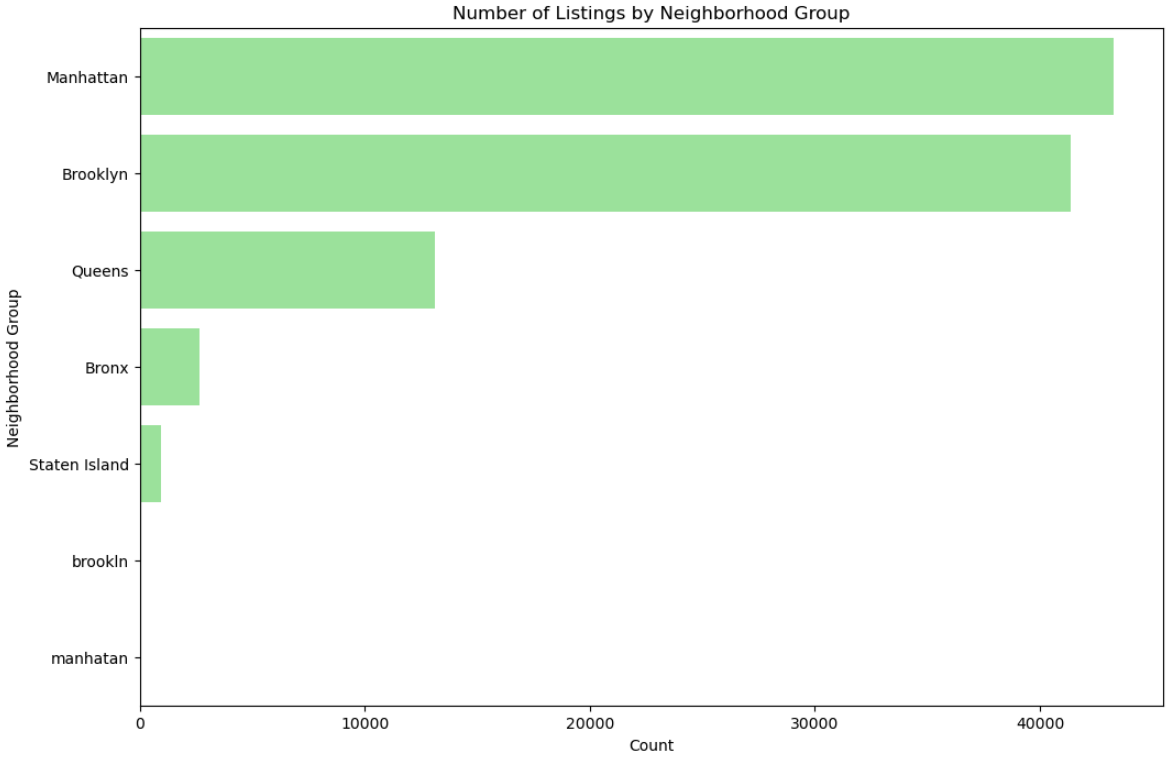


In [ ]:
```
The count plot shows a clear distribution of the different
room types available in the Airbnb dataset.
The majority of listings are for
'Entire home/apt' and 'Private room', with 'Shared room' and 'Hotel room'
being much less common. This insight can be useful
for understanding the availability and popularity of
different types of accommodations on Airbnb.
```

Neighborhood Analysis

Examine how listings are distributed across different neighborhoods.

In [68]:
```
plt.figure(figsize=(12, 8))
sns.countplot(y='neighbourhood group', data=df,color="lightgreen" , order=df['ne
plt.title('Number of Listings by Neighborhood Group')
plt.xlabel('Count')
plt.ylabel('Neighborhood Group')
plt.show()
```



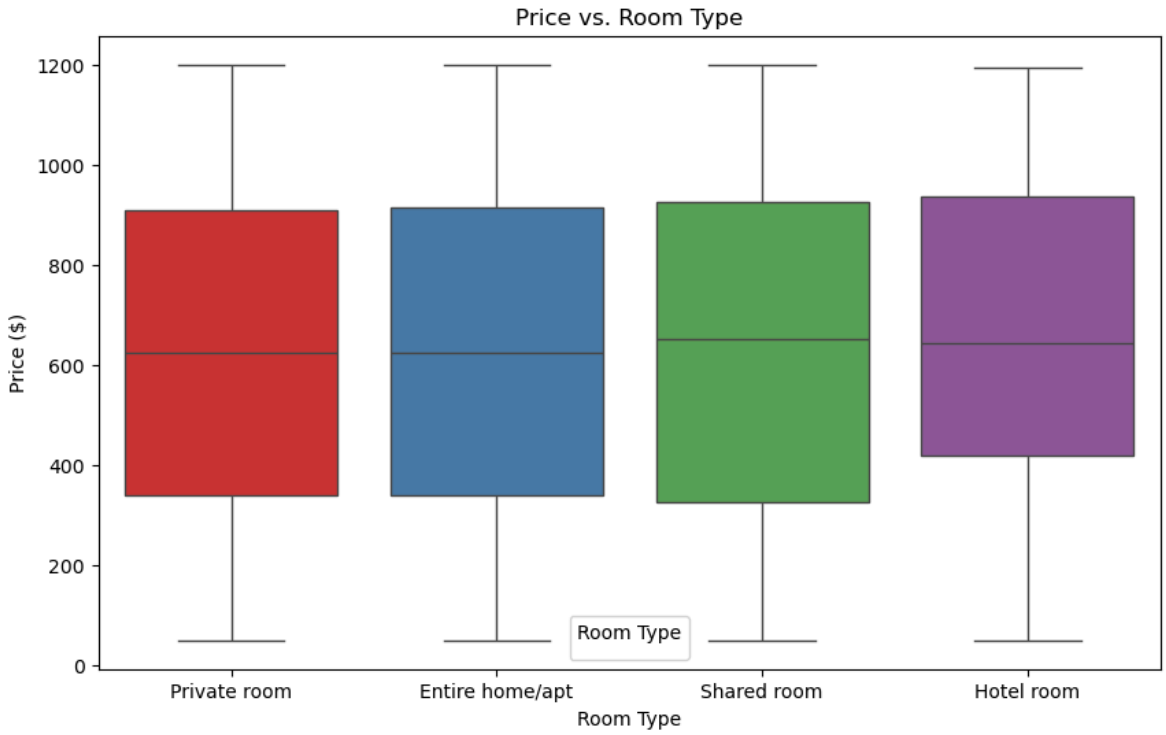
In [ ]:
```
The count plot shows a clear distribution
of the number of listings across different neighborhood
groups. Manhattan and Brooklyn dominate the listings,
suggesting they are prime locations for Airbnb. Queens,
Bronx, and Staten Island have fewer listings,
indicating less availability or popularity.
```

Price vs. Room Type

Visualize the relationship between price and room type

In [75]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='room type', y='price', hue='room type', data=df, palette='Set1')
plt.title('Price vs. Room Type')
plt.xlabel('Room Type')
plt.ylabel('Price ($)')
plt.legend(title='Room Type')
plt.show()
```

```
C:\Users\swati\AppData\Local\Temp\ipykernel_1952\1262699844.py:6: UserWarning: No
artists with labels found to put in legend.  Note that artists whose label start
with an underscore are ignored when legend() is called with no argument.
  plt.legend(title='Room Type')
```



In [ ]:
```
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.
```

In [79]:
```
df.head()
```

Out[79]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brookly
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhatta
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhatta
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhatta
5	1004098	Large Cozy 1 BR Apartment In Midtown East	45498551794	verified	Michelle	Manhatta

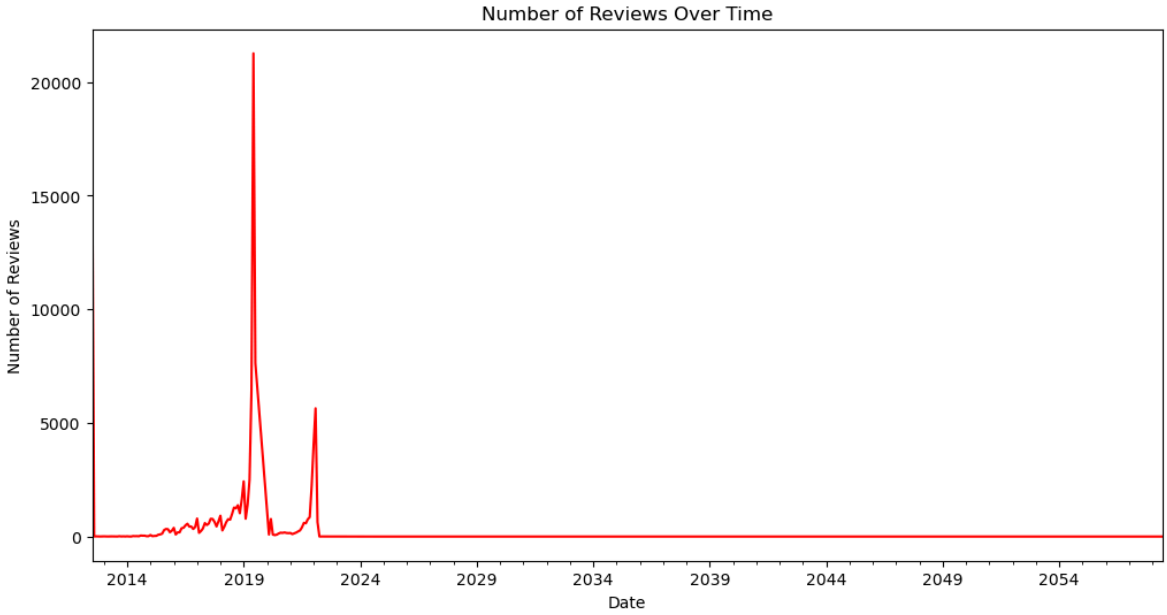
5 rows × 24 columns

Reviews Over Time

Plot the number of reviews over time.

In [82]:
```
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=(12, 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()
```



In [ ]:
```
The line plot provides a clear visualization
of the number of reviews over time.
It helps identify trends and patterns in review activity,
such as periods of high or low activity.
This information can be useful for understanding
    the dynamics of user engagement and the popularity
of Airbnb listings over time. The significant spikes
and drops in reviews might be worth further investigation
    to understand the underlying causes, such as changes
    in Airbnb policies, market conditions, or external events.
```

In [ ]:

In [ ]: