AIRBNB PROJECT

ANALYSIS WITH PYTHON

Introduction

Airbnb is an online marketplace that connects people looking to rent out their homes with those seeking accommodations. It allows individuals to offer their homes, apartments, or even a room to travelers, typically at competitive rates compared to traditional hotels.

KEY ASPECTS

- User Reviews
- Experiences
- Hosts
- Variety of Listings
- Guests

START EXPLORING THE DATA

LISTINGS DATA:

```
df= pd.read_csv('listings.csv')
```

[3] Python

```
df.head()
```

[4] Python

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	re
0	6606	Fab, private seattle urban cottage!	14942	Joyce	Other neighborhoods	Wallingford	47.65444	-122.33629	Entire home/apt	99.0	30	160	2023-08-05	
1	9419	Glorious sun room w/ memory foambed	30559	Angielena	Other neighborhoods	Georgetown	47.55017	-122.31937	Private room	76.0	2	196	2024-06-09	
2	9531	The Adorable Sweet Orange Craftsman	31481	Cassie	West Seattle	Fairmount Park	47.55495	-122.38663	Entire home/apt	189.0	3	97	2024-06-16	
3	9534	The Coolest Tangerine	31481	Cassie	West Seattle	Fairmount Park	47.55627	-122.38607	Entire home/apt	125.0	2	77	2023-12-27	

```
[5]  df.shape
```

```
...  (6442, 18)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6442 entries, 0 to 6441
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              6442 non-null   int64  
 1   name                            6442 non-null   object 
 2   host_id                         6442 non-null   int64  
 3   host_name                       6442 non-null   object 
 4   neighbourhood_group             6442 non-null   object 
 5   neighbourhood                   6442 non-null   object 
 6   latitude                        6442 non-null   float64
 7   longitude                       6442 non-null   float64
 8   room_type                       6442 non-null   object 
 9   price                           6011 non-null   float64
 10  minimum_nights                  6442 non-null   int64  
 11  number_of_reviews               6442 non-null   int64  
 12  last_review                     5601 non-null   object 
 13  reviews_per_month               5601 non-null   float64
 14  calculated_host_listings_count  6442 non-null   int64  
 15  availability_365                6442 non-null   int64  
 16  number_of_reviews_ltm           6442 non-null   int64  
 17  license                         5312 non-null   object 
dtypes: float64(4), int64(7), object(7)
memory usage: 906.0+ KB
```

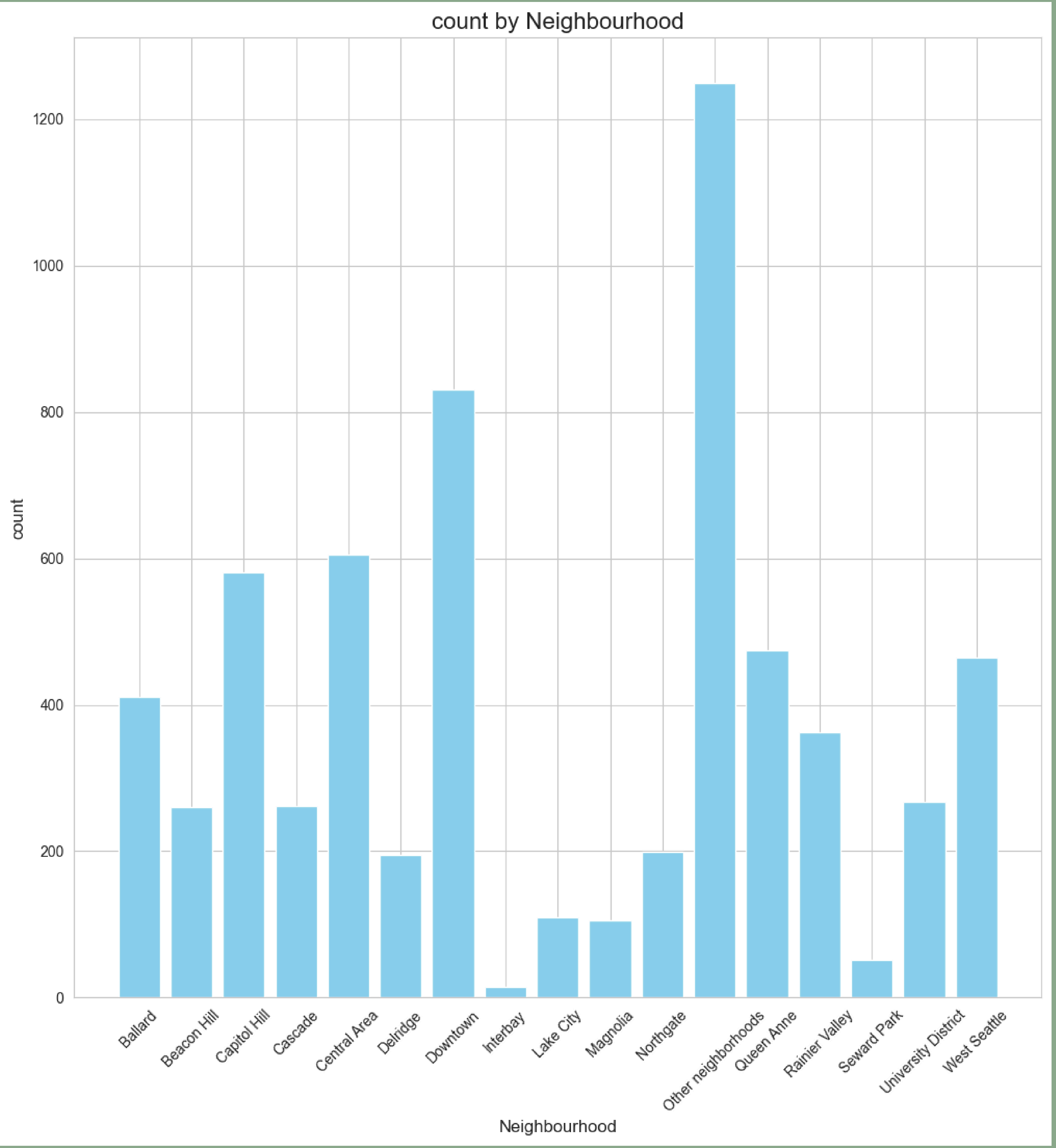
UNIVARIATE AND BIVARIATE ANALYSIS

```
import numpy as nd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

```
df1=df.groupby('neighbourhood_group')['id'].count().reset_index()
```

	neighbourhood_group	id
0	Ballard	411
1	Beacon Hill	260
2	Capitol Hill	581
3	Cascade	261
4	Central Area	605
5	Delridge	195
6	Downtown	831
7	Interbay	15
8	Lake City	110
9	Magnolia	105
10	Northgate	199
11	Other neighborhoods	1249
12	Queen Anne	474
13	Rainier Valley	362
14	Seward Park	52
15	University District	268
16	West Seattle	464

```
plt.figure(figsize=(12, 12))  # Optional: Set the figure size
plt.bar(df1['neighbourhood_group'], df1['id'], color='skyblue')
plt.title('count by Neighbourhood', fontsize=16)
plt.xlabel('Neighbourhood', fontsize=12)
plt.ylabel('count', fontsize=12)
plt.xticks(rotation=45)  # Rotate the x-axis labels for better readabilit

# Show the plot
plt.show()
```



- The Downtown area (831) and Central Area (605) follow "Other neighborhoods" as the second and third most represented neighborhoods, suggesting these are highly active or populated areas.
- Queen Anne (474), Capitol Hill (581), and West Seattle (464) are pointing to mid-level engagement

```python
df2=df.groupby('neighbourhood')['id'].count().reset_index()
```

```python
df2
```

	neighbourhood	id
0	Adams	126
1	Alki	117
2	Arbor Heights	24
3	Atlantic	131
4	Belltown	343
...	...	...
83	West Woodland	113
84	Westlake	18
85	Whittier Heights	70
86	Windermere	14
87	Yesler Terrace	69

88 rows × 2 columns

```python
df3=df.groupby('room_type')['id'].count().reset_index()
```

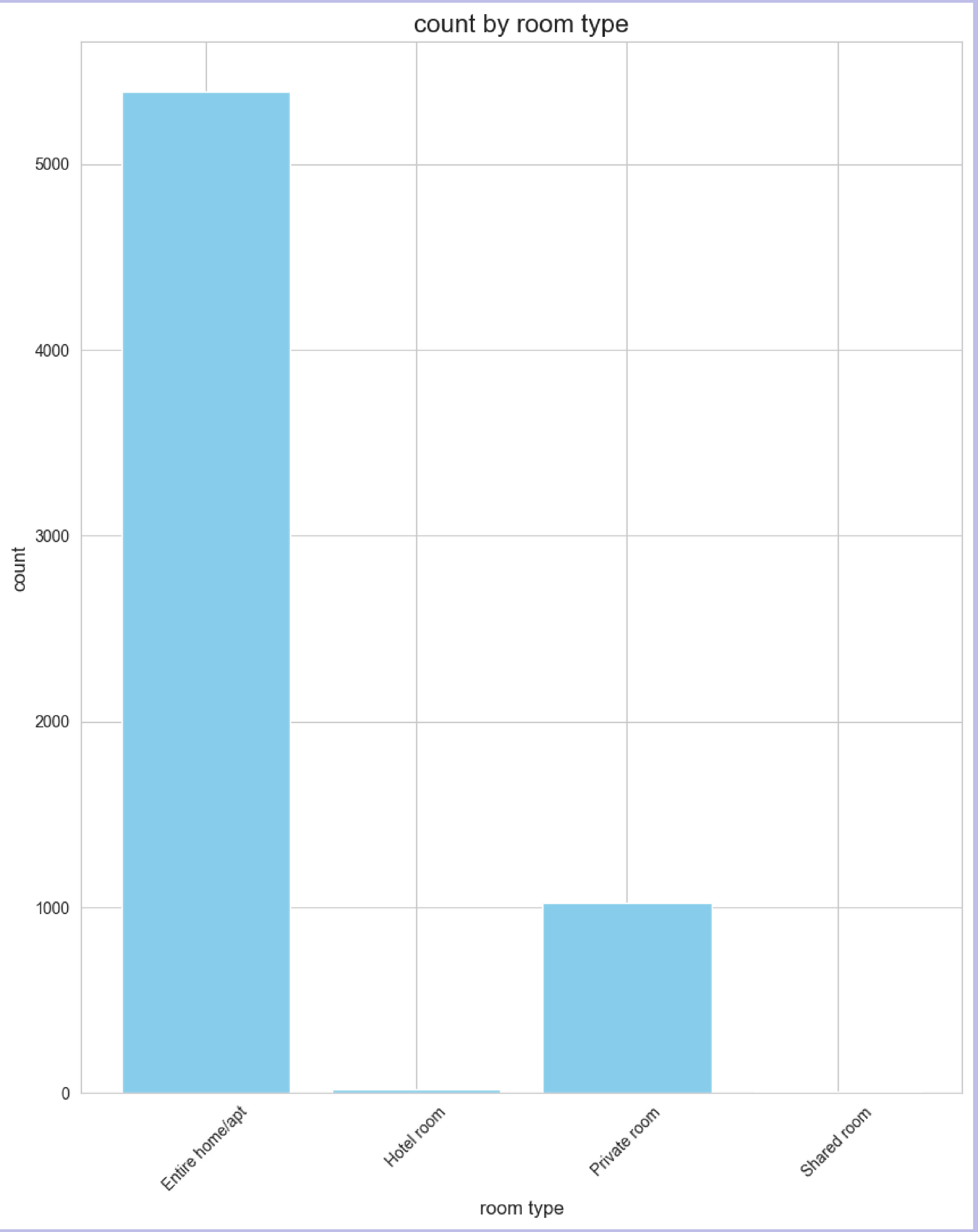
[20]

```python
df3
```

[21]

	room_type	id
0	Entire home/apt	5387
1	Hotel room	21
2	Private room	1024
3	Shared room	10

```python
plt.figure(figsize=(10, 12))  # Optional: Set the figure size
plt.bar(df3['room_type'], df3['id'], color='skyblue')
plt.title('count by room type', fontsize=16)
plt.xlabel('room type', fontsize=12)
plt.ylabel('count', fontsize=12)
plt.xticks(rotation=45)  # Rotate the x-axis labels for better readability

# Show the plot
plt.show()
```

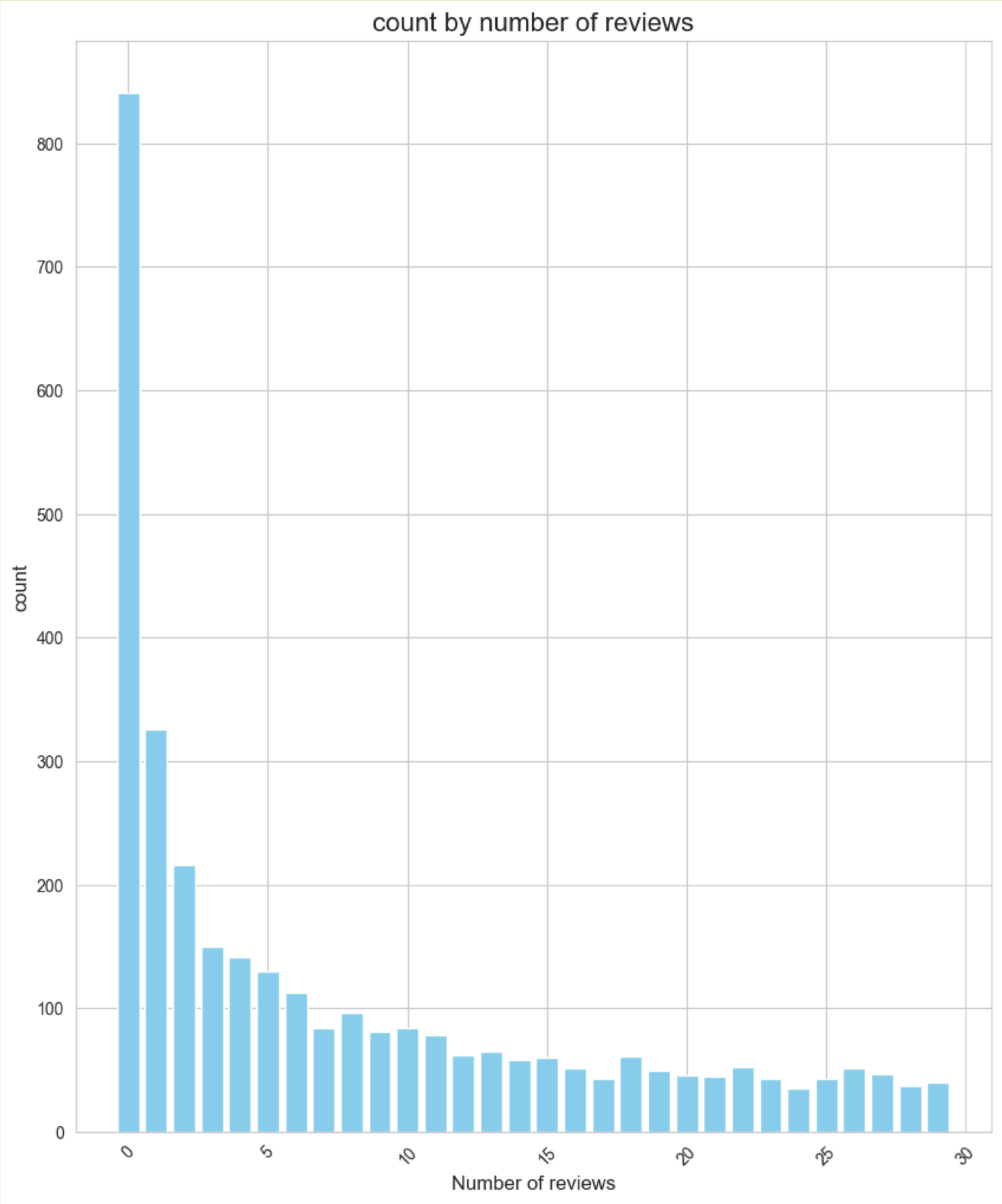


- "Entire home/apt" has the highest count with 5,387, indicating that this room type is the most commonly available or in demand

- "Private room" comes second with 1,024 which shows a significant demand for private rooms, likely driven by budget-conscious travelers .

```
df4=df.groupby('number_of_reviews')['id'].count().reset_index().head(30)
```

```
plt.figure(figsize=(10, 12))  # Optional: Set the figure size
plt.bar(df4['number_of_reviews'], df4['id'], color='skyblue')
plt.title('count by number of reviews', fontsize=16)
plt.xlabel('Number of reviews', fontsize=12)
plt.ylabel('count', fontsize=12)
plt.xticks(rotation=45)  # Rotate the x-axis labels for better readability

# Show the plot
plt.show()
```



- There are 841 listings with zero reviews, the highest count in the dataset. This could indicate a large number of new or less popular listings that haven't been reviewed yet.
- The number of listings decreases as the number of reviews increases.
- Listings with more than 20 reviews show a clear decline, with counts ranging between 35 and 50.

```python
df['last_review']=pd.to_datetime(df['last_review'])
```

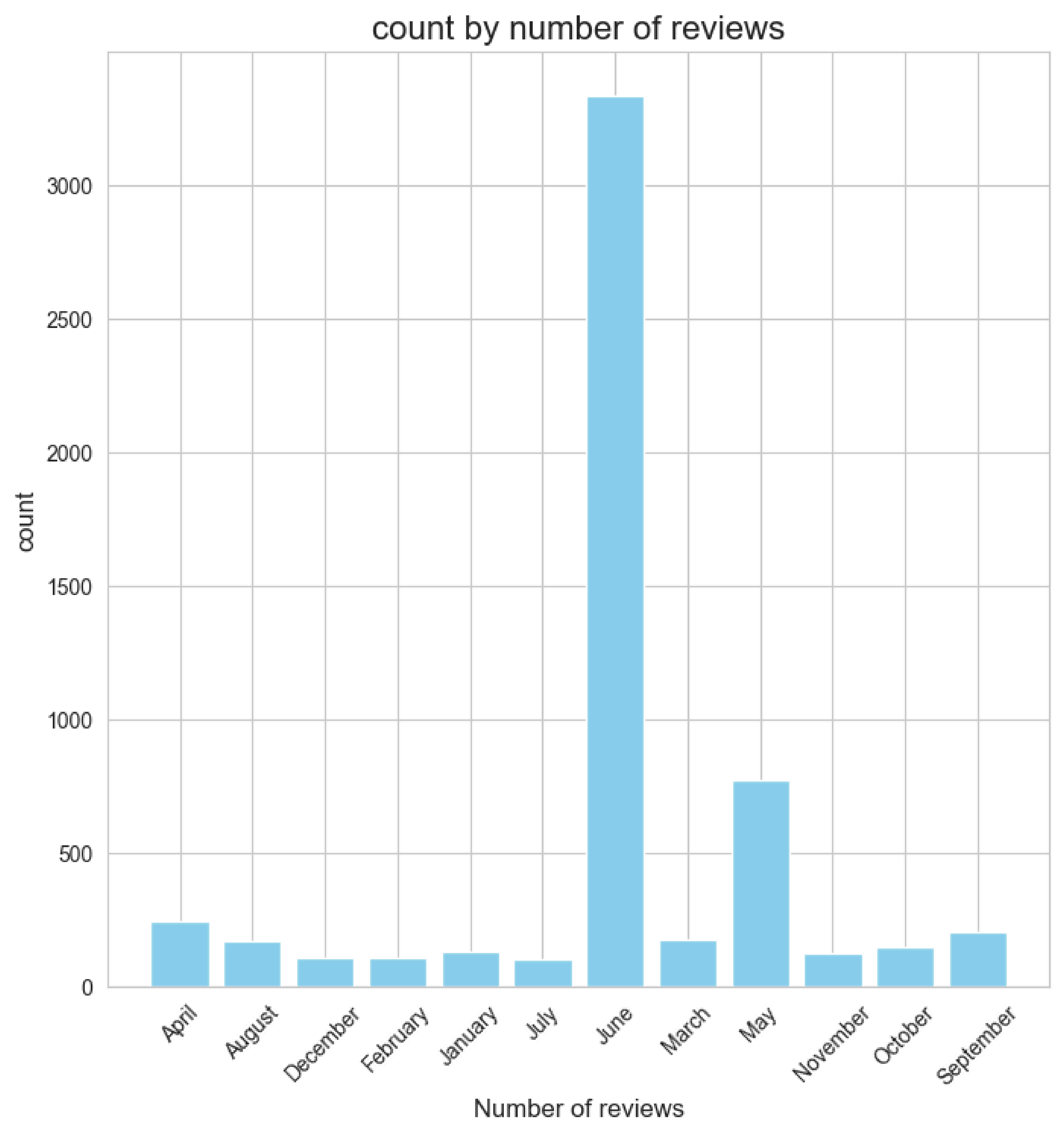
```python
df['last_review_month']=df['last_review'].dt.month_name()
```

```python
df5=df.groupby('last_review_month')['number_of_reviews'].count().reset_index()
```

```
df5
```

	last_review_month	number_of_reviews
0	April	245
1	August	167
2	December	107
3	February	106
4	January	128
5	July	99
6	June	3332
7	March	173
8	May	771
9	November	123
10	October	147
11	September	203

```
plt.figure(figsize=(8,8 ))  # Optional: Set the figure size
plt.bar(df5['last_review_month'], df5['number_of_reviews'], color='skyblue')
plt.title('count by number of reviews', fontsize=16)
plt.xlabel('Number of reviews', fontsize=12)
plt.ylabel('count', fontsize=12)
plt.xticks(rotation=45)  # Rotate the x-axis labels for better readability

# Show the plot
plt.show()
```



- June accounts for the highest percentage of reviews at 59.49%, showing a significant spike in activity compared to other months.
- May follows with 13.77% of reviews, further supporting a busy late spring and early summer period for bookings.

[32]

```python
df['last_review_YEAR']=df['last_review'].dt.year
```

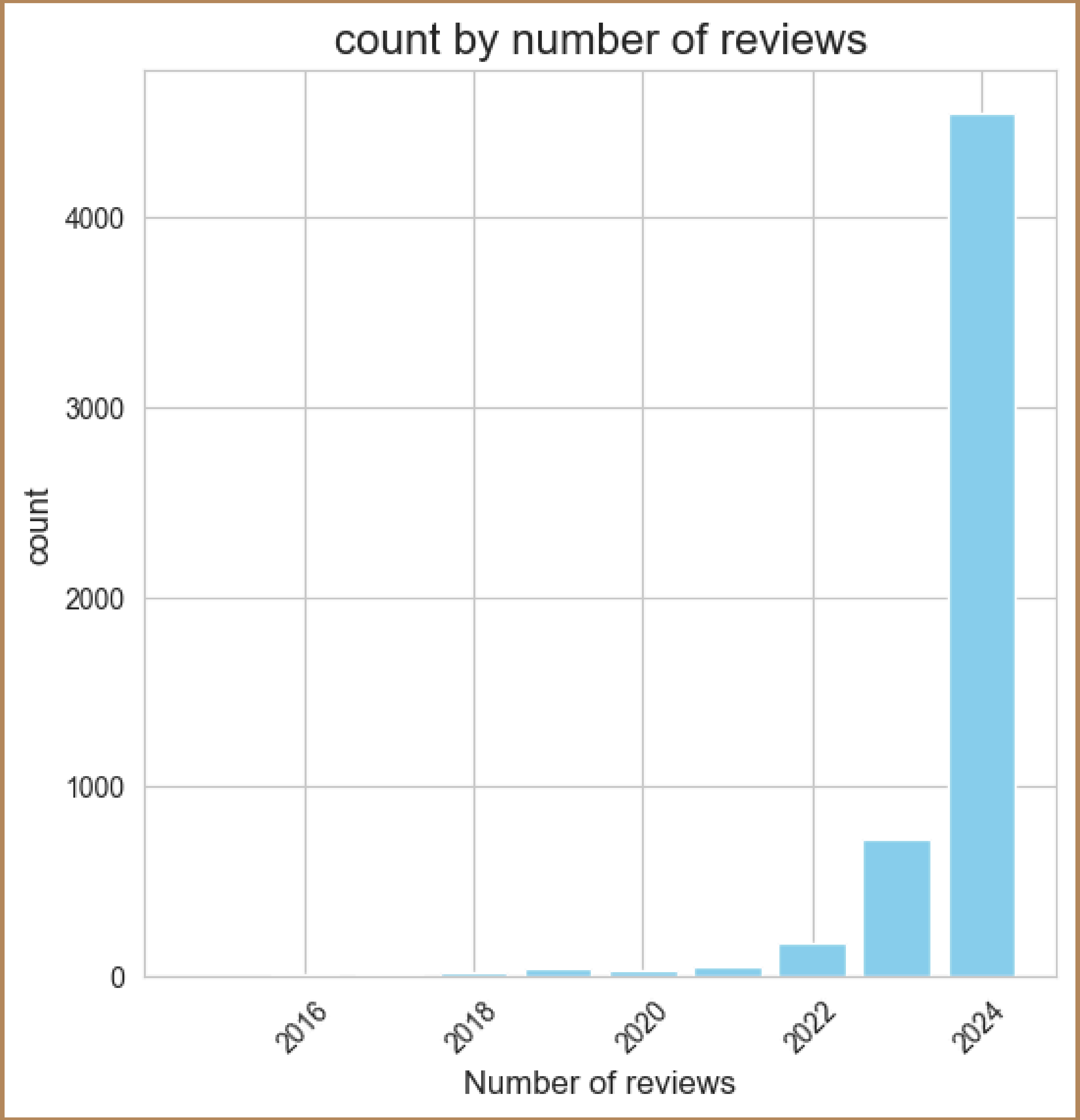
[33]

```python
df6=df.groupby('last_review_YEAR')['number_of_reviews'].count().reset_index()
```

```python
df6
```

	last_review_YEAR	number_of_reviews
0	2015.0	3
1	2016.0	6
2	2017.0	5
3	2018.0	16
4	2019.0	40
5	2020.0	28
6	2021.0	54
7	2022.0	178
8	2023.0	720
9	2024.0	4551

```
plt.figure(figsize=(6,6 ))  # Optional: Set the figure size
plt.bar(df6['last_review_YEAR'], df6['number_of_reviews'], color='skyblue')
plt.title('count by number of reviews', fontsize=16)
plt.xlabel('Number of reviews', fontsize=12)
plt.ylabel('count', fontsize=12)
plt.xticks(rotation=45)  # Rotate the x-axis labels for better readability

# Show the plot
plt.show()
```



- Starting in 2021, there is a significant rise in the number of reviews. In 2021, the reviews increased to 54, showing an 88% increase from the previous year (2020).
- Explosive Growth in 2023 and 2024
- The majority of reviews (44.68%) were received in 2024. When combined with 2023, these two years account for 51.74% of all reviews

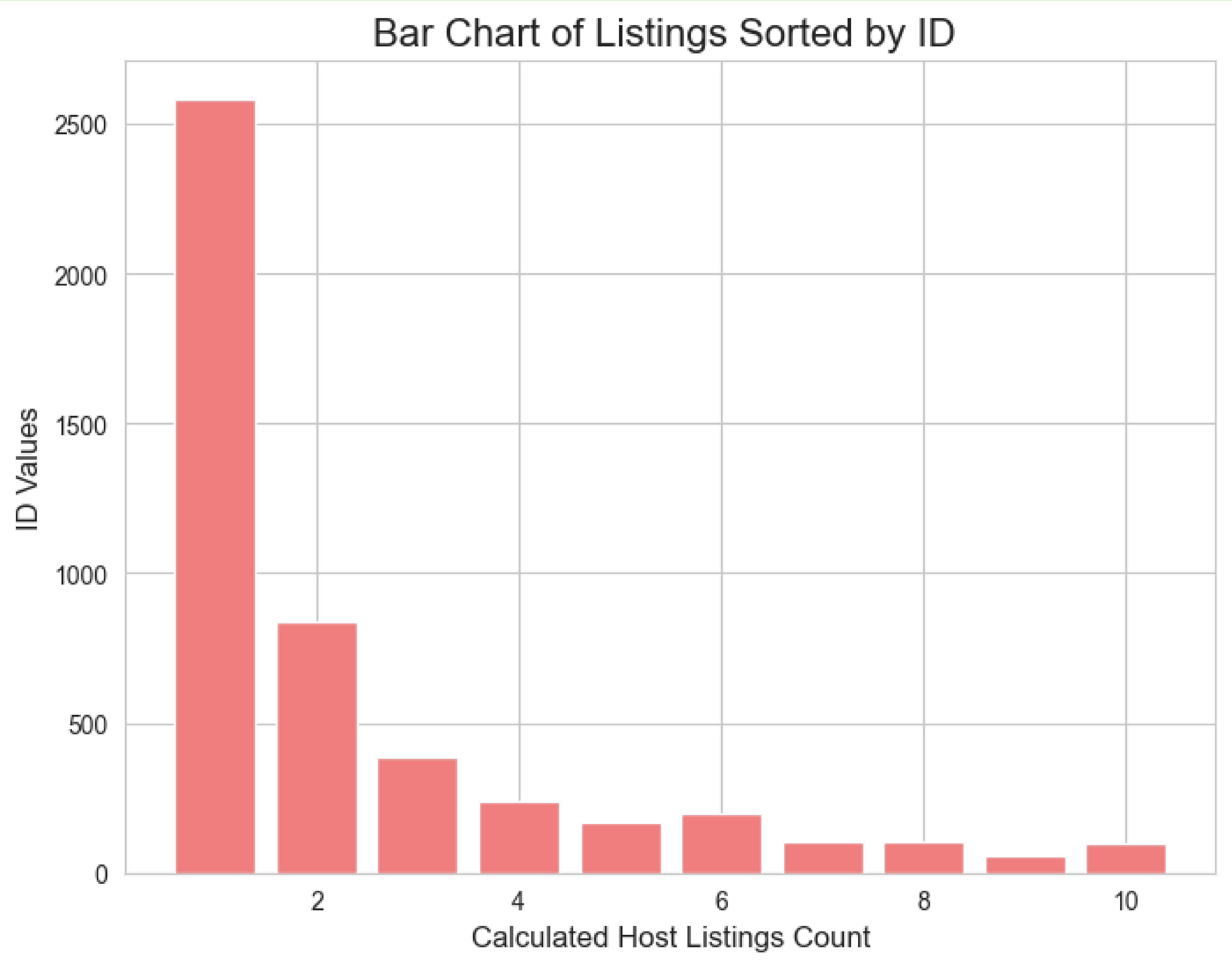
```
df7=df.groupby(['calculated_host_listings_count'])['id'].count().reset_index().head(10)
```

36]

```
df7
```

7]

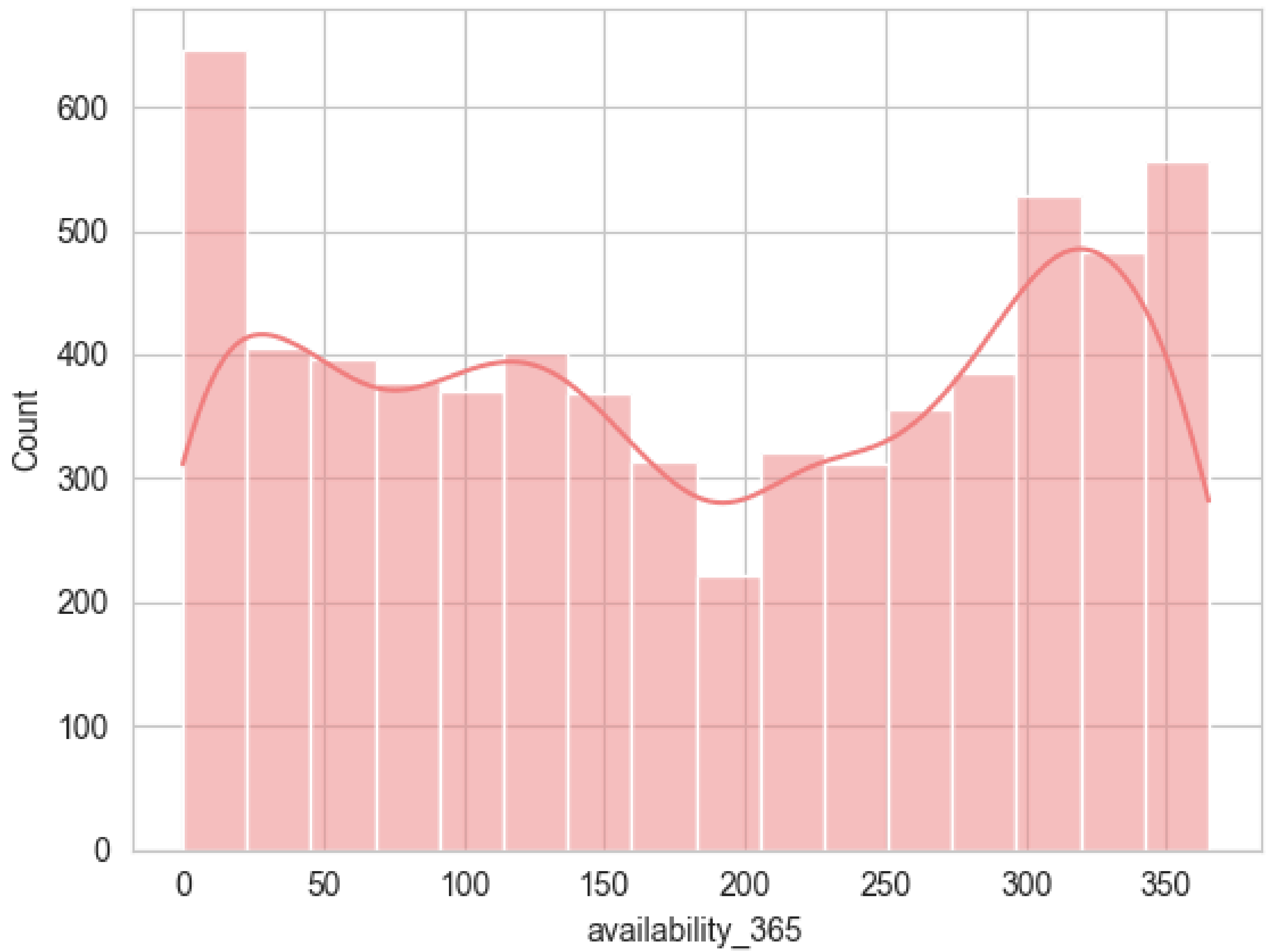
	calculated_host_listings_count	id
0	1	2585
1	2	840
2	3	387
3	4	236
4	5	170
5	6	198
6	7	105
7	8	104
8	9	54
9	10	100



- Most Hosts Have Few Listings
  - 61.34% of hosts manage only one listing
  - 19.93% of hosts manage two listings
  - Hosts managing between 3 and 6 listings represent about 9.18% to 4.70% of the total.

```python
import seaborn as sns
sns.histplot(x='availability_365', data=df,kde=True,color='lightcoral')
```

```
<Axes: xlabel='availability_365', ylabel='Count'>
```



- Some listings are nearly unavailable, which could suggest various reasons such as being reserved for personal use, long-term rentals, or simply inactive listings.
- On the other hand, there are listings available for 350 days a year, which suggests a more professional or business-driven approach, with these listings likely being short-term rentals or managed by full-time hosts.

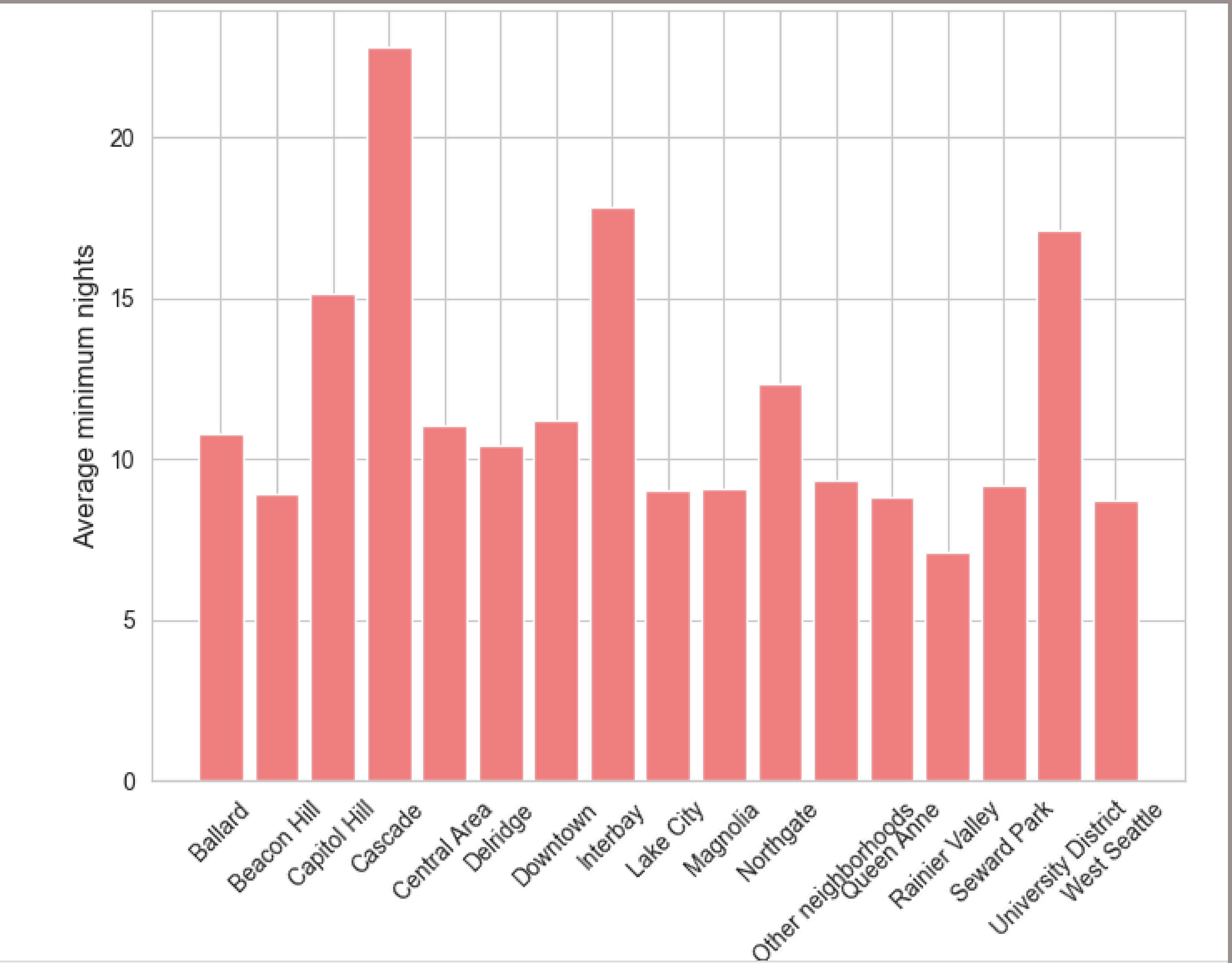
```python
df8=df.groupby('room_type')['price'].mean().reset_index()
```

```python
plt.figure(figsize=(8, 6))
plt.bar(df8['room_type'], df8['price'], color='lightcoral')

# Add labels and title
plt.title('Bar Chart of room type Sorted by ID ', fontsize=16)
plt.xlabel('Calculated room type', fontsize=12)
plt.ylabel('Average Price', fontsize=12)

# Display the plot
plt.show()
```

- Hotel rooms are approximately 5.3 times more expensive than shared rooms and 2.37 times more expensive than entire homes/apartments.
- Private rooms are about 1.24 times more expensive than shared rooms, offering more privacy at a moderate increase in price.

```
df9=df.groupby('neighbourhood_group')['minimum_nights'].mean().reset_index()
```

	neighbourhood_group	minimum_nights
0	Ballard	10.754258
1	Beacon Hill	8.907692
2	Capitol Hill	15.156627
3	Cascade	22.793103
4	Central Area	11.039669
5	Delridge	10.441026
6	Downtown	11.172082
7	Interbay	17.800000
8	Lake City	9.000000
9	Magnolia	9.095238
10	Northgate	12.321608
11	Other neighborhoods	9.336269
12	Queen Anne	8.831224
13	Rainier Valley	7.110497
14	Seward Park	9.153846
15	University District	17.111940
16	West Seattle	8.715517

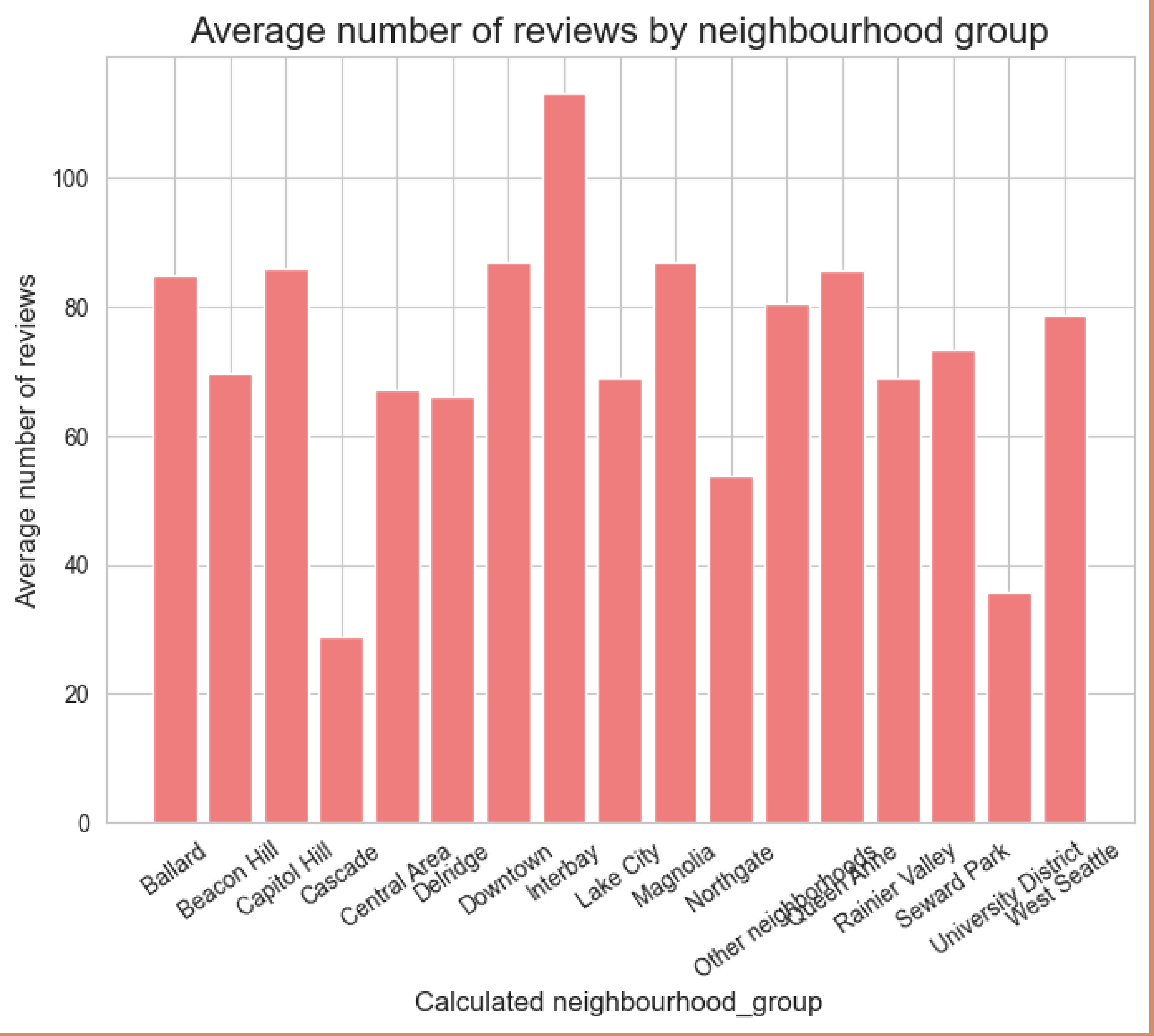


- Neighborhoods like **Cascade, Capitol Hill, Interbay, and University District** show a preference for longer-term stays likely attracting visitors

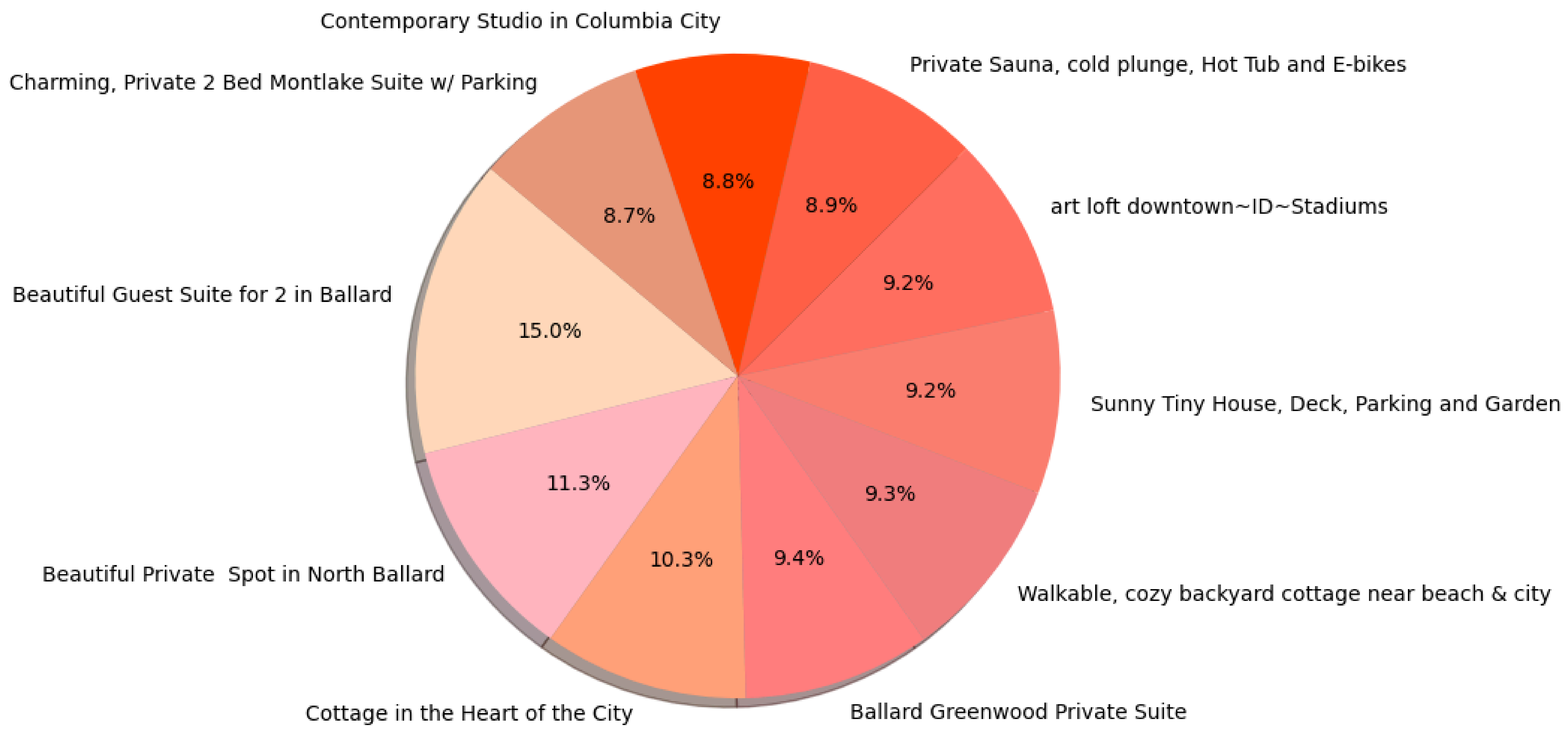
- Shorter minimum stays are more common in **Rainier Valley, West Seattle, Beacon Hill, and Queen Anne**, making these neighborhoods more suitable for short-term tourists .

```
df10=df.groupby('neighbourhood_group')['number_of_reviews'].mean().reset_index()
```

```
plt.figure(figsize=(8, 6))
plt.bar(df10['neighbourhood_group'], df10['number_of_reviews'], color='lightcoral')

# Add labels and title
plt.title('Average number of reviews by neighbourhood group', fontsize=16)
plt.xlabel('Calculated neighbourhood_group', fontsize=12)
plt.ylabel('Average number of reviews', fontsize=12)
plt.xticks(rotation=35)
# Display the plot
plt.show()
```



- Interbay, Downtown, and Magnolia are top performers in terms of guest reviews, showing high engagement and likely popularity among visitors.
- Neighborhoods with lower review counts, like Cascade and University District, may either be less popular or have fewer listings available.
- Capitol Hill, Queen Anne, and Ballard are well-established in terms of guest interactions, indicating steady demand in these neighborhoods.


Contemporary Studio in Columbia City
Private Sauna, cold plunge, Hot Tub and E-bikes
Charming, Private 2 Bed Montlake Suite w/ Parking
art loft downtown~ID~Stadiums
Beautiful Guest Suite for 2 in Ballard
Sunny Tiny House, Deck, Parking and Garden
Beautiful Private Spot in North Ballard
Walkable, cozy backyard cottage near beach & city
Cottage in the Heart of the City
Ballard Greenwood Private Suite
8.8%
8.9%
8.7%
9.2%
15.0%
9.2%
11.3%
9.3%
10.3%
9.4%

```python
df['occupancy_rate']= (365-df['availability_365'])/365
```

```python
df['occupancy_rate'].head(10)
```

✓ 0.0s

```
0    0.597260
1    0.076712
2    0.635616
3    0.997260
4    0.956164
5    0.736986
6    0.747945
7    0.139726
8    0.526027
9    0.882192
Name: occupancy_rate, dtype: float64
```

```python
df['net_revenue']=df['price']*(365-df['availability_365'])
```

✓ 0.0s

```python
df.groupby('room_type')['net_revenue'].mean().reset_index()
```

✓ 0.0s

	room_type	net_revenue
0	Entire home/apt	38399.101996
1	Hotel room	25849.700000
2	Private room	15488.637744
3	Shared room	11654.000000

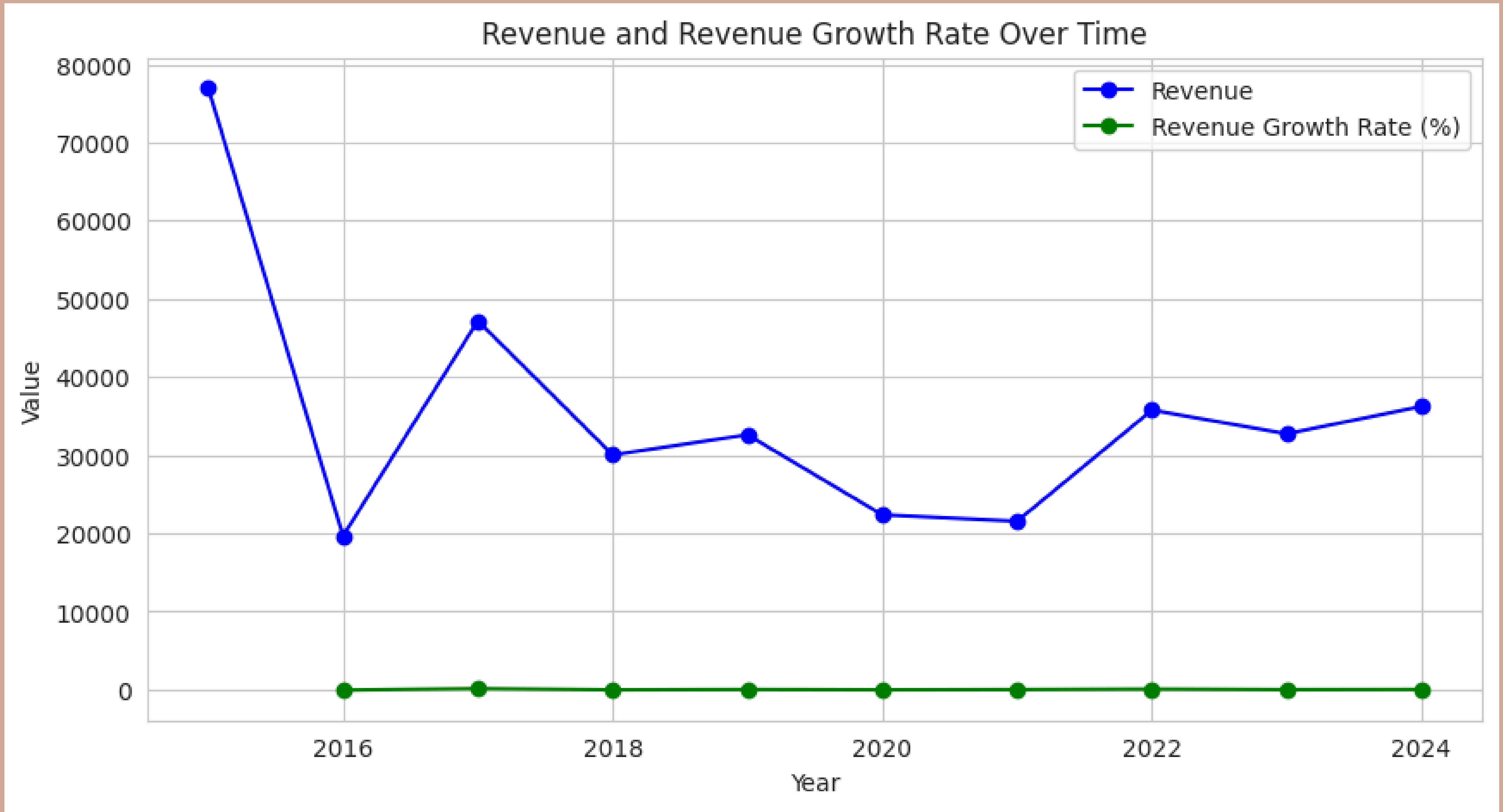
- With a net revenue of 38,399.10 (accounting for 41.57% of total revenue), entire homes or apartments are the highest revenue earners
- Hotel rooms contribute 27.99% of the total revenue with a net revenue of 25,849.70.

```python
data4=df.groupby('last_review_YEAR')['net_revenue'].mean().reset_index()
```

✓ 0.0s

\+ Code \+ Markdown

```python
data4['Revenue_Growth_Rate'] = data4['net_revenue'].pct_change() * 100
```



```python
data5=df.groupby('neighbourhood_group')['calculated_host_listings_count'].mean().reset_index()
```

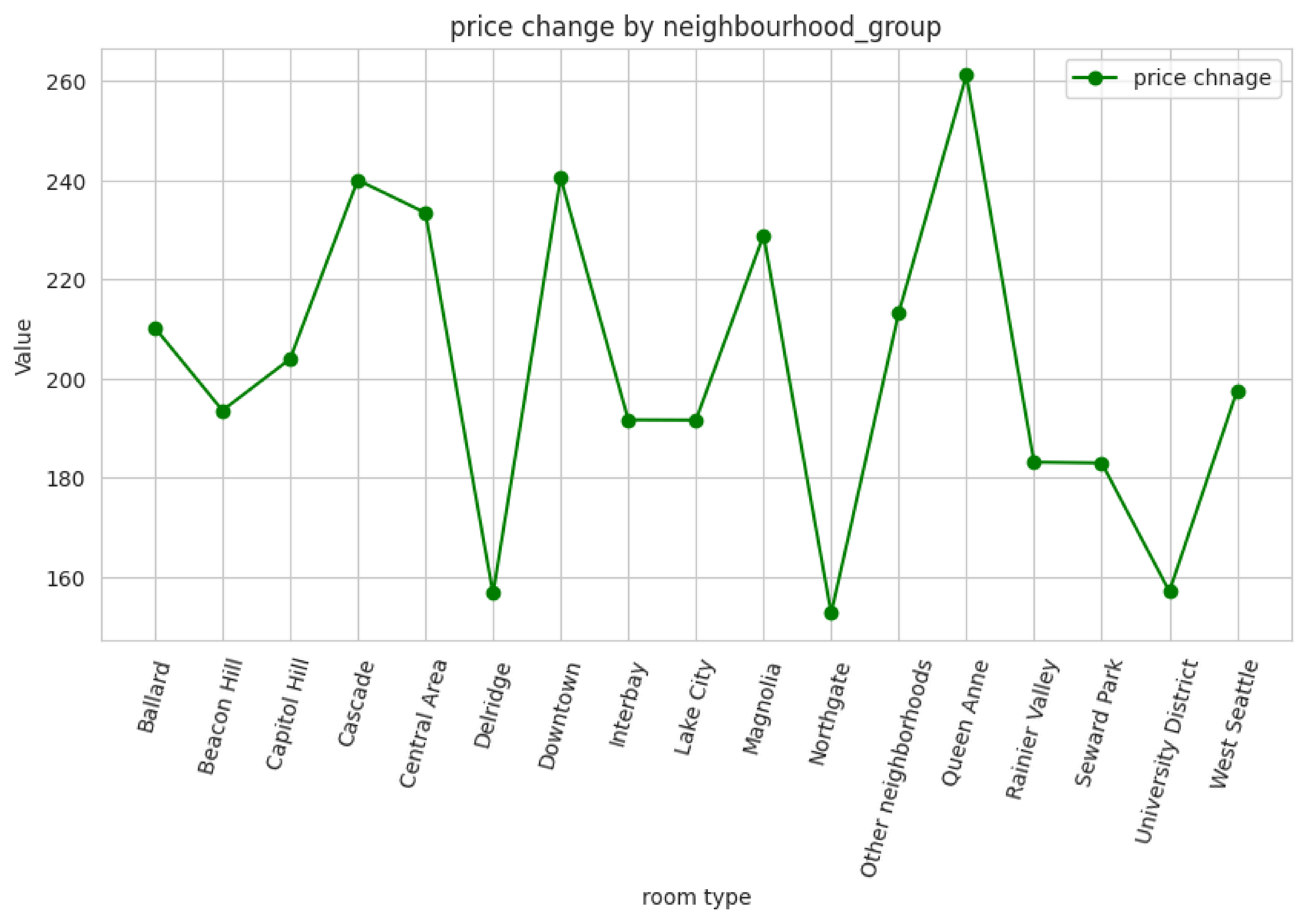
```python
data5=df.groupby('last_review_YEAR')['number_of_reviews'].mean().reset_index()
```

```python
data5['pct_chnge_qty']=data5['number_of_reviews'].pct_change() * 100
```

```python
data5['price_elasticity'] = data5['pct_chnge_qty'] / data4['Revenue_Growth_Rate']
```

	last_review_YEAR	number_of_reviews	pct_chnge_qty	price_elasticity
0	2015.0	23.666667	NaN	NaN
1	2016.0	10.500000	-55.633803	0.747161
2	2017.0	29.800000	183.809524	1.313035
3	2018.0	8.687500	-70.847315	1.948947
4	2019.0	54.200000	523.884892	61.146902
5	2020.0	90.321429	66.644702	-2.121453
6	2021.0	50.944444	-43.596503	11.999008
7	2022.0	26.606742	-47.773026	-0.724619
8	2023.0	34.358333	29.133939	-3.445921
9	2024.0	97.562294	183.955258	17.246746

```
data8=df.groupby('neighbourhood_group')['price'].mean().reset_index()
```



```python
host_avg_reviews = df.groupby('host_id')['reviews_per_month'].mean().reset_index()

# Merge with availability_365
host_info = df[['host_id', 'availability_365']].drop_duplicates()
host_avg_reviews = host_avg_reviews.merge(host_info, on='host_id')

# Sort hosts by highest average reviews per month and lowest availability_365
top_hosts = host_avg_reviews.sort_values(by=['reviews_per_month', 'availability_365'], ascending=[False, True])

print(top_hosts.head(10))
```

```
        host_id  reviews_per_month  availability_365
5694  569419884          59.533333               134
5695  569419884          59.533333               155
5696  569419884          59.533333               158
5050  437880135          15.560000                50
4330  240730356          13.510000               117
1477   18141906          12.000000                 3
3805  137418982          11.820000                37
3804  137418982          11.820000               155
1542   19928221          11.740000               216
4741  380578321          11.290000               274
```

RECOMMENDATIONS

- Invest in targeted marketing campaigns for entire homes/apartments to further boost their already high revenue contribution of 41.57%.
- Optimize pricing for hotel rooms to attract more short-term guests, as they have a strong revenue share but might be underutilized due to high costs.
- Entire homes/apartments dominate revenue, indicating high demand for private, long-term accommodation.
- Neighborhoods with high review counts, such as Interbay and Downtown, are likely to see higher guest satisfaction and demand, making them valuable areas to prioritize.

THANK YOU