

New York City Airbnb Data

Ezgi Efe

The libraries I used:

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

About my dataset:

```
df= pd.read_csv("C:/Users/ezgie/OneDrive/Masaüstü/pythonproject/.vscode/AB_NYC_2019.csv")
df.info()
```

```
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 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                            38843 non-null  object
13  reviews_per_month                     38843 non-null  float64
14  calculated_host_listings_count         48895 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
>>> []
```

df.head(10)

```
[5 rows x 16 columns]
```

```
>>> df.head(10)
```

	id	name	...	calculated_host_listings_count	availability_365
0	2539	Clean & quiet apt home by the park	...	6	365
1	2595	Skylit Midtown Castle	...	2	355
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	...	1	365
3	3831	Cozy Entire Floor of Brownstone	...	1	194
4	5022	Entire Apt: Spacious Studio/Loft by central park	...	1	0
5	5099	Large Cozy 1 BR Apartment In Midtown East	...	1	129
6	5121	BlissArtsSpace!	...	1	0
7	5178	Large Furnished Room Near B'way	...	1	220
8	5203	Cozy Clean Guest Room - Family Apt	...	1	0
9	5238	Cute & Cozy Lower East Side 1 bdrm	...	4	188

```
[10 rows x 16 columns]
```

```
>>> []
```

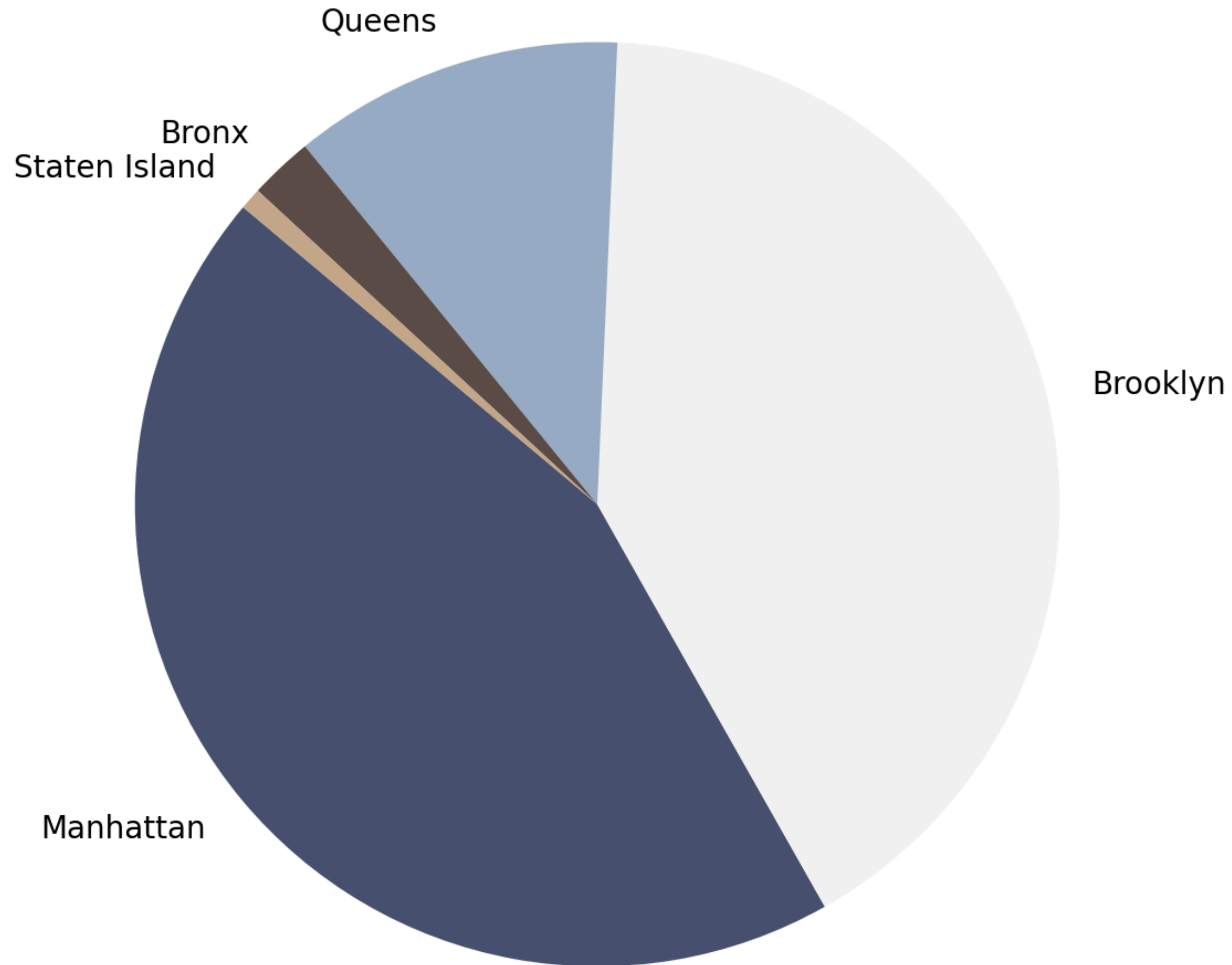
Filtering my data:

```
#Im not going to work with the whole data so i filter the colmns i want to work with  
selected_columns = ['host_id', 'neighbourhood_group', 'neighbourhood', 'room_type', 'price',  
                    'number_of_reviews', 'reviews_per_month']  
newdf = df[selected_columns]
```

How many housings in each neighbourhood group:

```
# I'm counting the datas in neighbourhood_group
neighbourhood_group_counts = newdf["neighbourhood_group"].value_counts()
# neighbourhood_group Pie Chart
plt.figure(figsize=(8, 8))
plt.pie(neighbourhood_group_counts, labels=neighbourhood_group_counts.index, startangle=140,
        colors=["#48516E", "#F4F4F4", "#9AAEC4", "#5B4D47", "#C6A687", "#F4F4F4"])
plt.title("Neighbourhood Group Distribution")
plt.show()
```

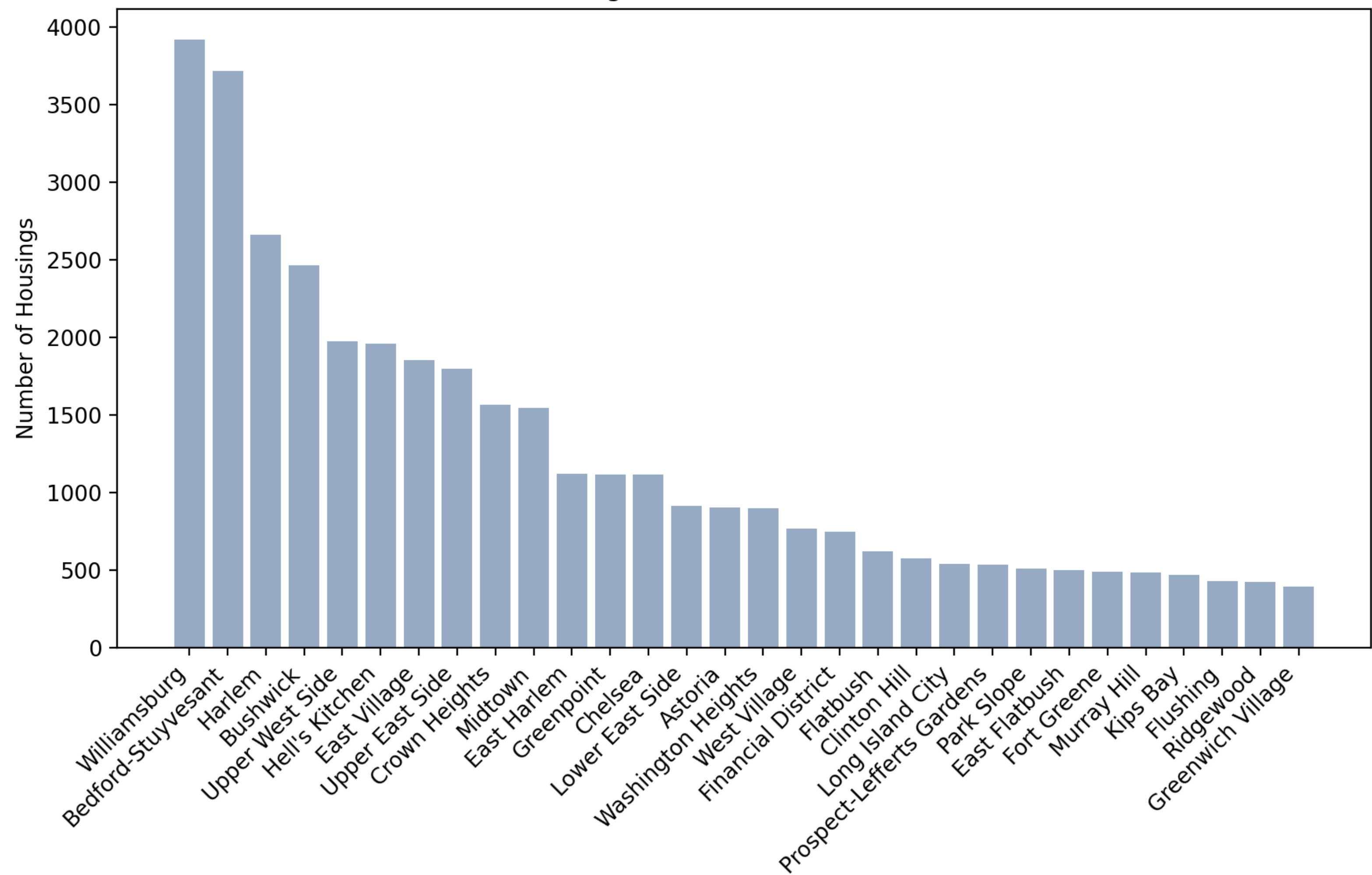
Neighbourhood Group Distribution



Top 30 neighbourhoods which have the most housings:

```
# I'm counting the data in neighbourhood column
neighbourhood_counts = newdf["neighbourhood"].value_counts()
top_neighbourhoods = neighbourhood_counts.head(30)
# Neighbourhood Bar Chart
plt.figure(figsize=(10, 10))
plt.bar(top_neighbourhoods.index, top_neighbourhoods.values, color="#9AAEC4")
plt.title("Neighbourhood Distribution")
plt.ylabel("Number of Housings")
plt.xticks(rotation=45, ha="right")
plt.subplots_adjust(bottom=0.3, left=0.1, right=0.9, top=0.9)
plt.show()
```


Neighbourhood Distribution

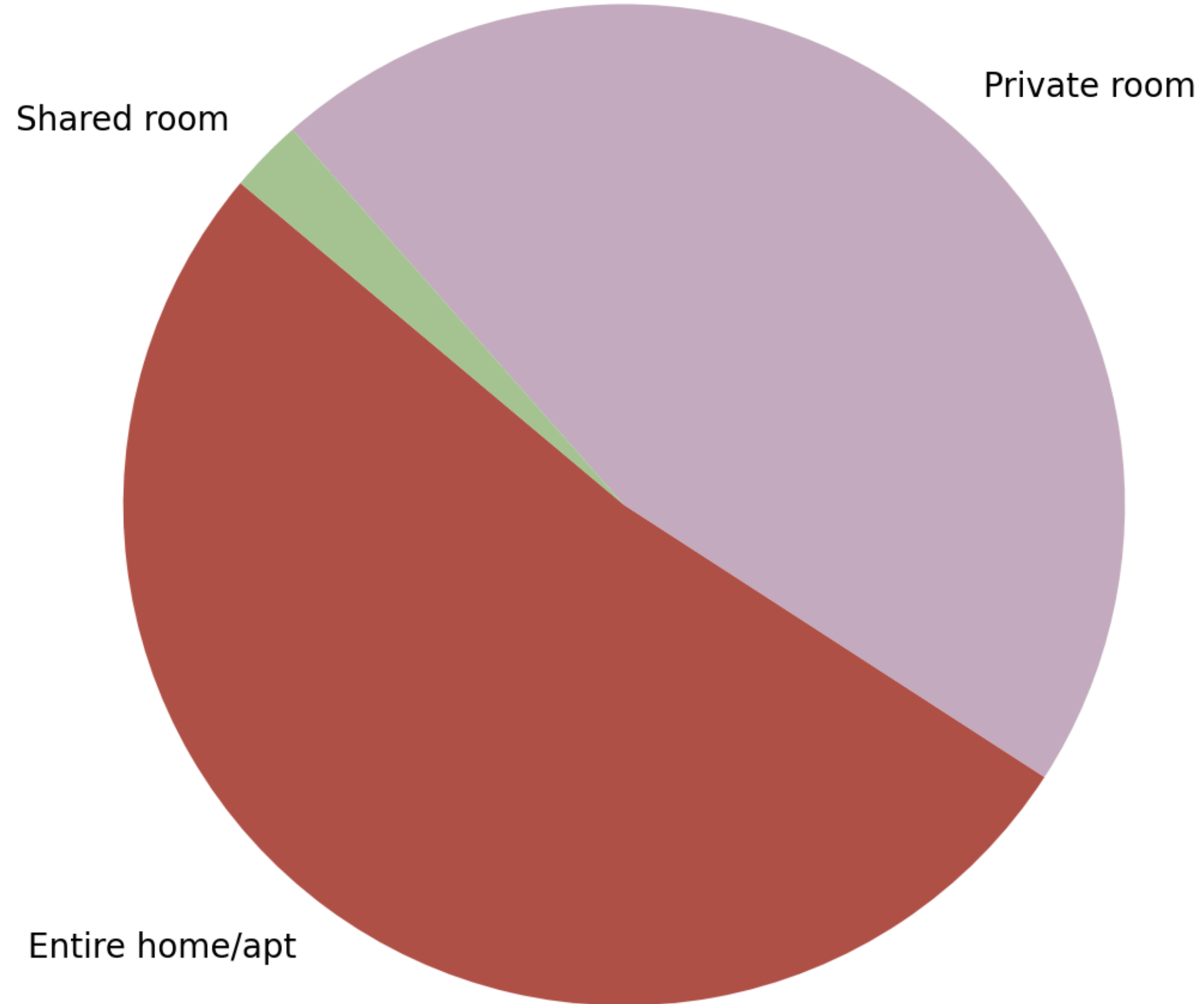


How is the distribution of housings according to room types:

```
# I'm counting the datas in room_type
room_type_counts = newdf["room_type"].value_counts()

# room_type Pie Chart
plt.figure(figsize=(8, 8))
plt.pie(room_type_counts, labels=room_type_counts.index, startangle=140,
        colors=["#B3534A", "#C4ACC0", "#A5C694"])
plt.title("Room Type Distribution")
plt.show()
```

Room Type Distribution



Number of hosts that has more than 2 housings in the system:

```
host_id_counts = newdf['host_id'].value_counts()  
filtered_host_ids = host_id_counts[host_id_counts >= 2]  
print("The number of hosts that who has more than 2 housings in the system:")  
print(filtered_host_ids.count())
```

The number of hosts that who has more than 2 housings in the system:

```
>>> print(filtered_host_ids.count())  
5154  
>>> □
```

Number of hosts that has more than 10 housings in the system:

```
filtered_host_ids = host_id_counts[host_id_counts >= 10]
print("The number of hosts that who has more than 10 housings in the system:")
print(filtered_host_ids.count())
```

The number of hosts that who has more than 10 housings in the system:

```
>>> print(filtered_host_ids.count())
```

```
115
```

```
>>> □
```

Number of hosts that has more than 20 housings in the system:

```
filtered_host_ids = host_id_counts[host_id_counts >= 20]
print("The number of hosts that who has more than 20 housings in the system:")
print(filtered_host_ids.count())
```

```
The number of hosts that who has more than 20 housings in the system:
>>> print(filtered_host_ids.count())
40
>>> □
```

The host who has the most housings in the system:

```
print("The host who has the most housings in the system:")  
print(host_id_counts.head(1))
```

```
The host who has the most housings in the system:  
>>> print(host_id_counts.head(1))  
219517861      327
```

The mean and median values of Manhattan:

```
#I want to see average prices for Manhattan
manhattan_data = newdf[newdf["neighbourhood_group"] == "Manhattan"]
#Calculating mean of the price of Manhattan
mean_price = manhattan_data["price"].mean()
print(f"The mean of the price of Manhattan: ${mean_price:.2f}")
#Calculating median of the price of Manhattan
median_price = manhattan_data["price"].median()
print(f"The median of the price of Manhattan: ${median_price:.2f}")
```

```
The mean of the price of Manhattan: $196.88
>>> print(f"The median of the price of Manhattan: ${median_price:.2f}")
The median of the price of Manhattan: $150.00
>>> []
```


The mean and median values of Brooklyn:

```
#I want to see average prices for Brooklyn
brooklyn_data = newdf[newdf["neighbourhood_group"] == "Brooklyn"]
#Calculating mean of the price of Brooklyn
mean_price = brooklyn_data["price"].mean()
print(f"The mean of the price of Brooklyn: ${mean_price:.2f}")
#Calculating median of the price of Brooklyn
median_price = brooklyn_data["price"].median()
print(f"The median of the price of Brooklyn: ${median_price:.2f}")
```

```
The mean of the price of Brooklyn: $124.38
>>> print(f"The median of the price of Brooklyn: ${median_price:.2f}")
The median of the price of Brooklyn: $90.00
>>> []
```

The mean and median values of Queens:

```
#I want to see average prices for Queens
queens_data = newdf[newdf["neighbourhood_group"] == "Queens"]
#Calculating mean of the price of Queens
mean_price = queens_data["price"].mean()
print(f"The mean of the price of Queens: ${mean_price:.2f}")
#Calculating median of the price of Queens
median_price = queens_data["price"].median()
print(f"The median of the price of Queens: ${median_price:.2f}")
```

```
The mean of the price of Queens: $99.52
>>> print(f"The median of the price of Queens: ${median_price:.2f}")
The median of the price of Queens: $75.00
>>> □
```

The mean and median values of Bronx:

```
#I want to see average prices for Bronx
bronx_data = newdf[newdf["neighbourhood_group"] == "Bronx"]
#Calculating mean of the price of Bronx
mean_price = bronx_data["price"].mean()
print(f"The mean of the price of Bronx: ${mean_price:.2f}")
#Calculating median of the price of Bronx
median_price = bronx_data["price"].median()
print(f"The median of the price of Bronx: ${median_price:.2f}")
```

The mean of the price of Bronx: \$87.50

```
>>> print(f"The median of the price of Bronx: ${median_price:.2f}")
```

The median of the price of Bronx: \$65.00

```
>>> []
```

The mean and median values of Staten Island:

```
#I want to see average prices for Staten Island
stateniland_data = newdf[newdf["neighbourhood_group"] == "Staten Island"]
#Calculating mean of the price of Staten Island
mean_price = stateniland_data["price"].mean()
print(f"The mean of the price of Staten Island: ${mean_price:.2f}")
#Calculating median of the price of Staten Island
median_price = stateniland_data["price"].median()
print(f"The median of the price of Staten Island: ${median_price:.2f}")
```

```
The mean of the price of Staten Island: $114.81
>>> print(f"The median of the price of Staten Island: ${median_price:.2f}")
The median of the price of Staten Island: $75.00
>>> □
```

```
#I want to see avarage number of reviews for Manhattan  
avg_review_manhattan= manhattan_data["number_of_reviews"].mean()  
print(f"Mean of number of reviews for Manhattan: {avg_review_manhattan:.1f}")
```

```
#I want to see avarage number of reviews for Brooklyn  
avg_review_brooklyn= brooklyn_data["number_of_reviews"].mean()  
print(f"Mean of number of reviews for Brooklyn: {avg_review_brooklyn:.1f}")
```

```
#I want to see avarage number of reviews for Queens  
avg_review_queens= queens_data["number_of_reviews"].mean()  
print(f"Mean of number of reviews for Queens: {avg_review_queens:.1f}")
```

```
#I want to see avarage number of reviews for Bronx  
avg_review_bronx= bronx_data["number_of_reviews"].mean()  
print(f"Mean of number of reviews for Bronx: {avg_review_bronx:.1f}")
```

```
#I want to see avarage number of reviews for Staten Island  
avg_review_statenisland= statenisland_data["number_of_reviews"].mean()  
print(f"Mean of number of reviews for Staten Island: {avg_review_statenisland:.1f}")
```

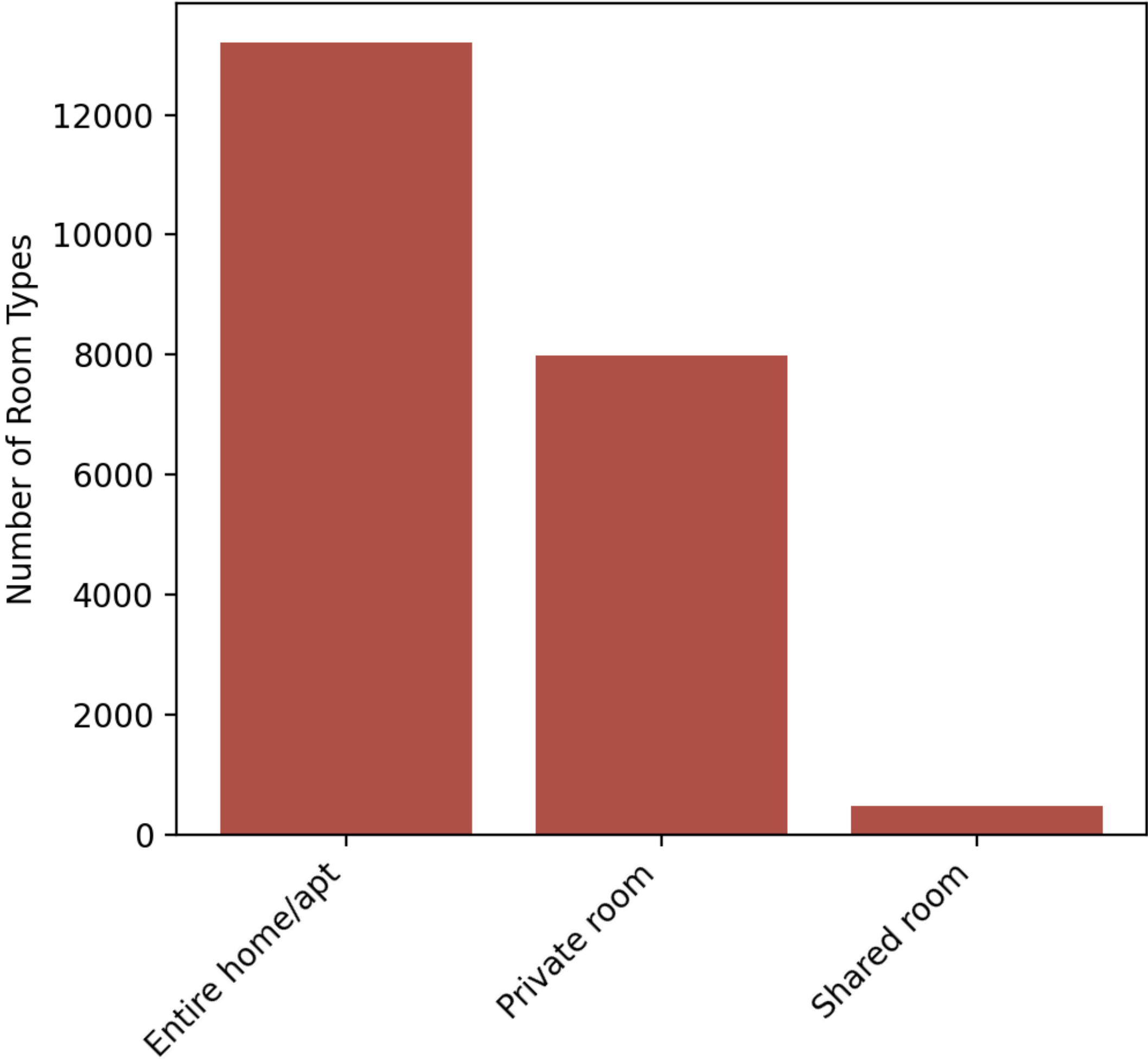
The output of the previous page:

```
Mean of number of reviews for Manhattan: 21.0  
Mean of number of reviews for Brooklyn: 24.2  
Mean of number of reviews for Queens: 27.7  
Mean of number of reviews for Bronx: 26.0  
>>> print(f"Mean of number of reviews for Staten Island: {avg_review_statenisland:.1f}")  
Mean of number of reviews for Staten Island: 30.9  
>>> []
```

The Distrubition of Room Types In Manhattan:

```
roomtype_counts = manhattan_data["room_type"].value_counts()  
# Room Types In Manhattan Bar Chart  
plt.figure(figsize=(6, 6))  
plt.bar(roomtype_counts.index, roomtype_counts.values, color="#B3534A")  
plt.title("Room Type Distribution In Manhattan")  
plt.ylabel("Number of Room Types")  
plt.xticks(rotation=45, ha="right")  
plt.subplots_adjust(bottom=0.3, left=0.2, right=0.9, top=0.9)  
plt.show()
```

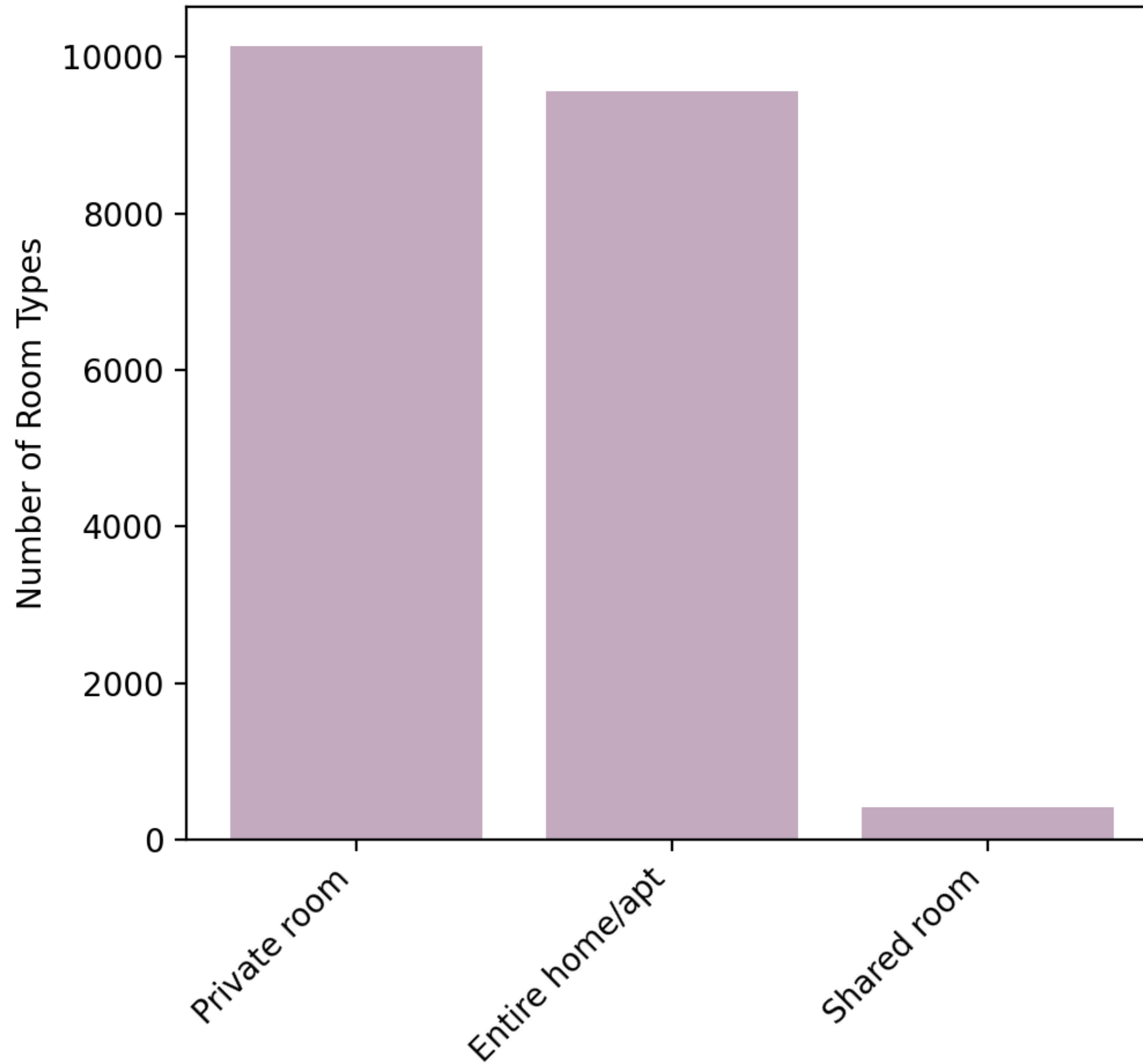
Room Type Distribution In Manhattan



The Distrubition of Room Types In Brooklyn:

```
roomtype_counts = brooklyn_data["room_type"].value_counts()  
# Room Types In Brooklyn Bar Chart  
plt.figure(figsize=(6, 6))  
plt.bar(roomtype_counts.index, roomtype_counts.values, color="#C4ACC0")  
plt.title("Room Type Distribution In Brooklyn")  
plt.ylabel("Number of Room Types")  
plt.xticks(rotation=45, ha="right")  
plt.subplots_adjust(bottom=0.3, left=0.2, right=0.9, top=0.9)  
plt.show()
```

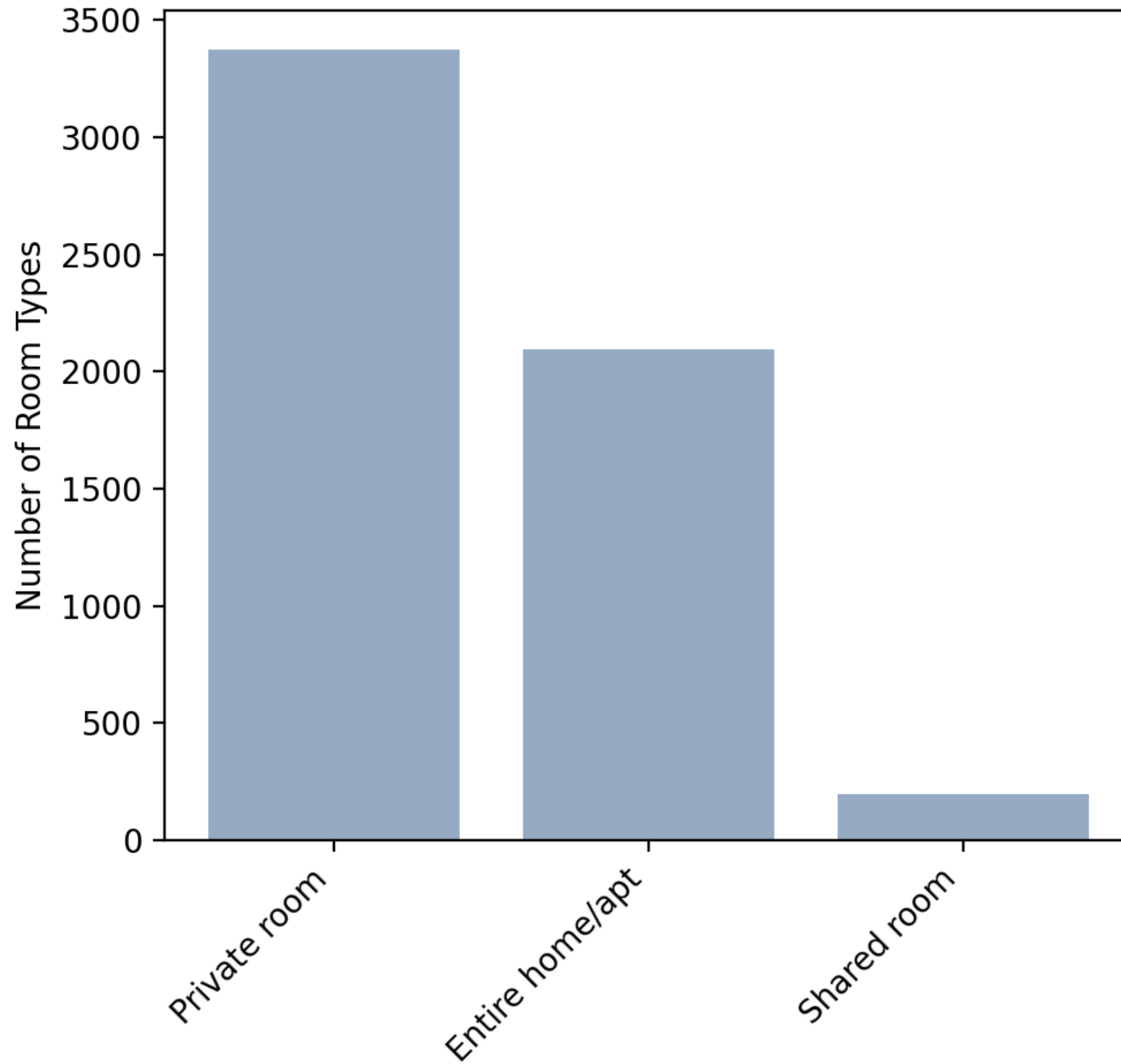
Room Type Distribution In Brooklyn



The Distrubition of Room Types In Queens:

```
roomtype_counts = queens_data["room_type"].value_counts()
# Room Types In Manhattan Bar Chart
plt.figure(figsize=(6, 6))
plt.bar(roomtype_counts.index, roomtype_counts.values, color="#9AAEC4")
plt.title("Room Type Distribution In Queens")
plt.ylabel("Number of Room Types")
plt.xticks(rotation=45, ha="right")
plt.subplots_adjust(bottom=0.3, left=0.2, right=0.9, top=0.9)
plt.show()
```

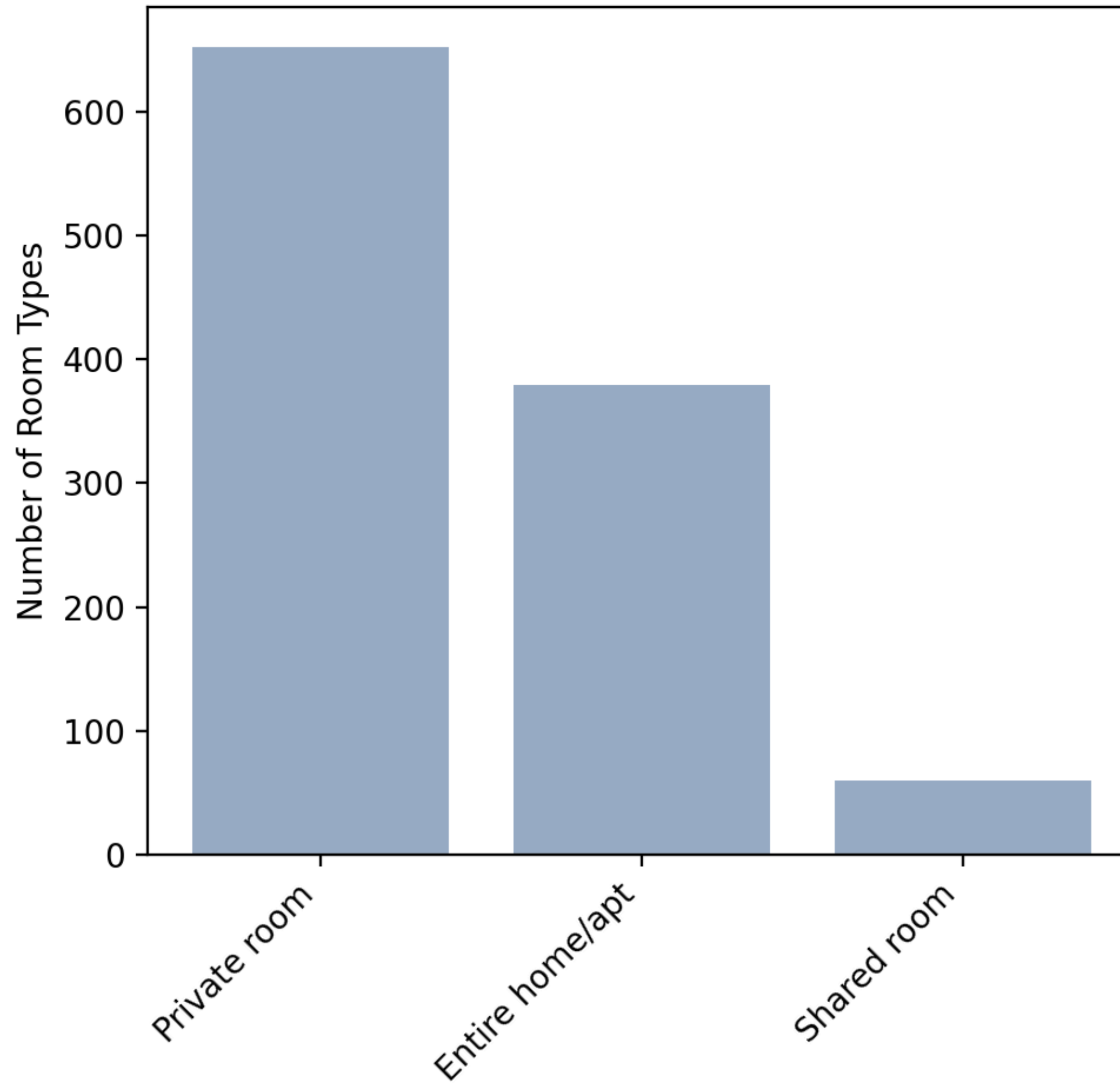
Room Type Distribution In Queens



The Distrubition of Room Types In Bronx:

```
roomtype_counts = bronx_data["room_type"].value_counts()
# Room Types In Bronx Bar Chart
plt.figure(figsize=(6, 6))
plt.bar(roomtype_counts.index, roomtype_counts.values, color="#9AAEC4")
plt.title("Room Type Distribution In Bronx")
plt.ylabel("Number of Room Types")
plt.xticks(rotation=45, ha="right")
plt.subplots_adjust(bottom=0.3, left=0.2, right=0.9, top=0.9)
plt.show()
```

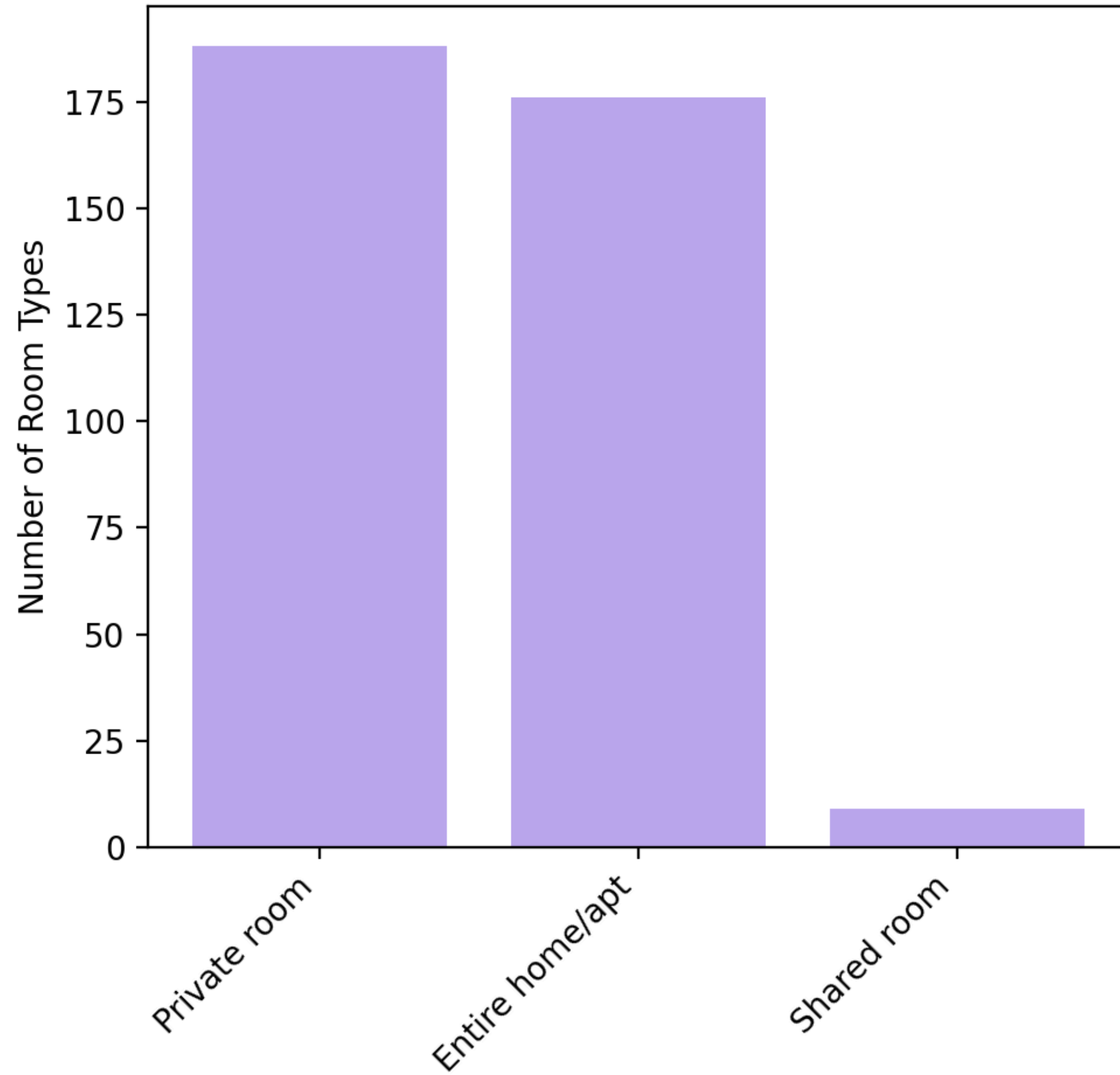
Room Type Distribution In Bronx



The Distrubition of Room Types In Staten Island:

```
roomtype_counts = statenisland_data["room_type"].value_counts()
# Room Types In Staten Island Bar Chart
plt.figure(figsize=(6, 6))
plt.bar(roomtype_counts.index, roomtype_counts.values, color="#BAA8EF")
plt.title("Room Type Distribution In Staten Island")
plt.ylabel("Number of Room Types")
plt.xticks(rotation=45, ha="right")
plt.subplots_adjust(bottom=0.3, left=0.2, right=0.9, top=0.9)
plt.show()
```

Room Type Distribution In Staten Island





Thank You!

Ezgi Efe