Airbnb Exploratory Data Analysis

Santhosh Kumar

2024-06-22

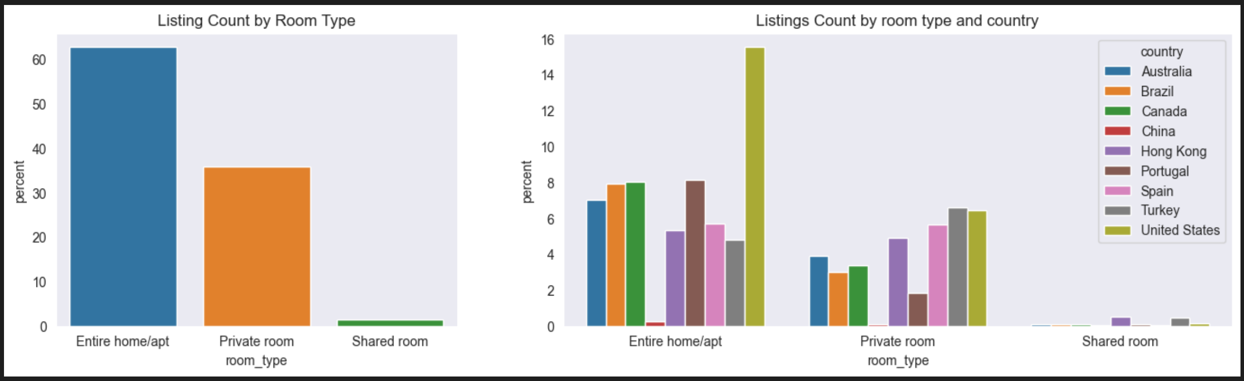
Table of contents

List of Figures

List of Tables

0.1 Room Type and countries

```
fig, axs = plt.subplots(1, 2, figsize=(16, 4), gridspec_kw={'width_ratios': [6, 10]})
axs[0].set_title('Listing Count by Room Type')
sns.countplot(x=df1['room_type'],hue=df1.room_type,stat="percent", ax=axs[0])
sns.countplot(x=df1['room_type'],hue=df1.country,stat="percent", ax=axs[1])
axs[1].set_title('Listings Count by room type and country')
plt.show()
```

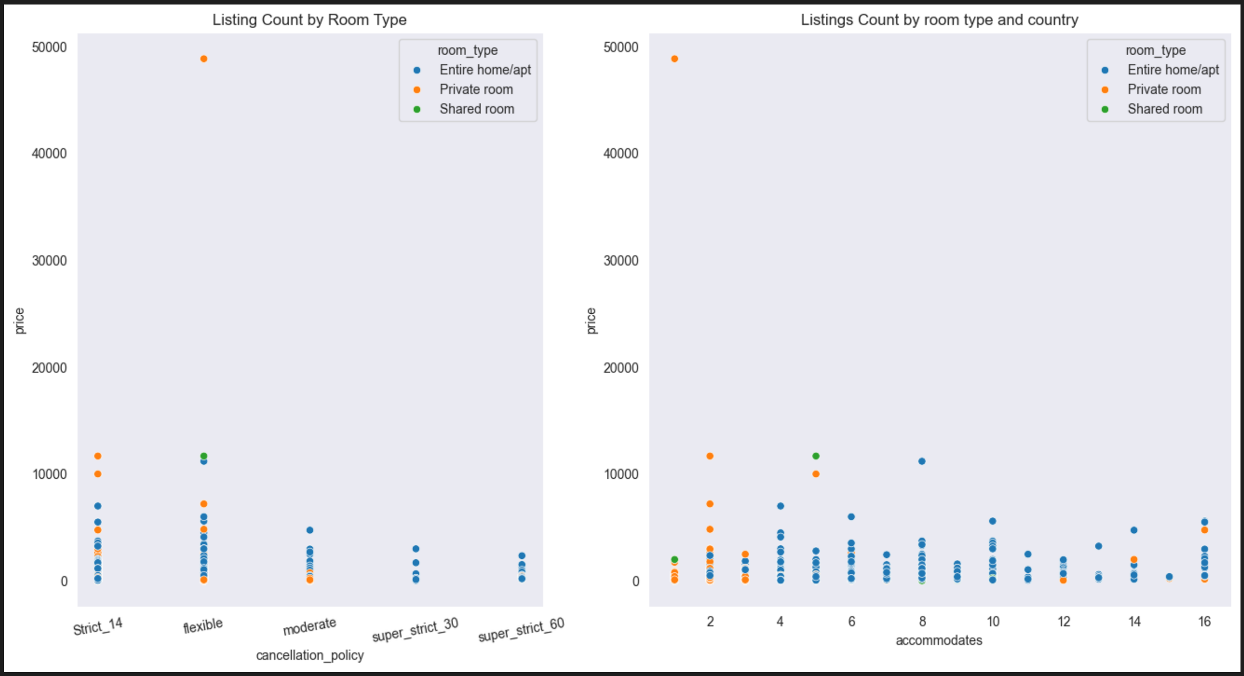


1. Most of listings are Entire Home/Apartment follwed by Provate Room and Shared Room
2. United States holds 50% higher listings in Entire Home/Apartment Room Type comparing other countries
3. All countries listings are more or less equal in Private Room
4. China is less contributor when comparing all countries

0.2 Listing price in terms cancellation policy, accomadates with Room Type

```
fig, axs = plt.subplots(1, 2, figsize=(16,8), gridspec_kw={'width_ratios': [8, 10]})
axs[0].set_title('Listing Count by Room Type')
sns.scatterplot(data=df1,x='cancellation_policy',y='price',hue='room_type',ax=axs[0])
```

```
axs[0].tick_params(axis='x', rotation=10)
sns.scatterplot(data=df1, x="accommodates", y="price", hue="room_type", ax=axs[1])
axs[1].set_title('Listings Count by room type and country')
plt.show()
```



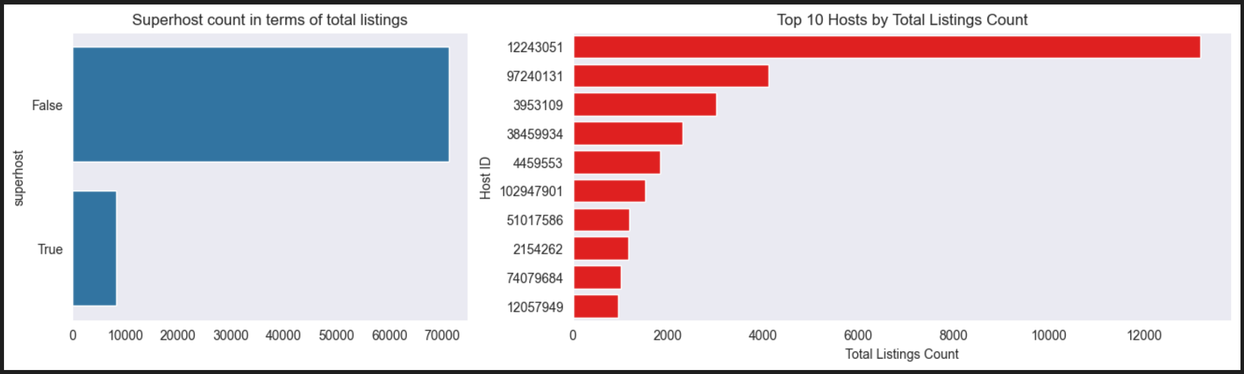
1. From the left plot, we can see that the most of the listings are booked under Strict 14 cancellation policy type
2. From th right side plot, we can see that one accomadate listings are mostly Private rooms
3. From th right side plot, we can see that the most of the listings are booked with 2 accomadates.

0.3 Top 10 Hosts and Super Host interms of listing count

```
fig, axs = plt.subplots(1, 2, figsize=(16, 4), gridspec_kw={'width_ratios': [6, 10]})
axs[0].set_title('Superhost count in terms of total listings')
top_superhost = df1.groupby('superhost')['host_total_listings_count'].sum()
sns.barplot(x=top_superhost.values, y=top_superhost.index, orient='h',ax=axs[0])

top_host = df1.groupby('host_id')['host_total_listings_count'].sum().nlargest(10)
top_host_superhost = df1[df1['host_id'].isin(top_host.index)]['superhost']
colors = ['green' if sh else 'red' for sh in top_host_superhost]
sns.barplot(x=top_host.values, y=top_host.index, orient='h',hue=top_host.index,palette=colors,ax=axs[1]
plt.xlabel('Total Listings Count')
plt.ylabel('Host ID')
axs[1].set_title('Top 10 Hosts by Total Listings Count')
plt.show()
```

```
C:\Users\sansu\AppData\Local\Temp\ipykernel_17204\575269493.py:9: UserWarning: The palette list has more v
```

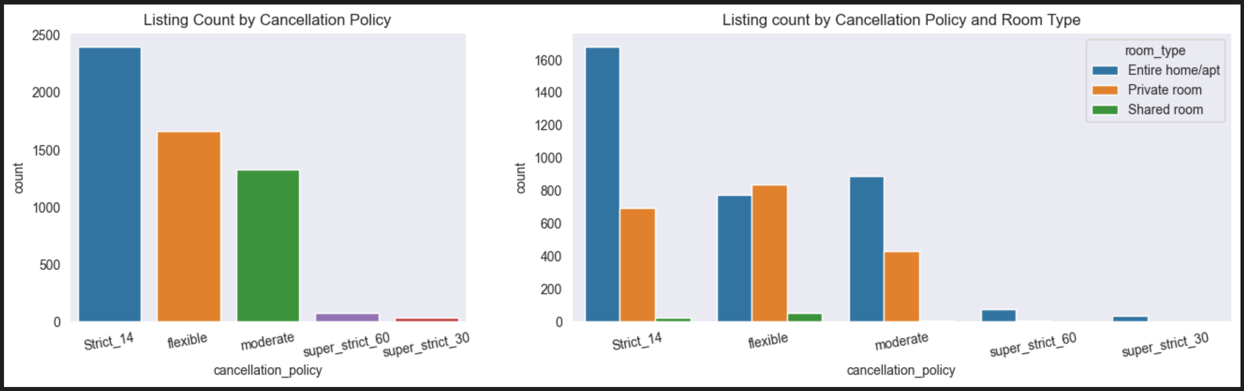
```
sns.barplot(x=top_host.values, y=top_host.index, orient='h',hue=top_host.index,palette=colors,ax=axs[1])
```



1. 80% of listings done by normal host
2. Top hosts in terms of total listing is also normal hosts

0.4 Cancellation Policy and Listings

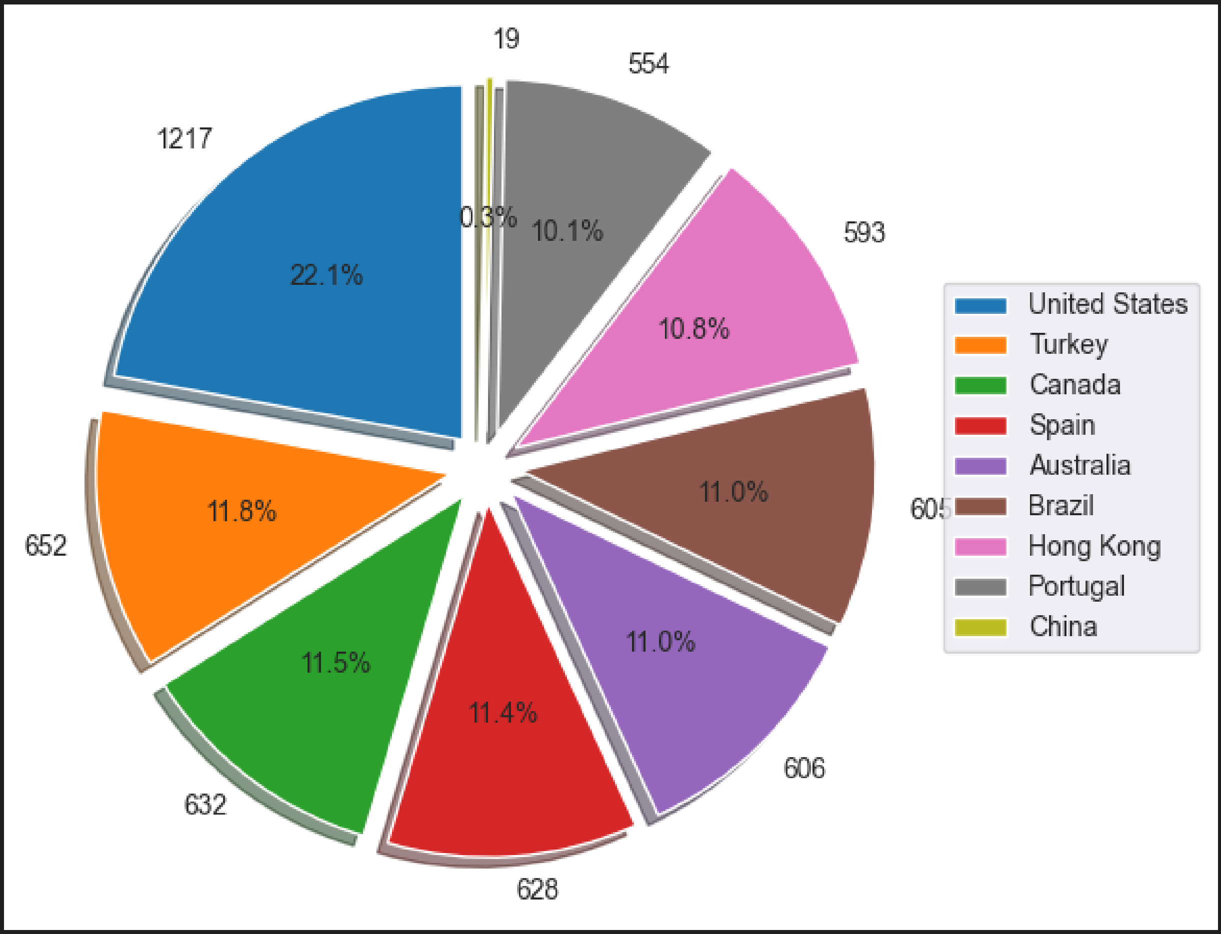
```
import seaborn as sns
import matplotlib.pyplot as plt

cancellation_order = df1['cancellation_policy'].value_counts().index

# Create subplots
fig, axs = plt.subplots(1, 2, figsize=(16, 4), gridspec_kw={'width_ratios': [6, 10]})

# First subplot
axs[0].set_title('Listing Count by Cancellation Policy')
sns.countplot(x=df1['cancellation_policy'], hue=df1.cancellation_policy, order=cancellation_order, ax=ax
axs[0].tick_params(axis='x', rotation=10)  # Rotate x-axis labels

# Second subplot
axs[1].set_title('Listing count by Cancellation Policy and Room Type')
sns.countplot(x=df1['cancellation_policy'], hue=df1.room_type, order=cancellation_order, ax=axs[1])
axs[1].tick_params(axis='x', rotation=10)  # Rotate x-axis labels

plt.show()
```



1. Most listings are from strict 14 cancellation policy
2. Most of the Private rooms are booked under flexible cancellation policy
3. 30 days cancellation is lesser than comparing all.

0.5 Top listings in terms of country

```
plt.figure(figsize=(15,6))
shape = df1.country.value_counts().values
labels = df1.country.value_counts().index
plt.pie(x=shape,labels=shape,explode=[.1] * len(shape),shadow=True, autopct = '%1.1f%%', startangle=90)
plt.legend(labels,loc="center left", bbox_to_anchor=(1, 0.5))
plt.show()
```

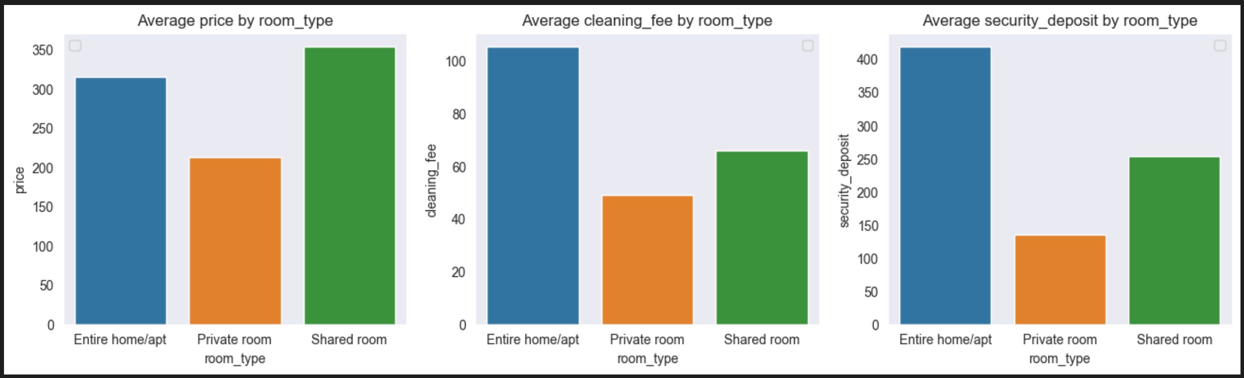


1. US is listed 21% of properties and holds predominant place when compare to other countries.
2. Followed by US, all other countries listed almost equally except China as it holds 0.3%.

0.6 Average price, Security Deposit, Cleasing Fee for each room type

```
df_price_room_type = pd.DataFrame(df1.groupby(['room_type'],observed=False).agg({'price': 'mean', 'clea
plt.figure(figsize=(16,4))
plt.subplot(131)
plt.title('Average price by room_type')
sns.barplot(data=df_price_room_type,x='room_type',y='price',hue='room_type')
plt.legend('')
plt.subplot(132)
plt.title('Average cleaning_fee by room_type')
sns.barplot(data=df_price_room_type,x='room_type',y='cleaning_fee',hue='room_type')
plt.legend('')
plt.subplot(133)
plt.title('Average security_deposit by room_type')
```

```
sns.barplot(data=df_price_room_type,x='room_type',y='security_deposit',hue='room_type')
plt.legend('')
plt.show()
```



1. Eventhough shared room has less numbers when compared to others it's average price, cleaning fee are high but security deposit is less.
2. Average Security Deposit it almost equal to Average price. We may also consider that not all the properties requesting Security Deposit. The average is derrived from the listings that are requested Security Deposit.
3. Entire home/apt holds predominant place which seems obvious. It is higher count.

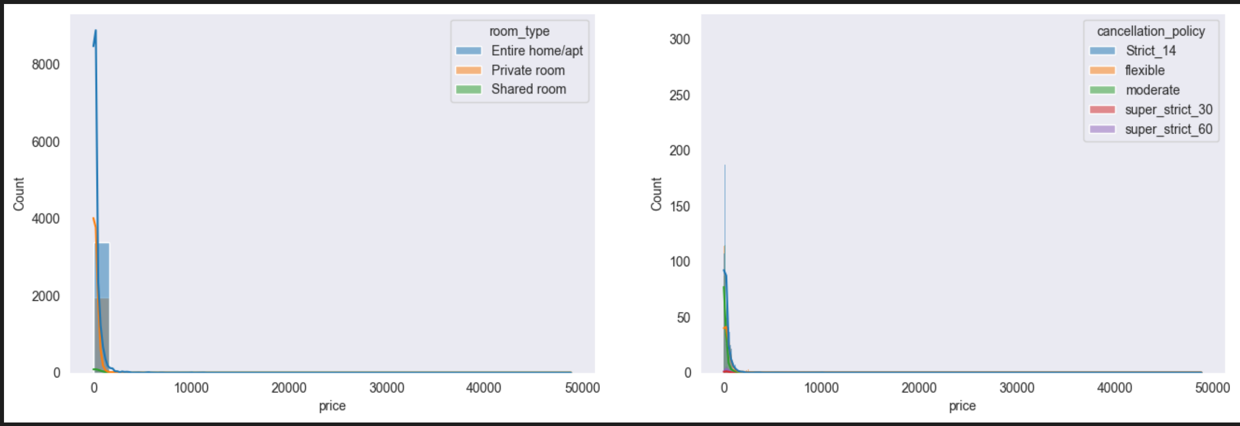
0.7 Comparison of Price Distributions and Density of Price Ranges for Each Room Type

```
plt.figure(figsize=(14,6))
sns.kdeplot(df1[df1.room_type=='Entire home/apt'].price,color='maroon',label='Entire home/apt',linewidth
sns.kdeplot(df1[df1.room_type=='Private room'].price,color='black',label='Private room',linewidth=2.5)
sns.kdeplot(df1[df1.room_type=='Shared room'].price,color='green',label='Shared room',linewidth=2.5)
plt.title('Room Type VS price distribution for price')
plt.xlim(0,800)
plt.show()
```

0.8 Comparison of Price Distributions and Density of Price Ranges for Each top 5 region

0.9 Price distribution in terms of room type and Cancellation poli

```
fig, axs = plt.subplots(1, 2, figsize=(16, 5))
plt.subplot(121)
sns.histplot(data=df1,x='price',bins=30,kde=True,hue='room_type')
plt.subplot(122)
sns.histplot(data=df1,x='price',kde=True,hue='cancellation_policy')
plt.show()
```

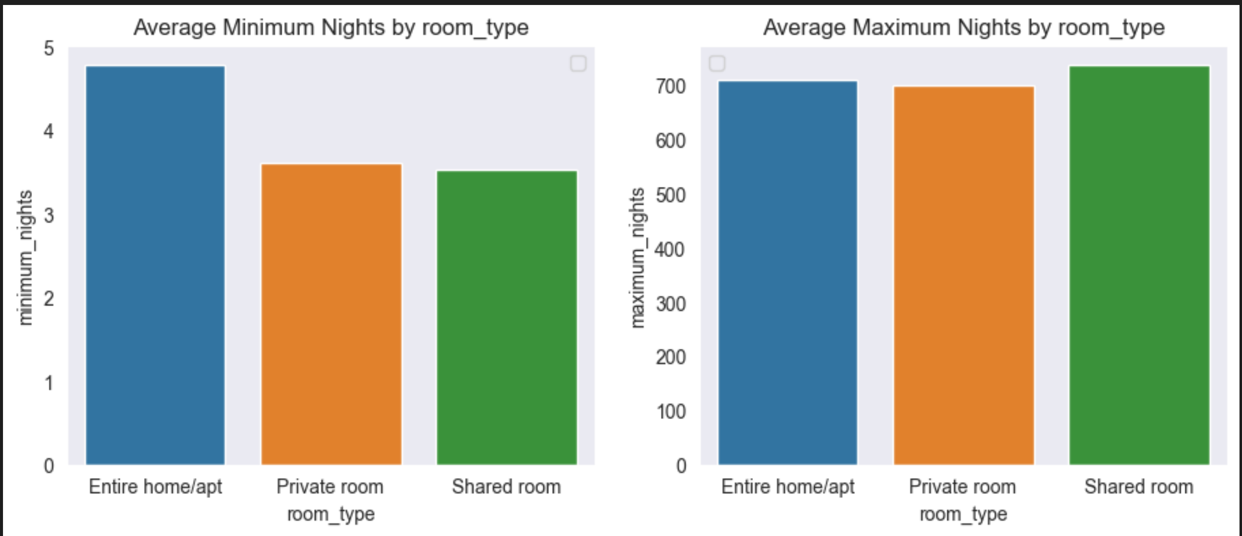


1. Entire home/apt tend to high proce followed by Private Room. Shared room very less comparitively

Entire home and Private room.
2. Strict 14 and Flexible cancellation type properties are listed most frequestly.

0.10 Minimum & maximun Nights by room type

```
min_max_ni8 = pd.DataFrame(df1.groupby(['room_type'],observed=False).agg({'minimum_nights': 'mean','maxi
plt.figure(figsize=(11,4))
plt.subplot(121)
plt.title('Average Minimum Nights by room_type')
sns.barplot(data=min_max_ni8,x='room_type',y='minimum_nights',hue='room_type')
plt.legend('')
plt.subplot(122)
plt.title('Average Maximum Nights by room_type')
sns.barplot(data=min_max_ni8,x='room_type',y='maximum_nights',hue='room_type')
plt.legend('')
plt.show()
```

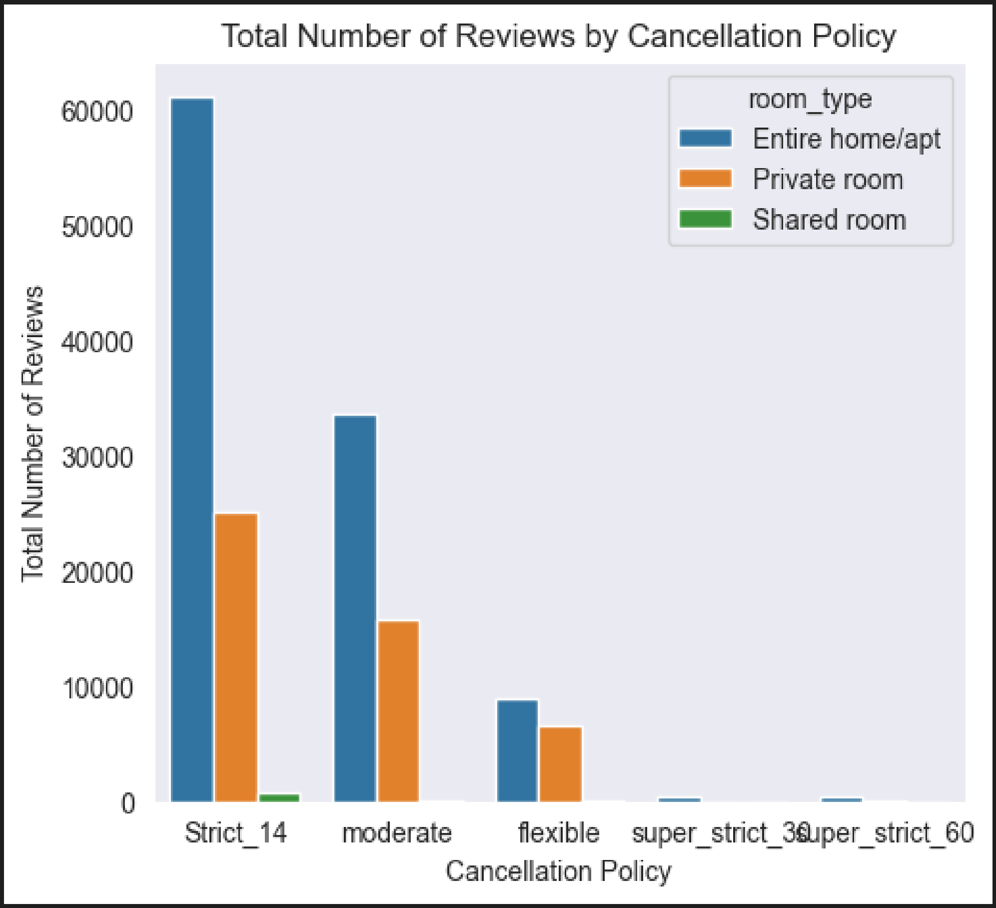


1. Average Minumum Night for Entire home/apt is aroun 5 days and Maximun night is around 700 days
2. Rememeber, as we discussed about these variables in Outlier Treatment, the minimum and Maximum stays may vary depends on several criteria and property type.

0.11 Reviews count interms of Cancellation policy and Room type

```
min_max_ni8 = pd.DataFrame(df1.groupby(['cancellation_policy','room_type'], observed=False)['number_of_r
min_max_ni8 = min_max_ni8.sort_values(by='number_of_reviews', ascending=False)

plt.figure(figsize=(12, 5))

plt.subplot(121)
sns.barplot(data=min_max_ni8, x='cancellation_policy', y='number_of_reviews', order=min_max_ni8['cancell
```

```
plt.title('Total Number of Reviews by Cancellation Policy')
plt.xlabel('Cancellation Policy')
plt.ylabel('Total Number of Reviews')
plt.show()
```

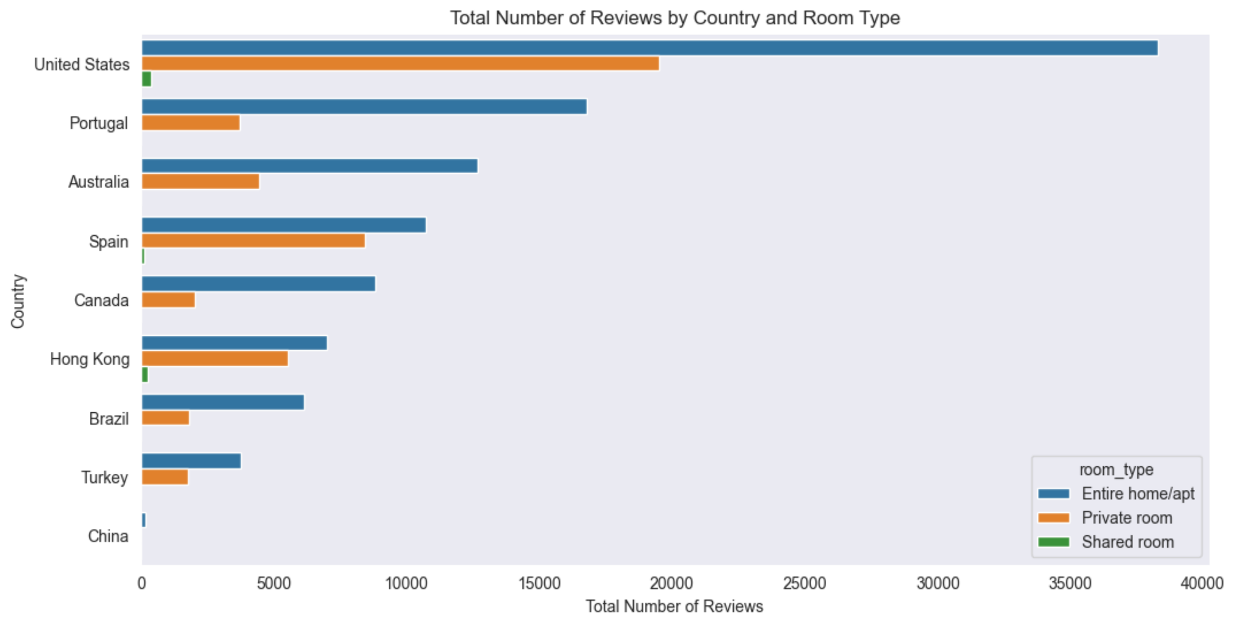


1. Strict 14 holds predominant number of reviews
2. Entire home/apt has more number of reviews in all aspects

0.12 Reviews count interms of Country and Room type

```
Region = pd.DataFrame(df1.groupby(['country','room_type'], observed=False)['number_of_reviews'].sum()).r
Region = Region.sort_values(by='number_of_reviews', ascending=False)

plt.figure(figsize=(12, 6))

sns.barplot(data=Region, y='country', x='number_of_reviews', order=Region['country'], orient='h',hue='ro
plt.title('Total Number of Reviews by Country and Room Type')
plt.xlabel('Total Number of Reviews')
plt.ylabel('Country')
```

```
plt.show()
```



1. US holds predominant number of reviews
2. Entire home/apt has more number of reviews in all aspects

0.13 Spatial Distribution of Listings in terms of top 10 Regions

```
import folium


top_map = df1.groupby('region', observed=False).agg({'latitude': 'first', 'longitude': 'first', 'price'
top_5_map=top_map.sort_values(by='price')

maps = []
# Iterate through each region
for index, row in top_5_map.iterrows():
    region_listings = df1[df1['region'] == row['region']]
    maps.append(row['region'])
    # Create a map centered around the mean latitude and longitude of the region
    m = folium.Map(location=[row['latitude'], row['longitude']], zoom_start=9)


    # Filter the DataFrame to get listings in the current region


    # Add scatter markers for each listing in the region
    for _, listing in region_listings.iterrows():
```

```python
            folium.Marker(location=[listing['latitude'], listing['longitude']],popup=row['region']).add_to(

        # Add the map to the list of maps
        maps.append(m)

    # Display all the maps in the list
    for m in maps:
        display(m)
```

```
'Sydney'

<folium.folium.Map at 0x1dc233b8510>

'Jordan'

<folium.folium.Map at 0x1dc2a130210>

'Aveiro District'

<folium.folium.Map at 0x1dc265a8ad0>

'Guangdong Province'

<folium.folium.Map at 0x1dc268b0090>

'Grande Porto'

<folium.folium.Map at 0x1dc265abc10>

'New Territories'

<folium.folium.Map at 0x1dc2b20a850>

'Catalonia'

<folium.folium.Map at 0x1dc45847590>

'Quebec'

<folium.folium.Map at 0x1dc22ccfb90>

'New York'

<folium.folium.Map at 0x1dc2abe1950>

'New South Wales'

<folium.folium.Map at 0x1dc29b26e50>

'Hawaii'

<folium.folium.Map at 0x1dc2f0a17d0>

'Hong Kong Island'

<folium.folium.Map at 0x1dc269059d0>

'Hong Kong'

<folium.folium.Map at 0x1dc31fedf50>

'Istanbul'
```

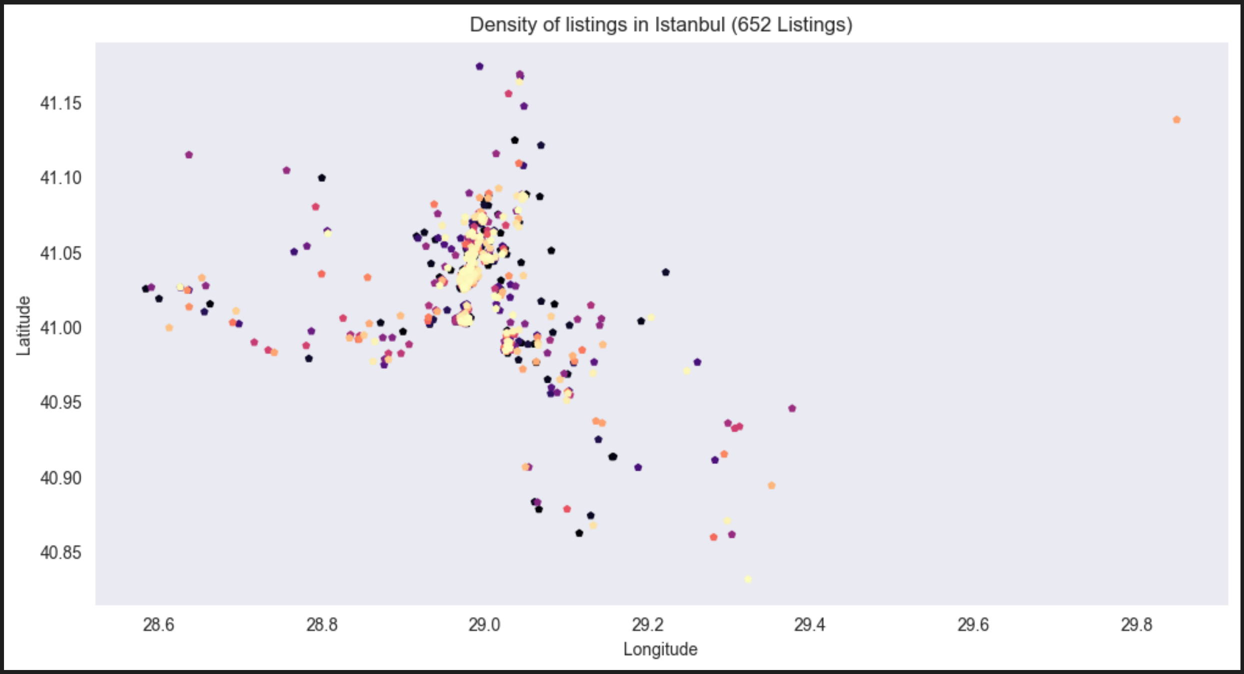
<folium.folium.Map at 0x1dc38e997d0>

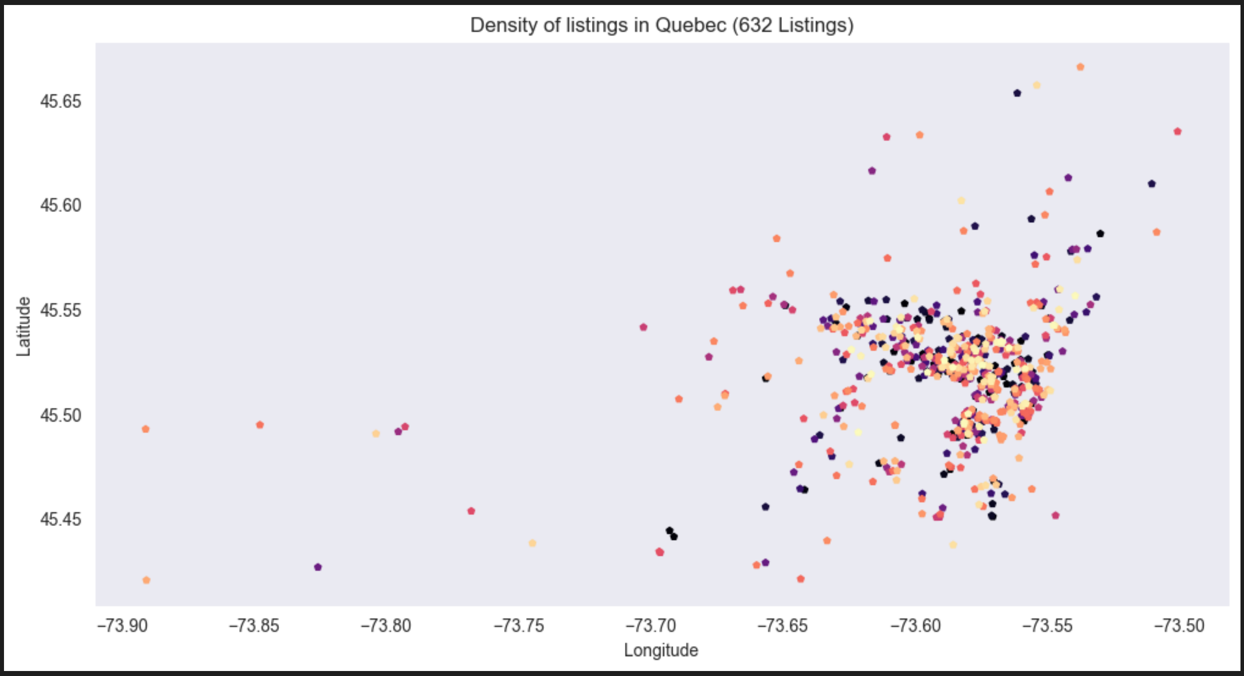
'Rio De Janeiro'

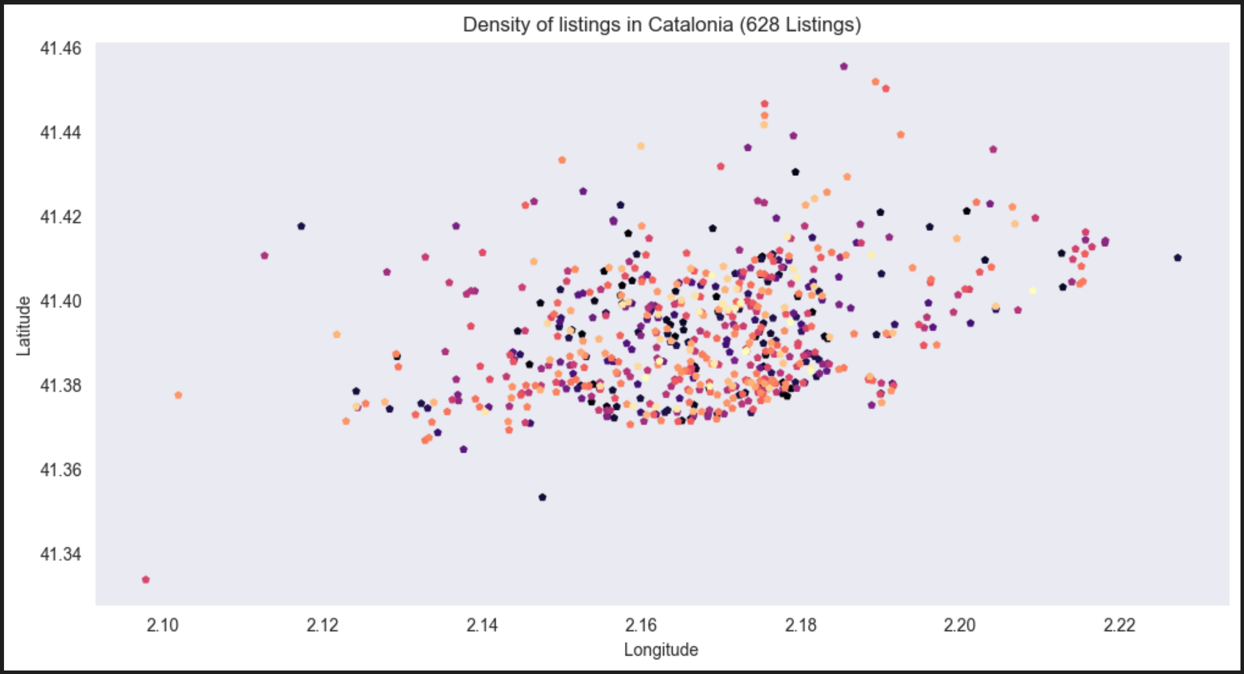
<folium.folium.Map at 0x1dc3950e910>

From above plot i could see that the 99% listings are nearby Beach, Lake, and River.

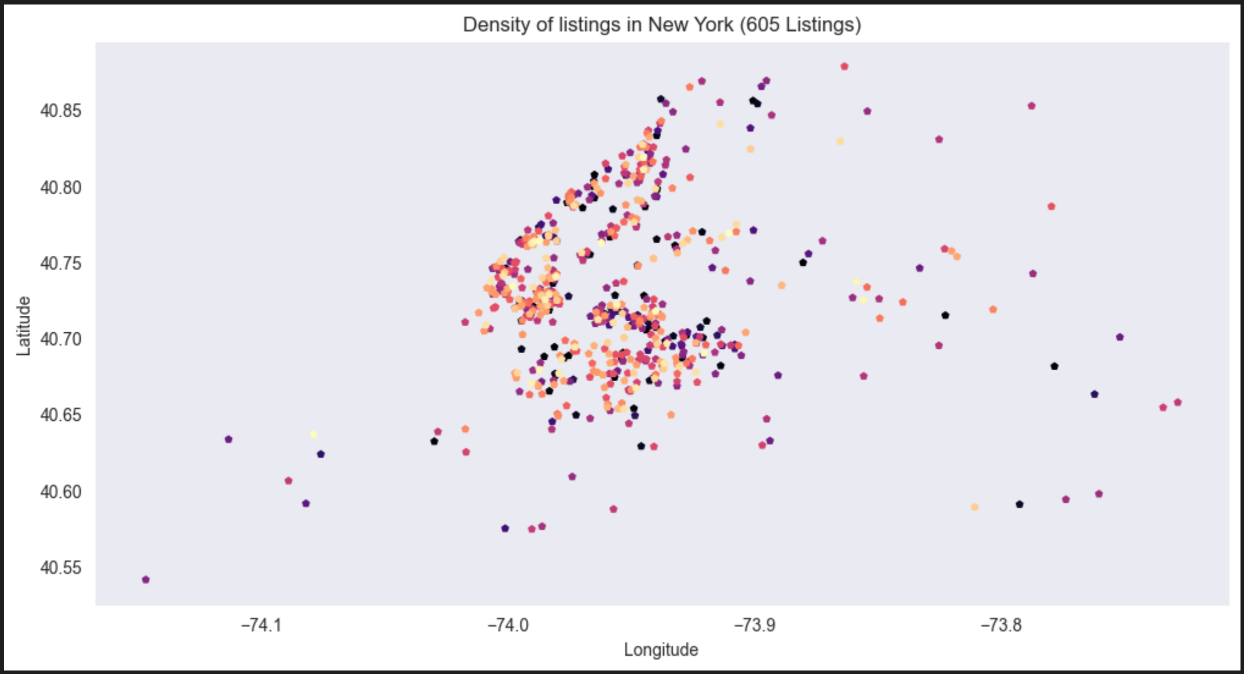
```
df_top_region = df1.groupby('region',observed=False).size().reset_index(name='size')
df_top_region=df_top_region.sort_values(by='size',ascending=False).nlargest(10,'size')

for index, row in df_top_region.iterrows():
    region = row['region']
    dff = df1[df1['region'] == region]  # Filter DataFrame for the current region
    lat = dff['latitude']
    lon = dff['longitude']

    xy = np.vstack([lat, lon])
    z = gaussian_kde(xy)(xy)
    idx = z.argsort()
    z = z[idx]

    plt.figure(figsize=(12, 6))
    plt.scatter(lon, lat, c=z, s=15,cmap='magma',marker="p")
    plt.xlabel("Longitude")
    plt.ylabel("Latitude")
    plt.title(f"Density of listings in {region} ({row.values[1]} Listings)")
    print()
    plt.show()
```

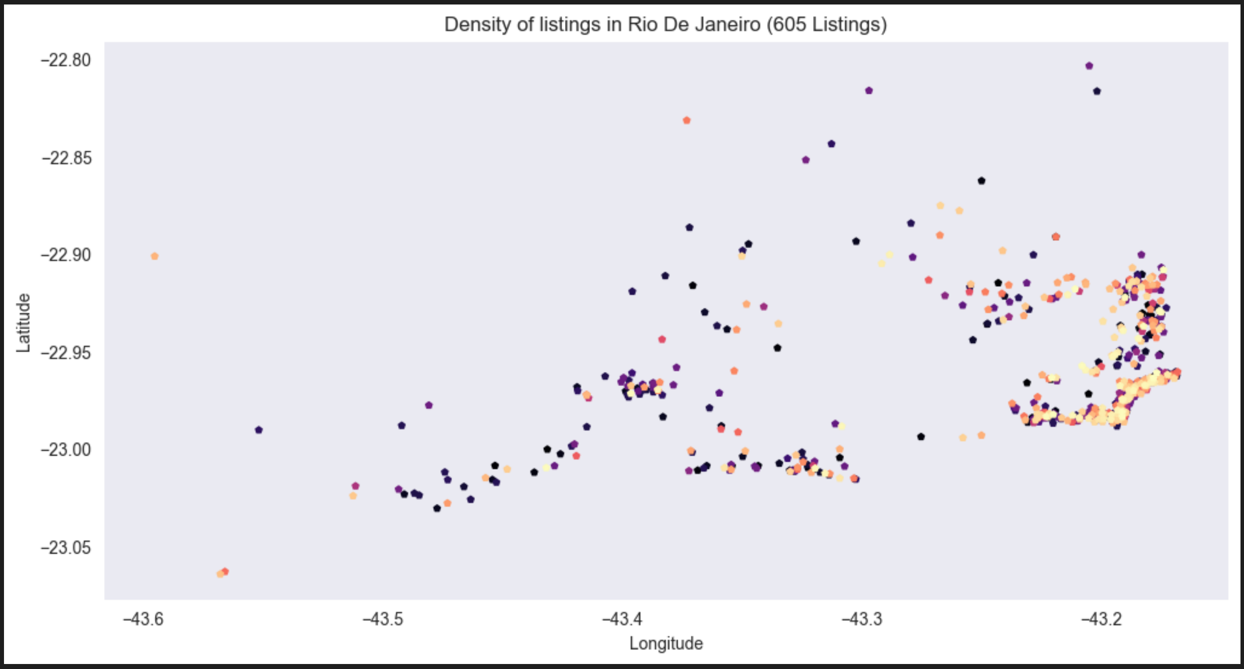

Density of listings in Istanbul (652 Listings)
Latitude
Longitude
41.15
41.10
41.05
41.00
40.95
40.90
40.85
28.6
28.8
29.0
29.2
29.4
29.6
29.8

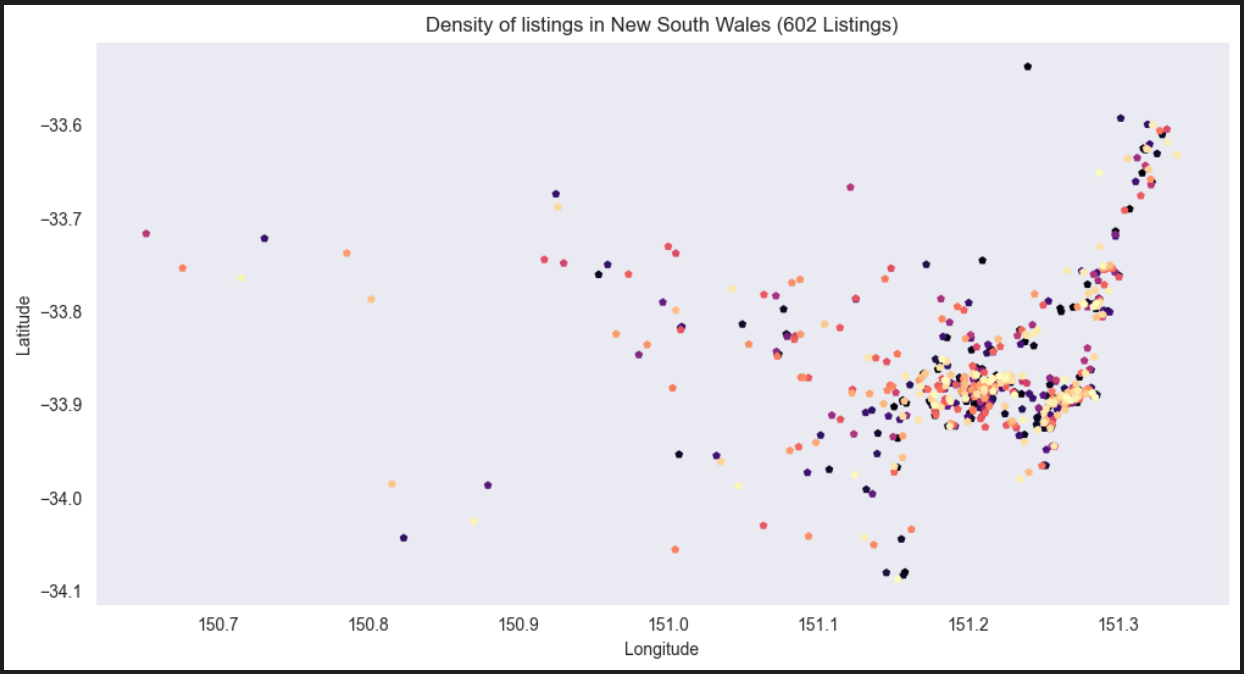

Density of listings in Quebec (632 Listings)
Latitude
Longitude
45.65
45.60
45.55
45.50
45.45
−73.90
−73.85
−73.80
−73.75
−73.70
−73.65
−73.60
−73.55
−73.50

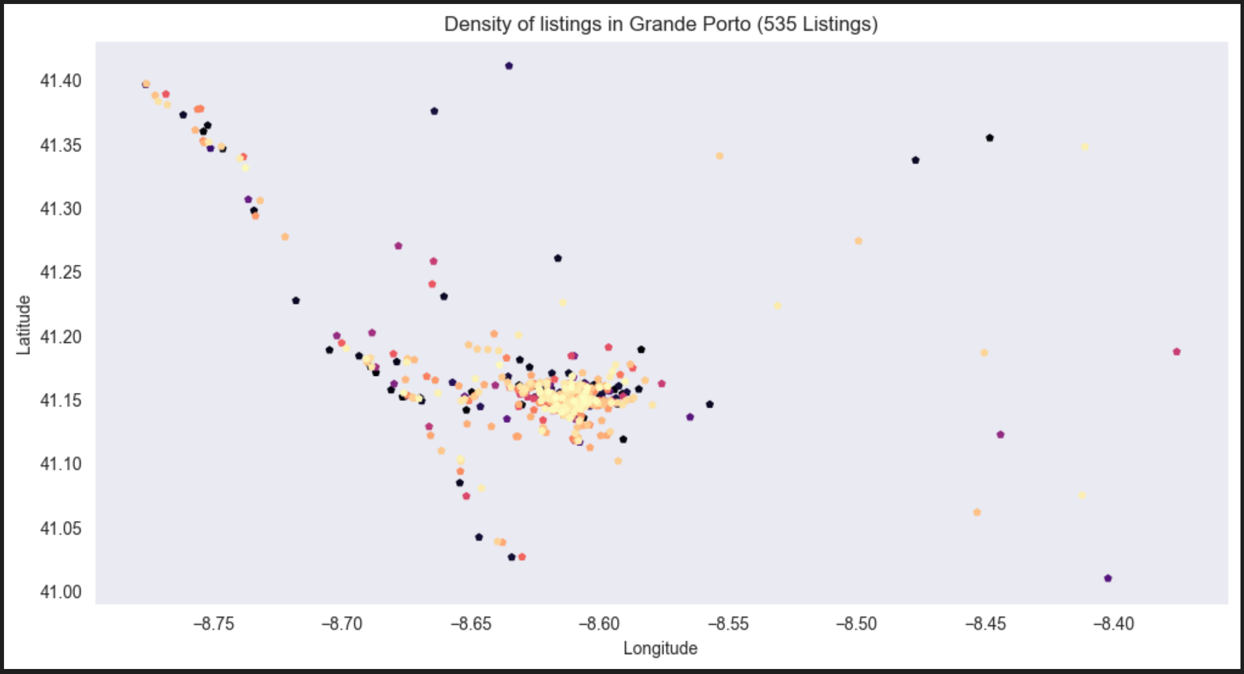

Density of listings in Catalonia (628 Listings)
Latitude
41.46
41.44
41.42
41.40
41.38
41.36
41.34
2.10
2.12
2.14
2.16
2.18
2.20
2.22
Longitude

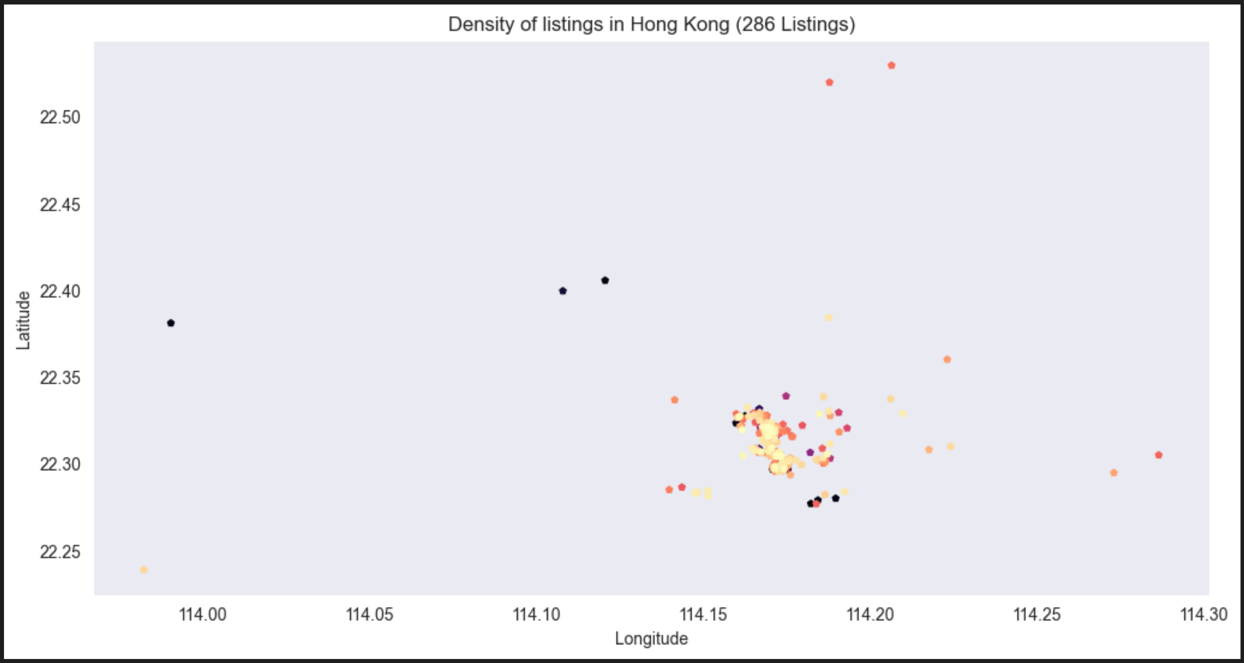

Density of listings in Hawaii (612 Listings)
Latitude
22.0
21.5
21.0
20.5
20.0
19.5
19.0
−159
−158
−157
−156
−155
Longitude

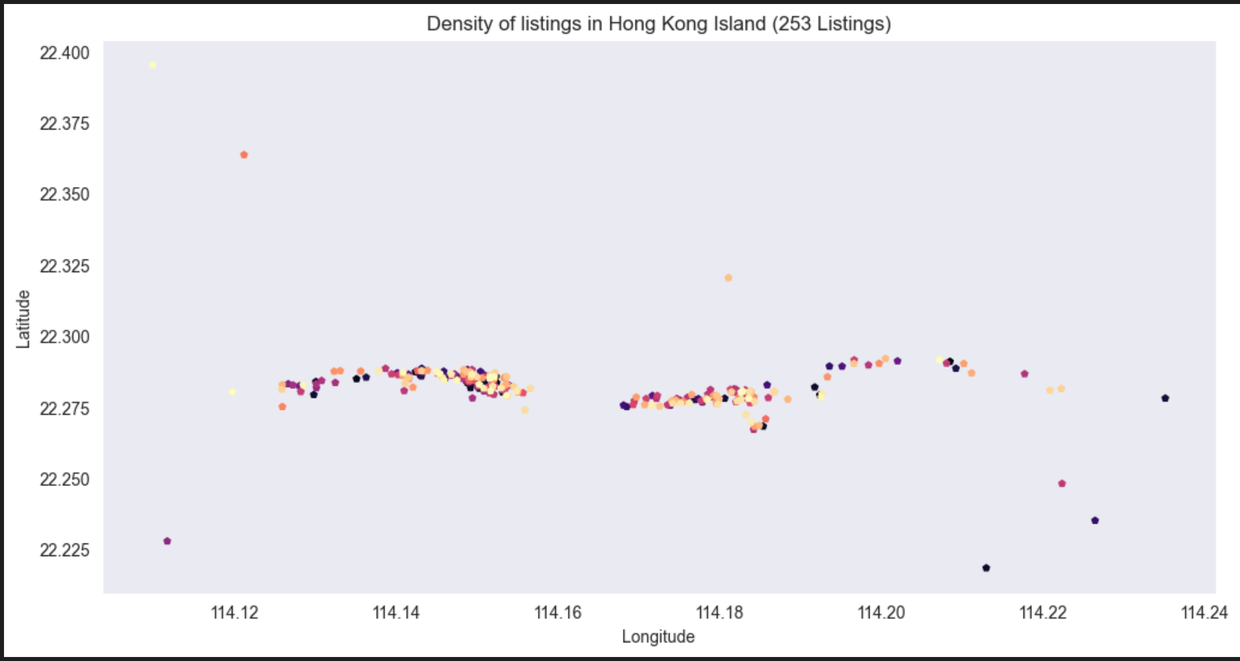

Density of listings in New York (605 Listings)
Latitude
Longitude
40.85
40.80
40.75
40.70
40.65
40.60
40.55
−74.1
−74.0
−73.9
−73.8


Density of listings in Rio De Janeiro (605 Listings)
Latitude
Longitude
−22.80
−22.85
−22.90
−22.95
−23.00
−23.05
−43.6
−43.5
−43.4
−43.3
−43.2


Density of listings in New South Wales (602 Listings)
Latitude
−33.6
−33.7
−33.8
−33.9
−34.0
−34.1
150.7
150.8
150.9
151.0
151.1
151.2
151.3
Longitude


Density of listings in Grande Porto (535 Listings)
Latitude
41.40
41.35
41.30
41.25
41.20
41.15
41.10
41.05
41.00
−8.75
−8.70
−8.65
−8.60
−8.55
−8.50
−8.45
−8.40
Longitude


Density of listings in Hong Kong (286 Listings)
Latitude
22.50
22.45
22.40
22.35
22.30
22.25
114.00
114.05
114.10
114.15
114.20
114.25
114.30
Longitude


Density of listings in Hong Kong Island (253 Listings)
Latitude
22.400
22.375
22.350
22.325
22.300
22.275
22.250
22.225
114.12
114.14
114.16
114.18
114.20
114.22
114.24
Longitude