```python
# Importing required libraries

# Library for data cleaning and data manipulation
import pandas as pd
import numpy as np

# Library for data visualization
import seaborn as sns
import matplotlib.pyplot as plt
import folium
from folium.plugins import MarkerCluster
import geopandas as gpd

# Library for normality test
import scipy.stats as stats

# Load the Dataset
df = pd.read_csv(r"C:\Users\user\OneDrive\Documents\Airbnb-Listings-Bangkok-Project\data\cleaned\data_cleaned.csv")

# Drop unnecessary columns from the dataset
df.drop(columns=["id", "host_id"], inplace=True)
```

4.1. Geospatial Competition and Listing Density

Goal: To investigate how the concentration of Airbnb listings in specific neighborhoods affects the level of competition among hosts and how it influences the pricing of listings in Bangkok.

Approach: Use the Kruskal-Wallis test to assess if there are significant differences in average prices across different neighborhoods, which will help identify whether competition (in terms of listing density) impacts pricing strategies.

Hypothesis Testing:

- **H0** : There is no significant difference in the median price of Airbnb listings between different neighborhoods (This means that neighborhood does not affect the price of listings)
- **H1** : There is a significant difference in the median price of Airbnb listings between different neighborhoods (This means that the location (neighborhood) significantly influences the price of Airbnb listings)

```python
from scipy import stats
# Group data by neighbourhood and extract the price data for each neighborhood
test_hip = [df[df['neighbourhood'] == neighbourhood]['price']
                for neighbourhood in df['neighbourhood'].unique()]

# Perform Kruskal-Wallis test
kruskal_result = stats.kruskal(*test_hip)

# Step 3: Display the results of Kruskal-Wallis
```

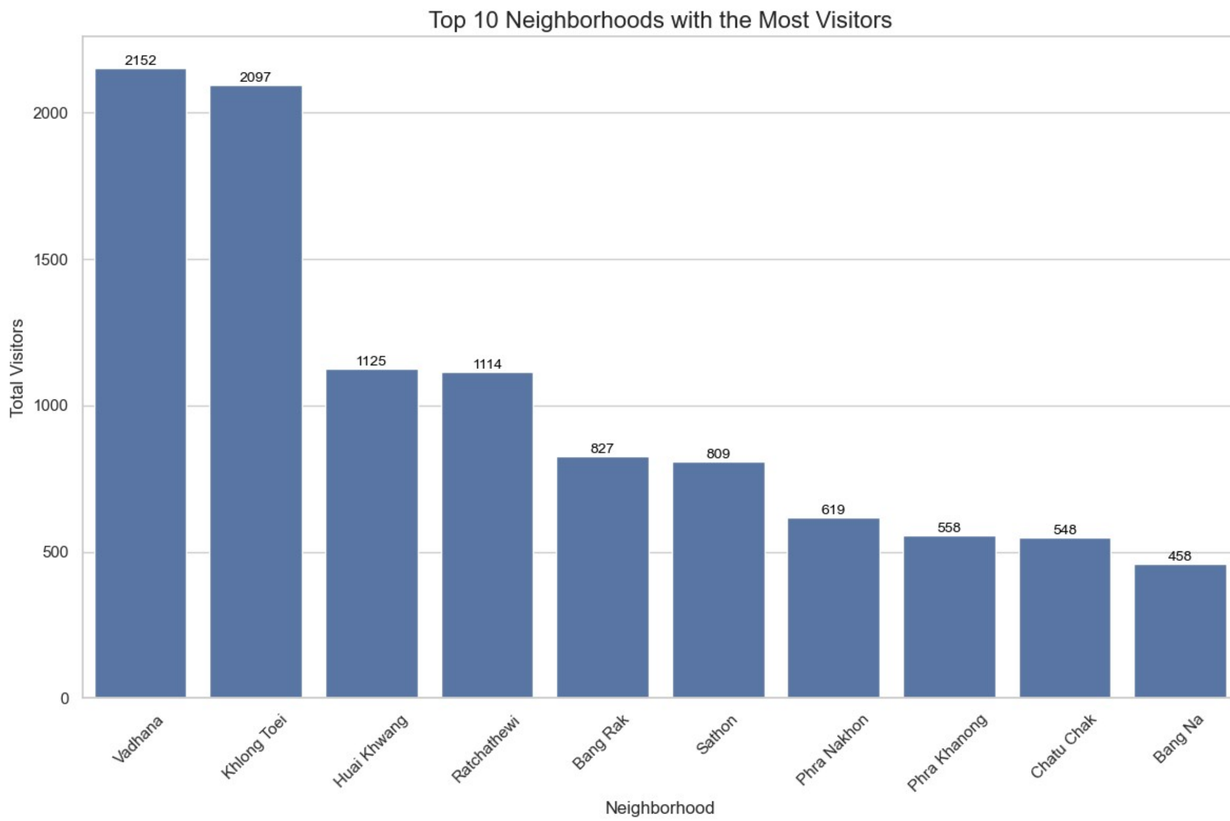
```
print("Kruskal-Wallis H-statistic:", kruskal_result.statistic)
print("Kruskal-Wallis p-value:", kruskal_result.pvalue)

# Step 4: Interpret the results
if kruskal_result.pvalue > 0.05:
    print(f'''
    pvalue={kruskal_result.pvalue}. pvalue > 0.05. Except Ho.
    I don't have enough evidence to conclude that the median price
across different neighborhoods in Bangkok is significantly different.
    ''')
else:
    print(f'''
    pvalue={kruskal_result.pvalue}. pvalue <= 0.05. Reject Ho.
    I have enough evidence to conclude that the median price across
different neighborhoods in Bangkok is significantly different.
    ''')
```

```
Kruskal-Wallis H-statistic: 2015.1185729999513
Kruskal-Wallis p-value: 0.0

    pvalue=0.0. pvalue <= 0.05. Reject Ho.
    I have enough evidence to conclude that the median price across
different neighborhoods in Bangkok is significantly different.
```

Conclusion: These results indicate that '**neighbourhoods**' have a significant impact on '**price**' compared to standard seasonal advertising. This means that this '**neighbourhoods**' is effective in increasing '**price**' at the airbnb listings Bangkok.

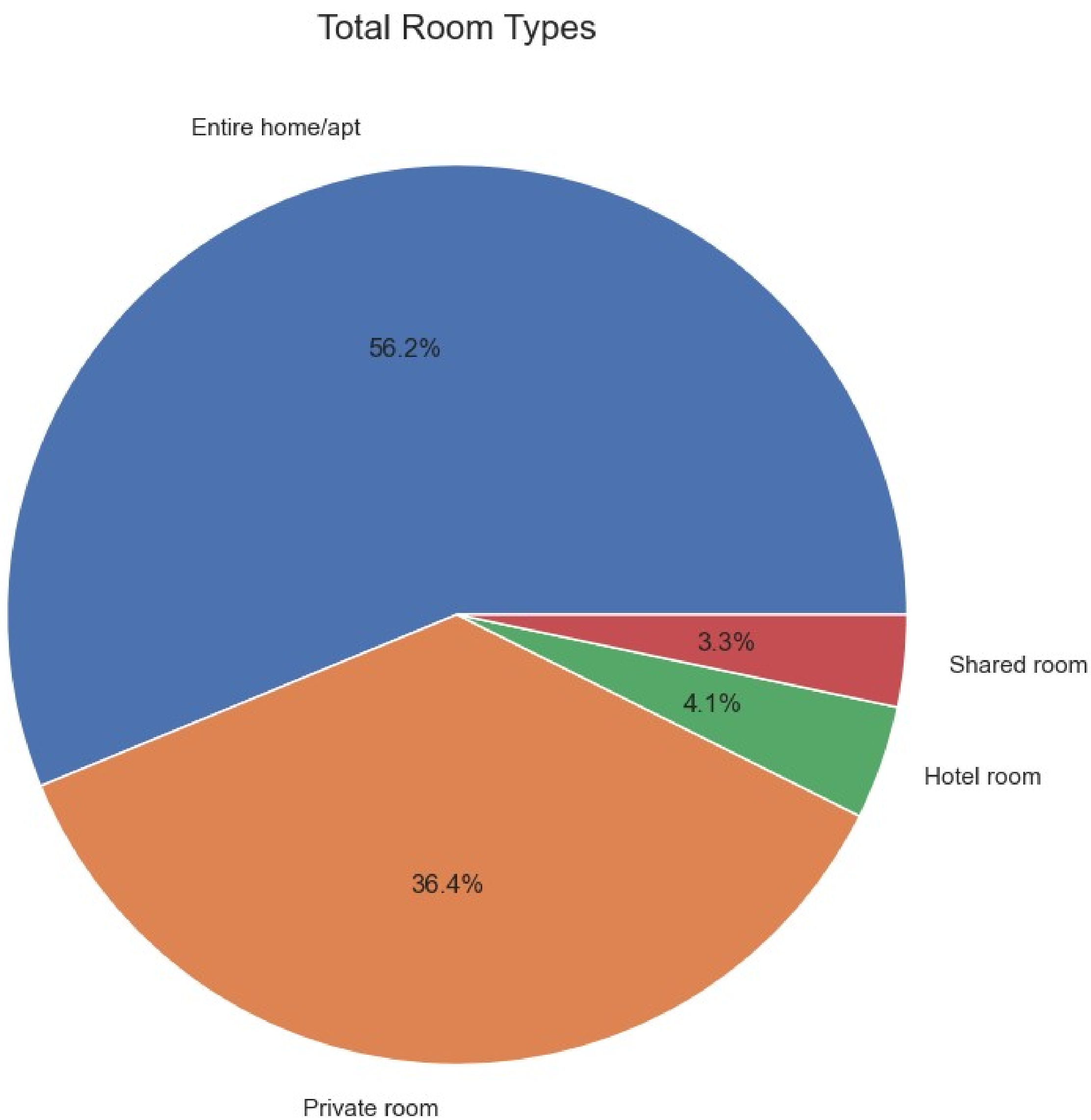
```
# Aggregate total visitors by neighborhood
neighborhood_data = df.groupby('neighbourhood').agg({
    'number_of_reviews': 'count',  # Summing the number of reviews
}).reset_index()

# Sort by the total number of visitors and select the top 10
neighborhoods
top_10_neighborhoods =
neighborhood_data.sort_values('number_of_reviews',
ascending=False).head(10)

# Visualization
plt.figure(figsize=(12, 8))
ax = sns.barplot(
    data=top_10_neighborhoods,
    x='neighbourhood',
    y='number_of_reviews'
)

# Add numbers above each bar
for index, bar in enumerate(ax.patches):
```

```
    x = bar.get_x() + bar.get_width() / 2
    y = bar.get_height()
    ax.text(x, y, f'{int(y)}', ha='center', va='bottom', fontsize=10,
color='black')

plt.title('Top 10 Neighborhoods with the Most Visitors', fontsize=16)
plt.xlabel('Neighborhood', fontsize=12)
plt.ylabel('Total Visitors', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
# Count the occurrences of each room type
room_type_counts = df['room_type'].value_counts()

# Pie chart visualization
plt.figure(figsize=(8, 8))
plt.pie(
    room_type_counts,
    labels=room_type_counts.index,
    autopct='%1.1f%%'  # Display percentages
)

plt.title('Total Room Types', fontsize=16)
```

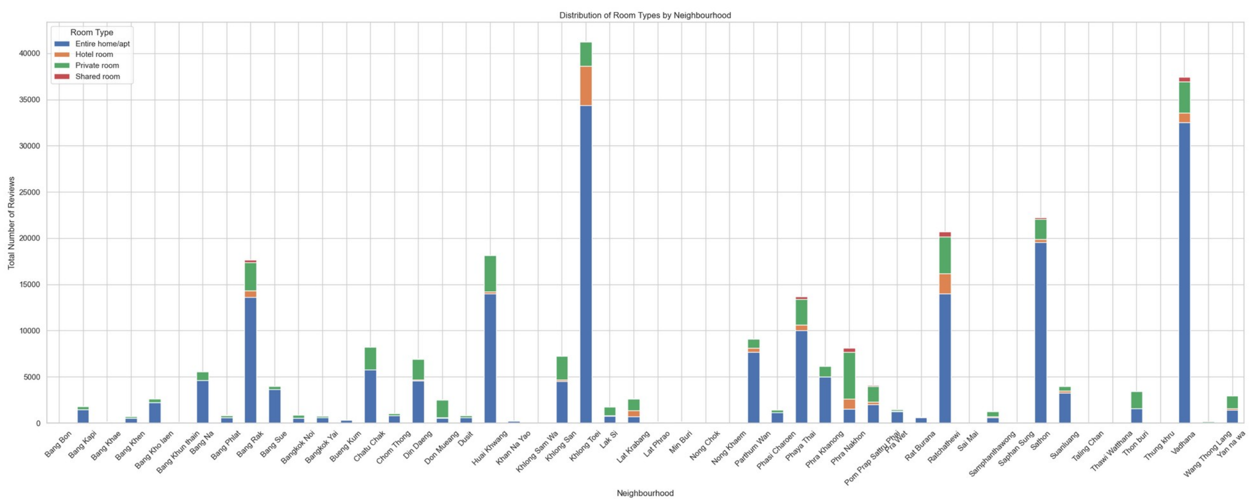
```
plt.tight_layout()
plt.show()
```



Insight : Neighborhoods with a higher customers across entire home/apartment are likely to be popular tourist destinations or have a high concentration of Airbnb hosts.

```
# Group data by 'neighbourhood' and 'room_type'
grouped_data = df.groupby(["neighbourhood", "room_type"])
['number_of_reviews'].sum().unstack(fill_value=0)

# Reset index for plotting
grouped_data = grouped_data.reset_index()
```

```python
# Create a stacked bar plot
ax = grouped_data.set_index('neighbourhood').plot(kind='bar',
stacked=True, figsize=(25, 10))

# Formatting the plot
plt.title('Distribution of Room Types by Neighbourhood')
plt.xlabel('Neighbourhood')
plt.ylabel('Total Number of Reviews')
plt.xticks(rotation=45)
plt.legend(title='Room Type')
plt.tight_layout()  # To ensure the plot doesn't get cut off

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



Insight: The "Entire home/apt" category consistently dominates the distribution across most neighborhoods. This suggests a strong preference for independent accommodations among Airbnb users. Neighborhoods with a higher number of listings across all room types are likely to be popular tourist destinations or have a high concentration of Airbnb hosts.

Visualizing Listing Density and Price Distribution

```python
import json
import folium

# Aggregate the data to get the average price per neighbourhood
price_mean = df.groupby('neighbourhood').agg({'price':
'mean'}).reset_index()

# Load the GeoJSON file
geojson_path = 'C:/Users/user/OneDrive/Documents/Airbnb-Listings-
Bangkok-Project/notebooks/Bangkok-districts.geojson'

with open(geojson_path, 'r') as f:
```

```
    districts_geojson = json.load(f)

# Merge the GeoJSON and the price_mean data to include the average
price in the GeoJSON properties
for feature in districts_geojson['features']:
    neighbourhood_name = feature['properties']['dname_e']
    match = price_mean[price_mean['neighbourhood'] ==
neighbourhood_name]
    if not match.empty:
        feature['properties']['average_price'] =
match['price'].values[0]
    else:
        feature['properties']['average_price'] = 'N/A'

# Bangkok coordinates
lat = 13.736717
long = 100.523186

# Create a Folium map for average price
bangkok_map = folium.Map(
    location=[lat, long],
    zoom_start=10,
    dragging=False,
    zoomControl=True,
    scrollWheelZoom=False,
    doubleClickZoom=False
)
tiles = 'https://tile.openstreetmap.de/{z}/{x}/{y}.png'
attr = 'Map <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors'
folium.TileLayer(tiles=tiles, attr=attr).add_to(bangkok_map)

# Add a choropleth layer to the map
choropleth = folium.Choropleth(
    geo_data=districts_geojson,
    name='choropleth',
    data=price_mean,
    columns=['neighbourhood', 'price'],
    key_on='feature.properties.dname_e',  # Key for matching the
GeoJSON properties
    fill_color='Set1',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Average Airbnb Price'
).add_to(bangkok_map)

# Add tooltips
folium.GeoJson(
    districts_geojson,
```

```python
    style_function=lambda feature: {
        'fillColor': 'transparent',
        'color': 'transparent',
        'weight': 0,
    },
    tooltip=folium.GeoJsonTooltip(
        fields=['dname_e', 'average_price'],
        aliases=['Neighbourhood', 'Average Price'],
        localize=True,
        sticky=False
    )
).add_to(bangkok_map)

# Add markers for top 10 tourist destinations
tourist_spots = {
    "Grand Palace": [13.7515, 100.4927],
    "Wat Arun": [13.7437, 100.4886],
    "Khao San Road": [13.7594, 100.4976],
    "Chatuchak Market": [13.7990, 100.5500],
    "Siam Paragon": [13.7462, 100.5349],
    "Wat Pho": [13.7466, 100.4923],
    "MBK Center": [13.7446, 100.5296],
    "Yaowarat (Chinatown)": [13.7403, 100.5072],
    "Lumphini Park": [13.7314, 100.5391],
    "Siam Amazing Park": [13.8151, 100.6321]
}
for name, coords in tourist_spots.items():
    folium.Marker(
        location=coords,
        popup=f"Tourist Spot: {name}",
        icon=folium.Icon(color='red', icon='info-sign')
    ).add_to(bangkok_map)

# Add markers for airports
airports = {
    "Suvarnabhumi Airport (BKK)": [13.6900, 100.7500],
    "Don Mueang Airport (DMK)": [13.9125, 100.6074]
}
for airport, coords in airports.items():
    folium.Marker(
        location=coords,
        popup=f"Airport: {airport}",
        icon=folium.Icon(color='black', icon='plane')
    ).add_to(bangkok_map)

# Add markers for universities
universities = {
    "Chulalongkorn University": [13.7462, 100.5320],
    "Thammasat University": [13.7564, 100.4931],
    "Mahidol University": [13.8185, 100.3282]
```

```
}
for university, coords in universities.items():
    folium.Marker(
        location=coords,
        popup=f"University: {university}",
        icon=folium.Icon(color='green', icon='university')
    ).add_to(bangkok_map)

# Display the map
bangkok_map
```

```
<folium.folium.Map at 0x225d7b14d40>
```

Insight:

- **Geographical Distribution:** In the central district, there are significant clusters of Airbnb listings, suggesting a high demand for short-term rentals in these tourist-heavy and business-centric areas. These regions are likely favored by both international tourists and business travelers due to their proximity to key attractions and transit options.
- **Room Type Distribution: The red markers** for **entire home/apartment listings** are **concentrated in popular areas**, reflecting the demand for larger, more private accommodations by tourists. On the other hand, **private rooms (green markers)** are more **widely distributed** across non-central neighborhoods, indicating a greater supply of budget-friendly options for solo travelers or those seeking shorter stays.
- **Most Popular Neighbourhood:** Vadhana **with over 2000 listings** is the **most popular neighborhood** for Airbnb in Bangkok. This suggests that it's a highly sought-after area for tourists or has a significant number of residents renting out their properties.
- **Proximity to Key Locations:** The proximity of Airbnb listings to **key tourist spots**, **such as Grand Palace and Wat Arun**, influences the pricing of accommodations in those areas. Listings near **Suvarnabhumi Airport or Don Mueang Airport** cater to travelers seeking convenience and proximity to flight terminals. Similarly, areas near **Chulalongkorn University or Mahidol University** may attract budget-conscious students, with slightly lower pricing compared to the city center."