اونيۏرسيتي مليسيا ڤهڠ
UNIVERSITI MALAYSIA PAHANG

PROJECT BSD2333 DATA WRANGLING

TITLE:
AIRBNB MUNICH, BAVARIA GERMANY 2019 ANALYSIS

LECTURER'S NAME:
MOHD KHAIRUL BAZLI BIN MOHD AZIZ

Group name: Wranglersbsd

MATRIC ID	NAME	SECTION
SD21010	NOR FARAWAHIDA BT ABDULLAH	01G
SD21015	NURUL ATHIRAH BT RAMLI	01G
SD21016	HANIS SHAHIRA BT SHAHIMI	01G
SD21012	NUR ADIBAH BT MOHD ZUHAIRI	01G

AIRBNB MUNICH, BAVARIA GERMANY 2019 ANALYSIS

1.0 Synopsis

1.1 Description of the assignment

On December 9, 2020, Airbnb conducted its Initial Public Offering (IPO), marking the moment when a privately held company becomes publicly traded by selling its shares to the general public. The trading of Airbnb's shares commenced on December 10, 2020. Remarkably, the company's shares were valued at USD144.71 on that day, an astonishing 112.00% increase compared to the IPO price of USD 68.00. This surge in share value propelled Airbnb's market worth to USD 100 Billion, surpassing well-established hotel industry giants such as Marriott International, Hilton, and Hyatt.

Founded in 2008, Airbnb has developed a user-friendly platform that appeals to a wide range of users. It serves as a connection hub between travelers or individuals seeking short-term accommodations and hosts who are willing to rent out their properties. Despite not owning any hotels, Airbnb operates in over 80,000 cities worldwide and has welcomed more than 300 million guests. It is widely recognized as the largest hotel company in existence.

Airbnb's business strategy plays a significant role in fulfilling its mission. The company follows a platform business model, acting as an intermediary between suppliers and consumers instead of directly owning the services it offers. In return for its services, Airbnb earns a commission ranging from 9% to 15%. Another crucial aspect of Airbnb's business strategy is its extensive integration of technology into various aspects of its operations. The company places great emphasis on its community and fosters trust among its stakeholders as a core element of its business strategy.

Among Airbnb's notable initiatives is the provision of website data for major cities worldwide. Through the platform called Inside Airbnb, users can access and download substantial datasets, enabling the development of various Data Science projects and solutions.

However, despite its success as a disruptive force in the travel industry, Airbnb does face certain weaknesses. These include the challenge of ensuring direct control over service quality and the susceptibility of its brand image to various scandals. Additionally, the replicability of Airbnb's business model and the absence of traditional amenities like reception, housekeeping, and room service may deter certain customer segments.

1.2 Problem to be solve

The database used needs a cleaning process as there are massive outliers and also null values. We had dropped a few columns since the columns would not be used in our analysis. We had set parameters which will be considered outliers since the amount of outliers are too many.

1.3 Questions to be answered

1. What is the proportion of the property types in Munich on Airbnb? And what is the proportion by neighbourhood?
2. What is the average property price in the different neighbourhoods of Munich?
3. Which regions or neighborhoods that we can see sparse areas or gaps with fewer Airbnb listings?
4. What is the overall trend between the minimum nights requirement and rental prices? Is there a noticeable correlation between these variables?

1.4 Objectives

1. To identify the proportion of the property types in Munich on Airbnb and the proportion by neighbourhood.
2. To determine the average property price in the different neighbourhoods of Munich.
3. To observe the regions or neighbourhoods that have sparse and gaps with fewer Airbnb listings.
4. To analyze the overall trend between the minimum nights requirement and rental prices.

1.5 Basic description of the data

No.	Attributes	Explanation
1.	id	Property id number
2.	name	Property name announced
3.	host_id	Host id number
4.	host_name	Host name
5.	neighbourhood_group	There are no values and will be deleted
6.	neighbourhood	Neighbourhood name
7.	latitude	Property latitude
8.	longitude	Property longitude
9.	room_type	Property type announced
10.	price	Daily rate
11.	minimum_nights	Minimum number of nights to book
12.	number_of_reviews	Total number of reviews for the property
13.	last_review	Date of the last review
14.	reviews_per_month	Number of reviews per month
15.	calculated_host_listings_count	Number of properties from the same host
16.	availability_365	Number of days available in the calendar year
17.	number_of_reviews_ltm	Number of reviews in the last twelve months
18.	license	Host license

The coloured rows are the only attributes that were used in our analysis while the others were dropped since they were not to be used in our analysis.

2.0 Packages Required

Before running the Python codes, the libraries that need to be installed are as follows in order for the given Python codes to be run:

1. **Pandas**
   Pandas is the core package that is often used in the Be a Data Wrangler assignment.
   To store the data in readable data format, Pandas' Data Frame will be implemented. We also used pd.read_csv() to retrieve the CSV file into DataFrame format. We also use slicing to index the data by [] to filter the data and select particular sets of Column and Variable.

2. **Numpy**
   NumPy is an essential package for scientific computing in Python as a foundational library. It provides a versatile multidimensional array object with a collection of functions to perform mathematical and logical operations on arrays. It has helped in efficiently counting the occurrences of a specific condition in an array or data frame column.

3. **Matplotlib**
   Matplotlib is a popular data visualization library in Python that provides a wide range of tools for creating static, animated, and interactive visualizations. It allows you to create various types of plots, including line plots, bar plots, scatter plots, histograms, pie charts, and more. From the visualizations, meaningful insights can be extracted from it.

4. **MissingNo**
   'missingno' is a Python library that provides a convenient way to visualize and analyze missing data in a dataset. It allows you to quickly identify patterns and trends in missing values, helping you to understand the completeness of your data.

5. **Geopandas**

   Geopandas expands the functionality of the well-known pandas data manipulation toolkit to handle geographical data. It offers features and tools for performing spatial operations and analysis on geographical data, including points, lines, and polygons.

6. **Seaborn**

   Seaborn is a Python data visualization library. It offers a sophisticated user interface for producing educational and aesthetically pleasing statistical visuals. By providing a large range of built-in capabilities and settings that improve the aesthetics and readability of plots, Seaborn makes it easier to create sophisticated visualizations.

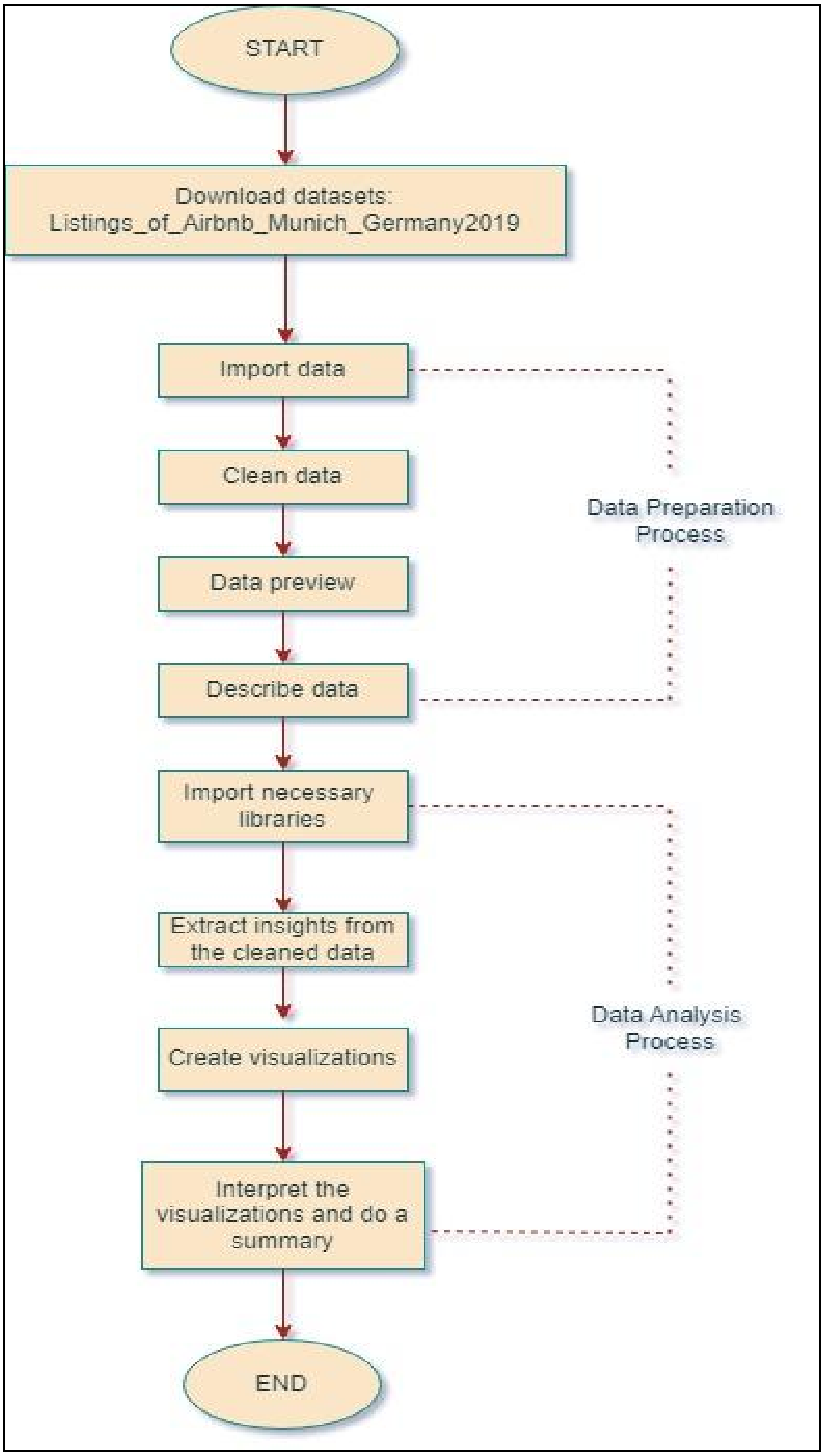
7. **Folium**

   Folium is a Python library that enables the generation of interactive leaflet maps and visualizations. It utilizes the Leaflet.js mapping library and provides a convenient way for users to create maps directly in Jupyter notebooks or web applications. Folium provides a user to create interactive maps and visualize geospatial data in Phyton.

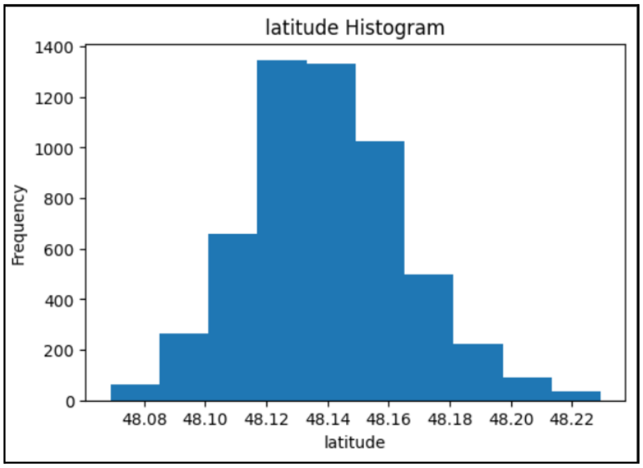
8. **Pillow**

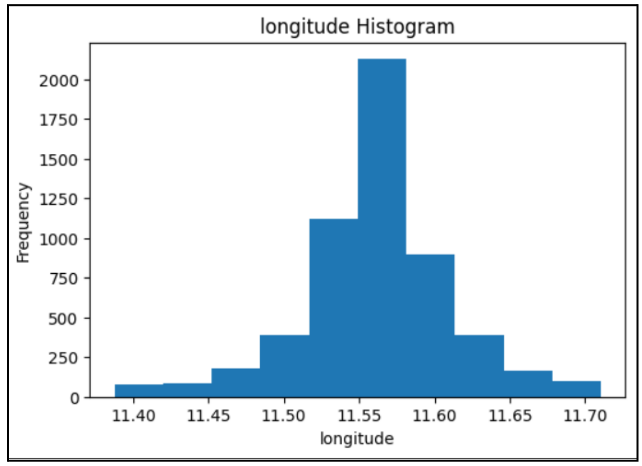
   Pillow is a popular Python library for image processing and manipulation. It is a fork of the Python Imaging Library (PIL) and provides an easy-to-use API for working with images in various formats. Pillow finds extensive application in diverse fields, such as computer vision, web development, scientific research, and digital art, where image processing and manipulation are integral. Its rich feature set and user-friendly nature make it a robust and effective image processing tool for Python-based projects involving image handling.

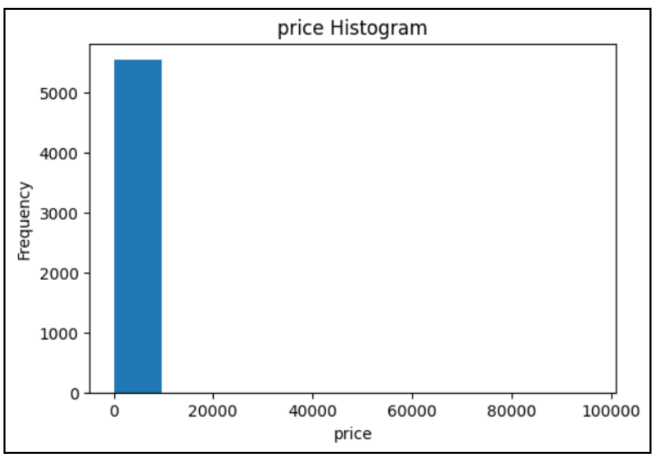
3.0 Data Preparation

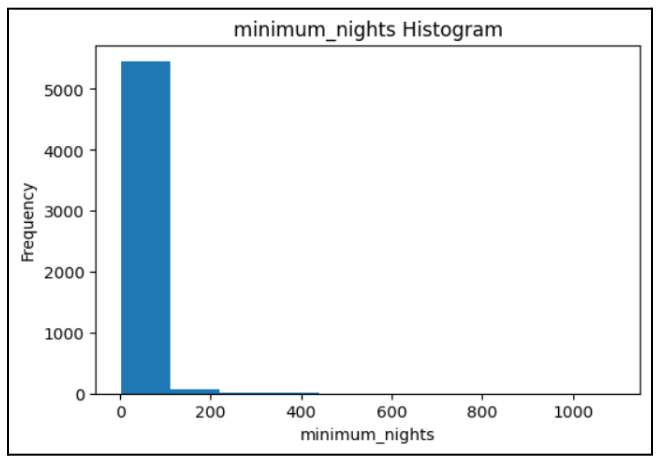


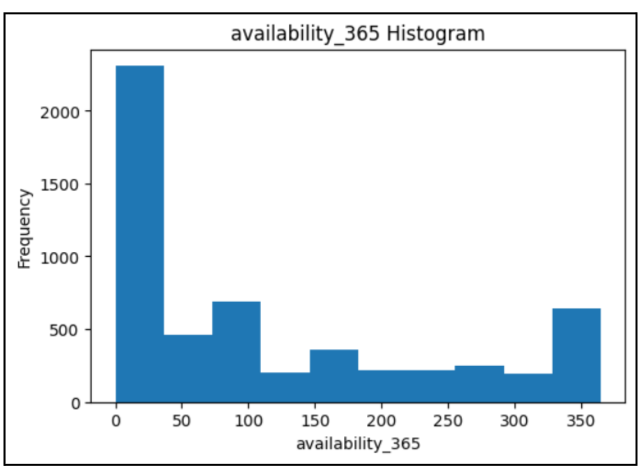
4.0 Exploratory Data Analysis







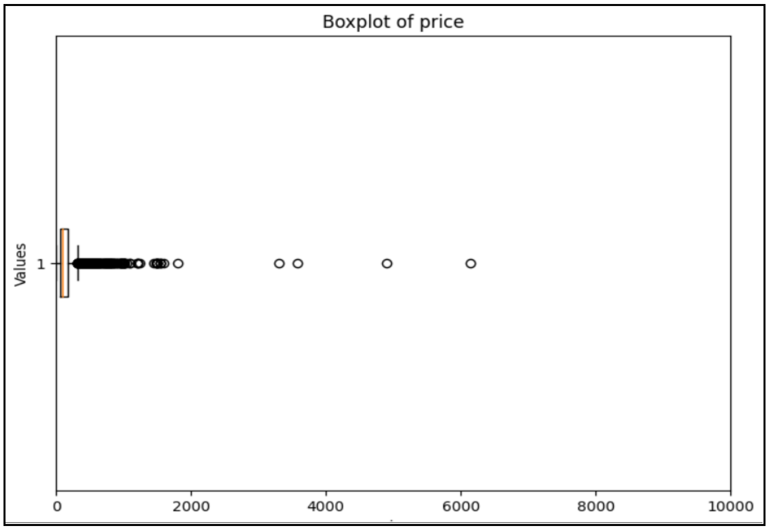


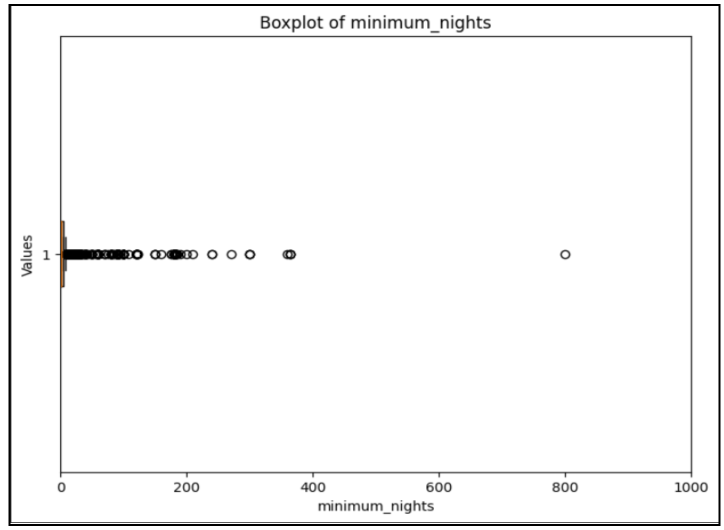


Through the five histograms above , it is possible to verify the presence of outliers in the variables 'price' and 'minimum_nights'. The values do not follow a distribution and distort the entire graphical presentation.

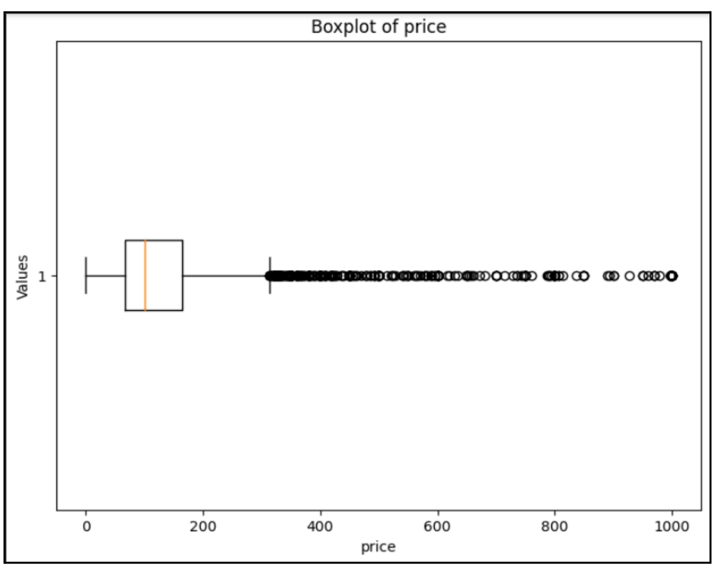
	latitude	longitude	price	minimum_nights	availability_365
count	5533.000000	5533.000000	5533.000000	5533.000000	5533.000000
mean	48.139611	11.562330	170.836978	9.009398	114.627327
std	0.025659	0.048663	1308.536534	31.365482	125.219271
min	48.068870	11.387475	0.000000	1.000000	0.000000
25%	48.122560	11.538820	66.000000	1.000000	0.000000
50%	48.137080	11.564030	100.000000	2.000000	72.000000
75%	48.155700	11.585310	167.000000	4.000000	206.000000
max	48.229500	11.710610	96274.000000	1095.000000	365.000000

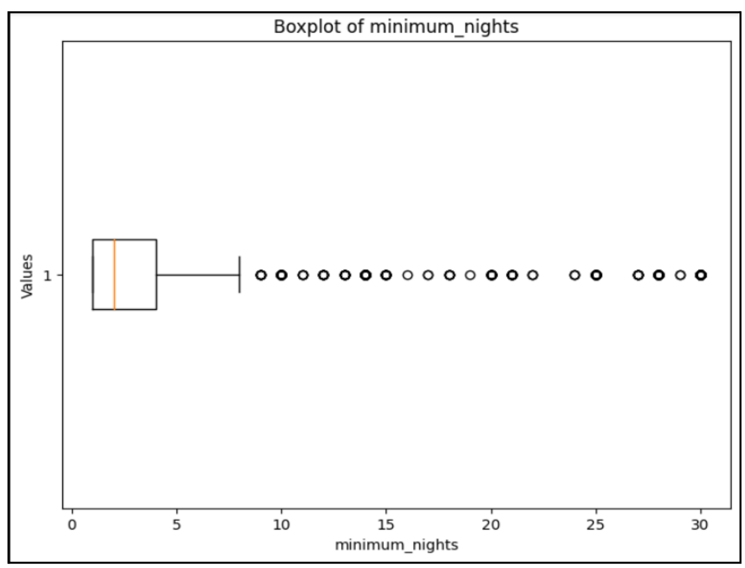
The variable 'price' has 75% of its values below 167, but its maximum value is 96274, which is well above the values obtained up to 75% of the dataset, which proves the presence of outliers. The variable 'price' presents minimum values equal to 0. Understanding the Airbnb business, it is known that no one rents any property on Airbnb for free. The variable 'minimum_nights' has 75% of its values below 4, but its maximum value is 1095, which is well above the values obtained up to 75% of the dataset, which proves the presence of outliers.

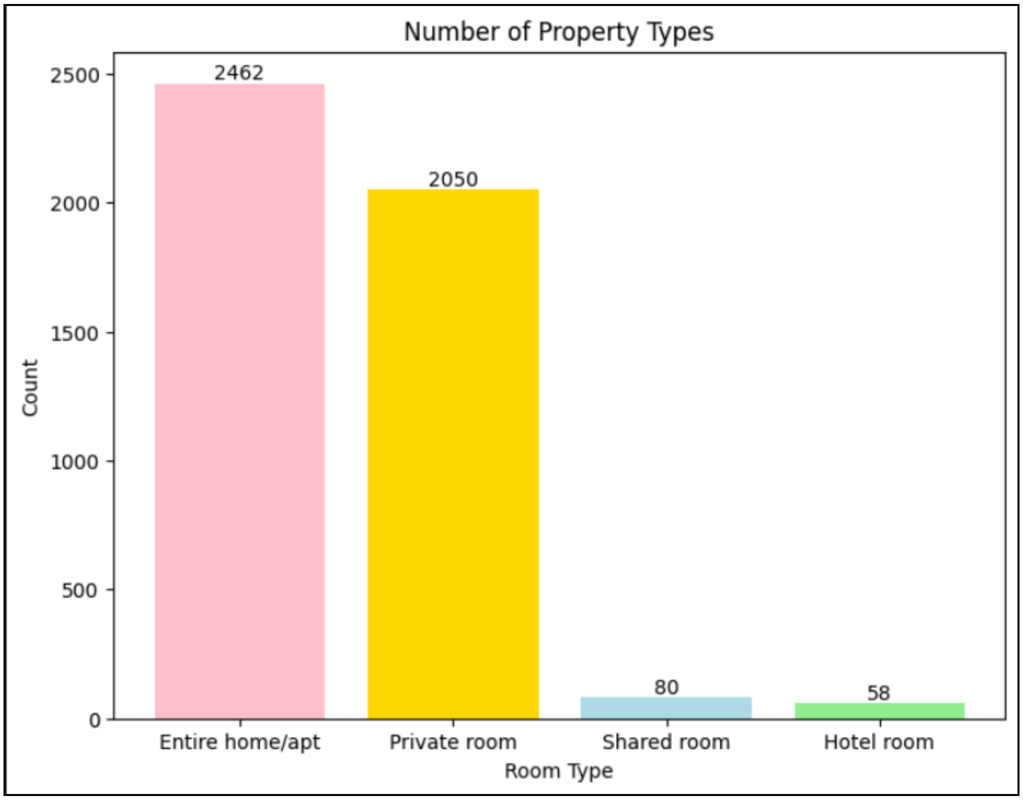




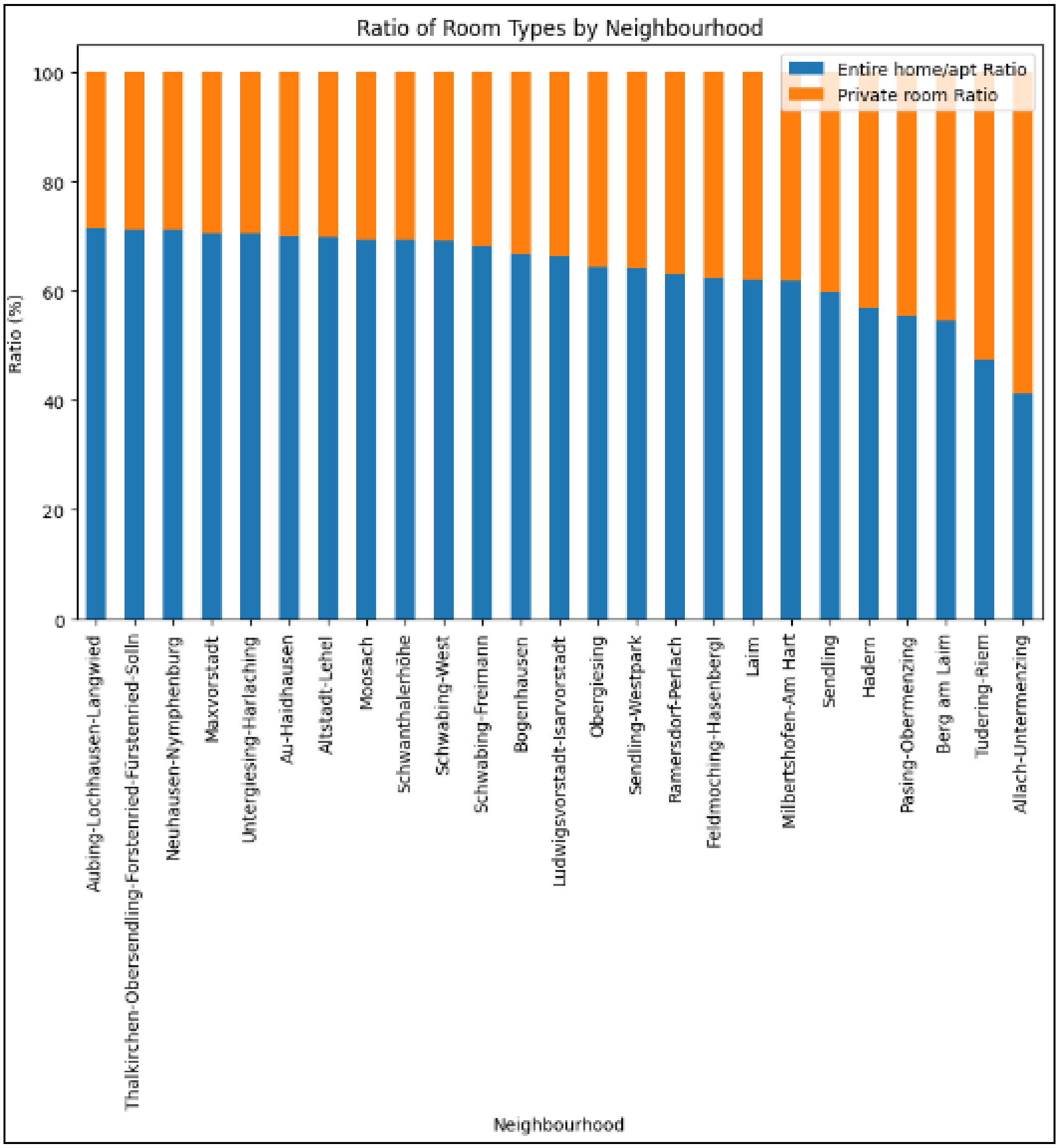
Above, the boxplot for the variable 'price' visually shows the information contained in the summary statistics. As a parameter, all data greater than 1000 will be considered an outlier in this analysis. Above, the boxplot for the variable 'minimum_nights' visually shows the information contained in the summary statistics. As a parameter, all data greater than 30 will be considered an outlier in this analysis. We had set parameters which will be considered outliers since the amount of outliers are too many. Below is the boxplot after removing outliers applied.



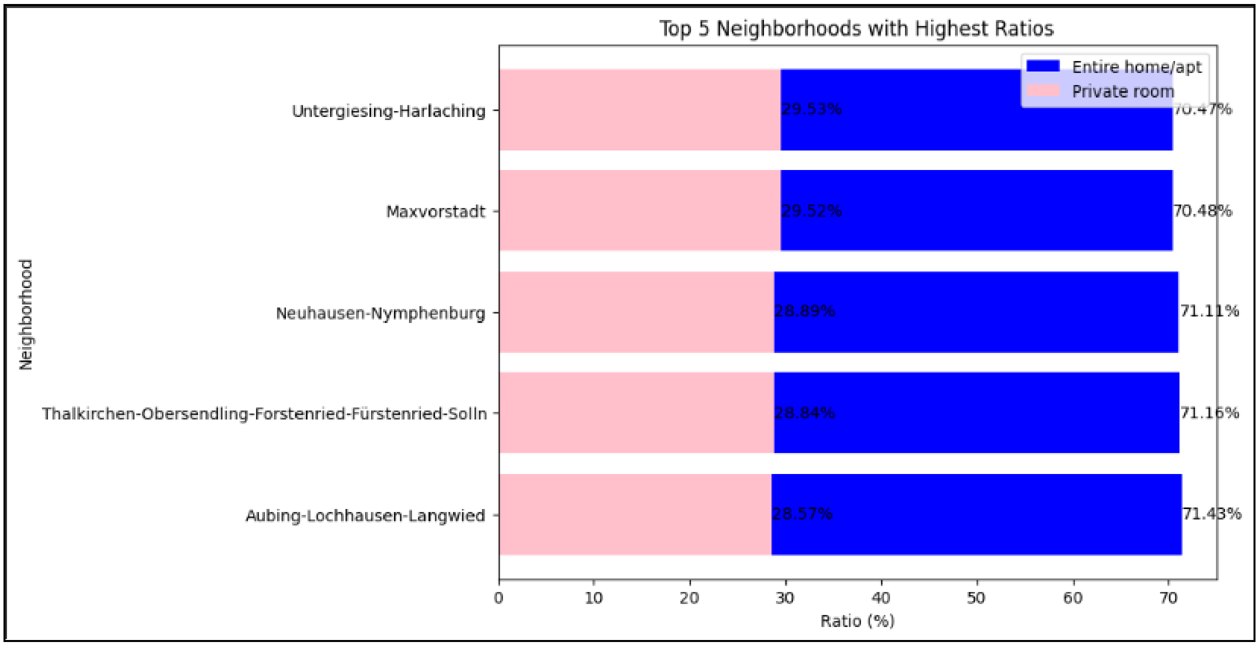




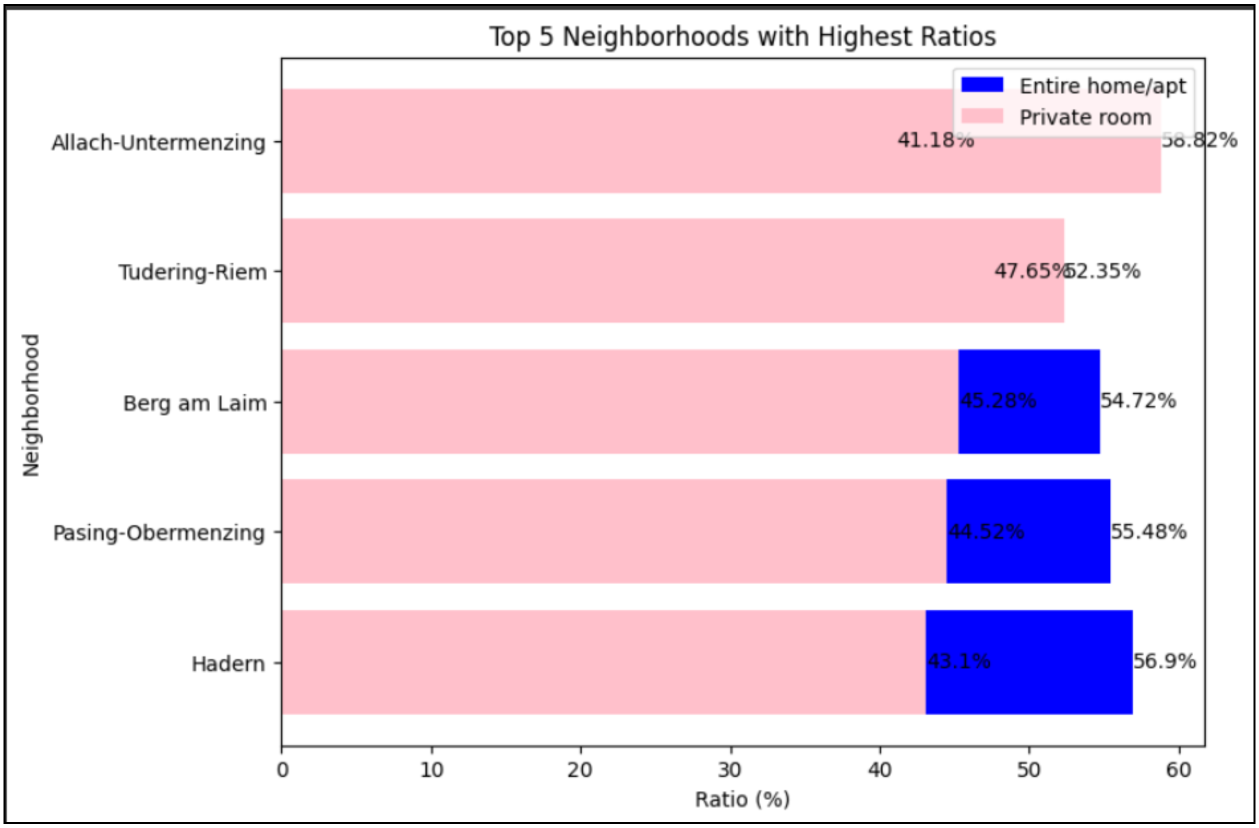
As the property types 'Shared room' and 'Hotel room' are not relevant to the number of properties being rented on Airbnb in the city of Munich, we will continue this analysis in the neighbourhoods using only the property types 'Entire home/apt' and 'Private room'. In Munich, Germany, hotel rooms and shared rooms might not be as common as other property types for a variety of reasons. First off, shared rooms often offer little in the way of personal space or privacy, which may not be what most travellers like. Munich is a well-liked travel destination, drawing a variety of tourists who frequently seek out more secluded and pleasant lodgings. Additionally, Munich has a large supply of hotels with a variety of alternatives and amenities, which could reduce the demand for hotels with listings on Airbnb. Additionally, Munich's cultural tastes and accepted travel practices may have an impact on how popular shared rooms and hotel rooms are, with a stronger desire for complete homes or private rooms that offer a more individualised and opulent experience.



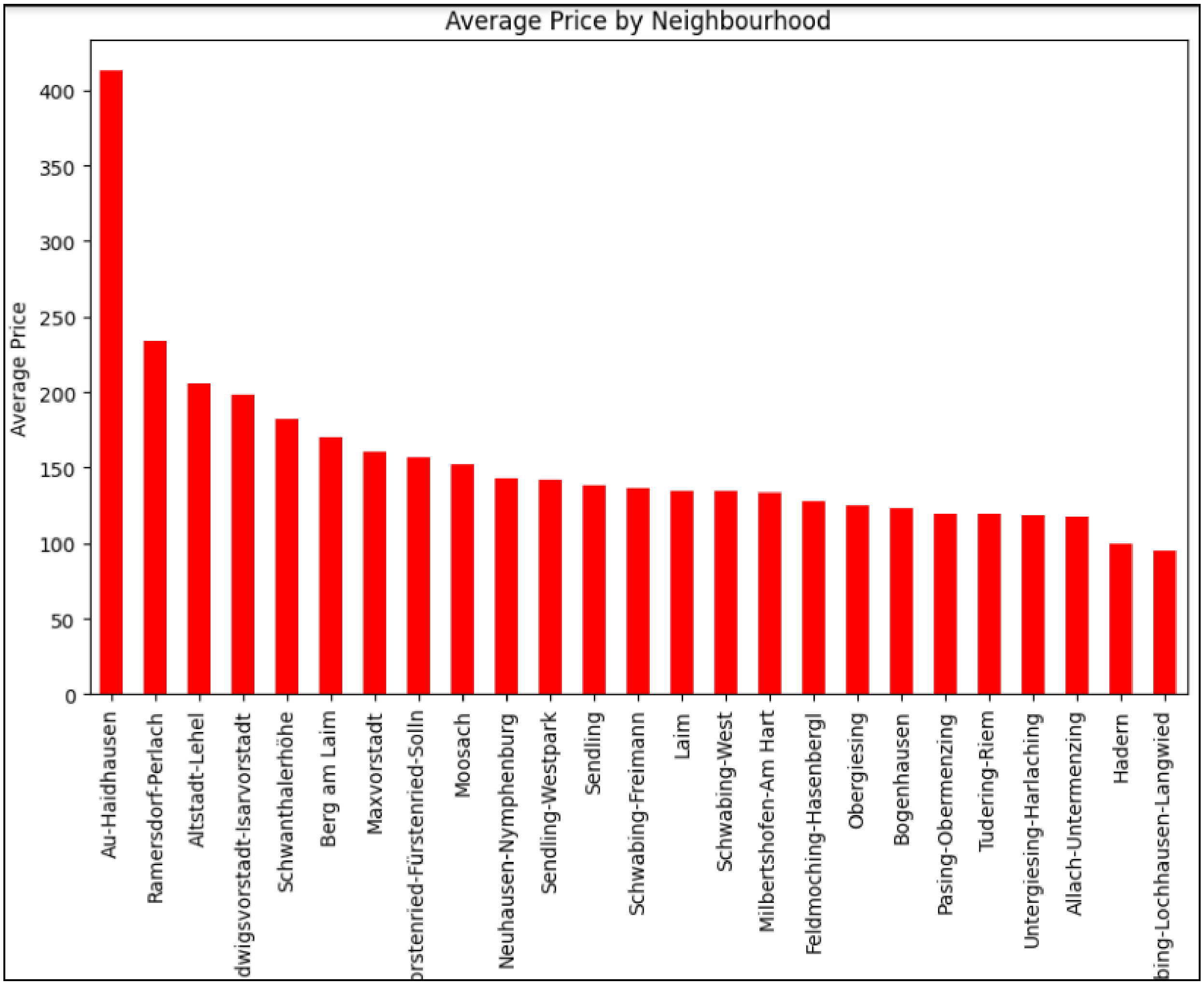
It can be seen that the distribution of property types of 'Entire home/apt' and 'Private Room' in Munich is well balanced in most neighbourhoods. This indicates that both types of accommodations are available and in demand, providing a diverse range of options for travelers. A well-balanced distribution of property types can be beneficial for both hosts and guests. Hosts have the flexibility to offer different types of accommodations based on their property and preferences, while guests have the opportunity to choose the type of accommodation that suits their needs and preferences.



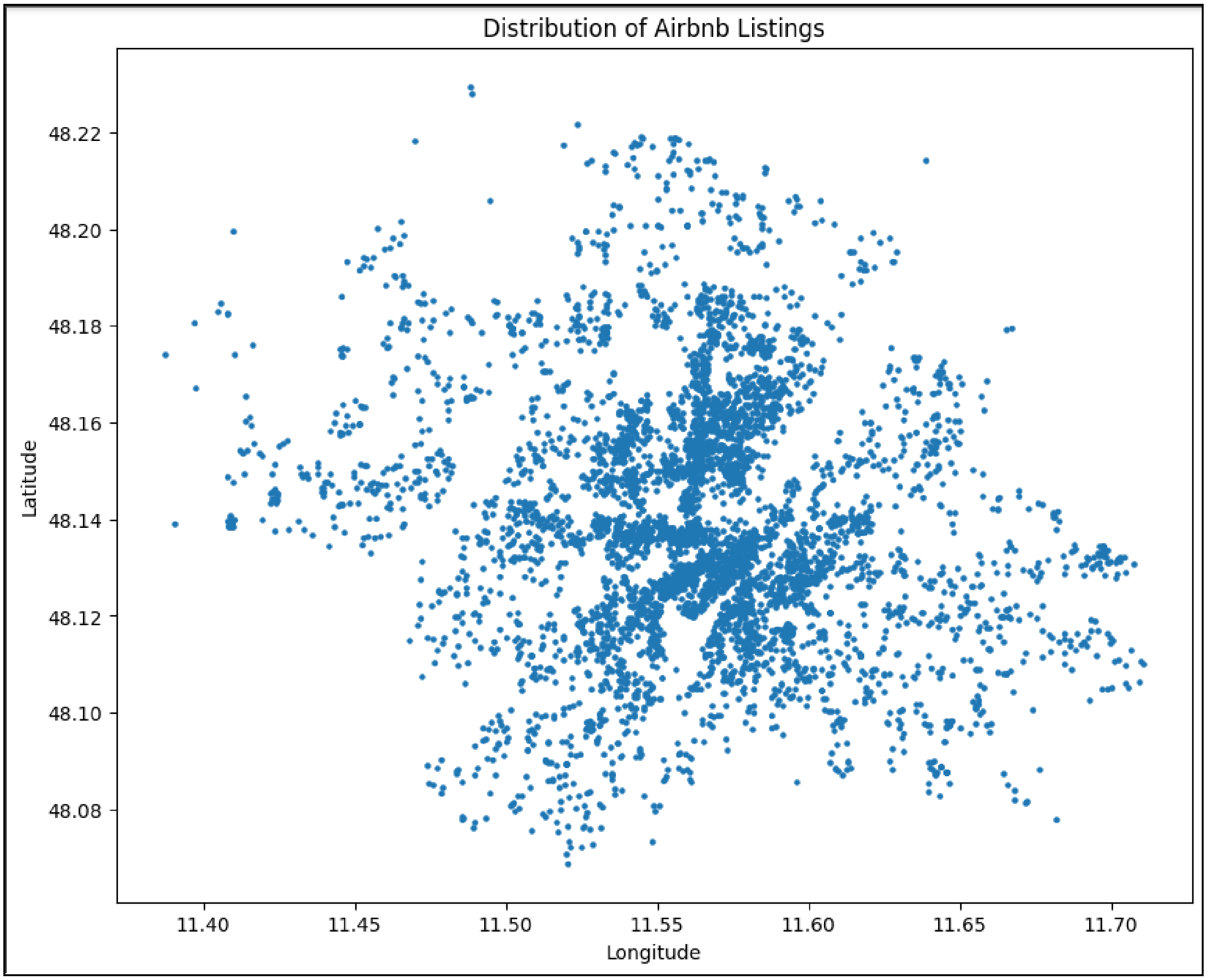
Above we can see the neighbourhoods where the proportion of property type 'Entire home/apt' is higher, so if the Airbnb user wants to stay in one of the neighbourhoods listed above, there is a greater chance that this user will find offers of this type property available. These neighborhoods in Munich may have a higher concentration of residential properties or property owners who are more inclined to rent out their entire homes/apartments rather than individual rooms.



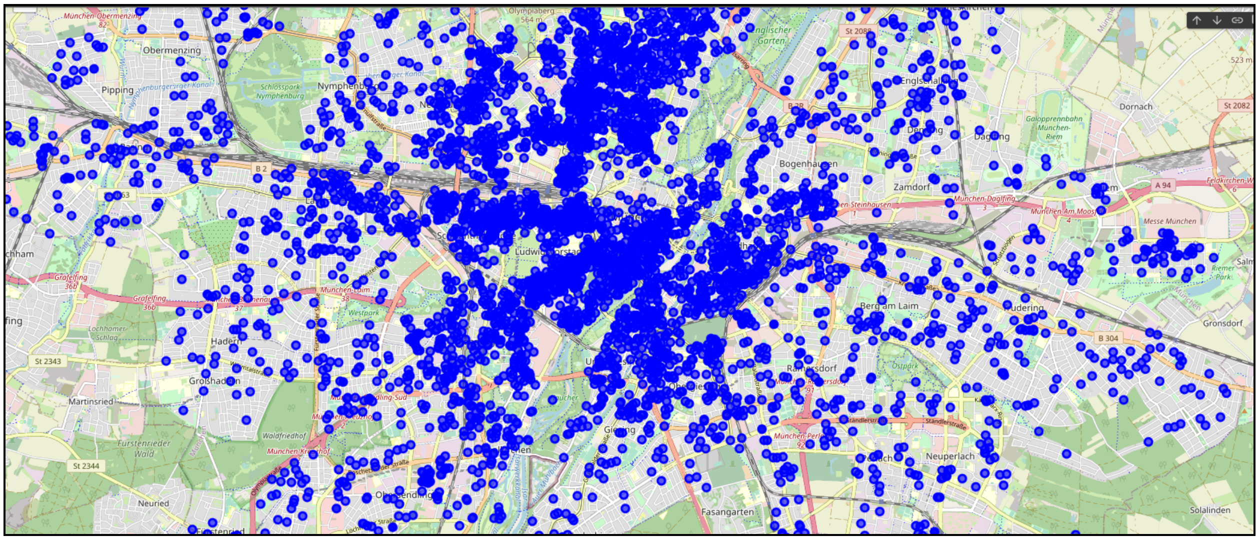
Above, we can see the neighbourhoods where the proportion of property type 'Private room' is higher, so if the Airbnb user wants to stay in one of the neighbourhoods listed above, there is a greater chance that this user will find offers for this type of property available. The unique characteristics of the neighborhoods, such as their location, amenities, atmosphere, or target audience, might make them more attractive for travelers or tenants seeking private room accommodations. For example, neighborhoods near universities or popular tourist destinations often have a higher demand for private rooms.



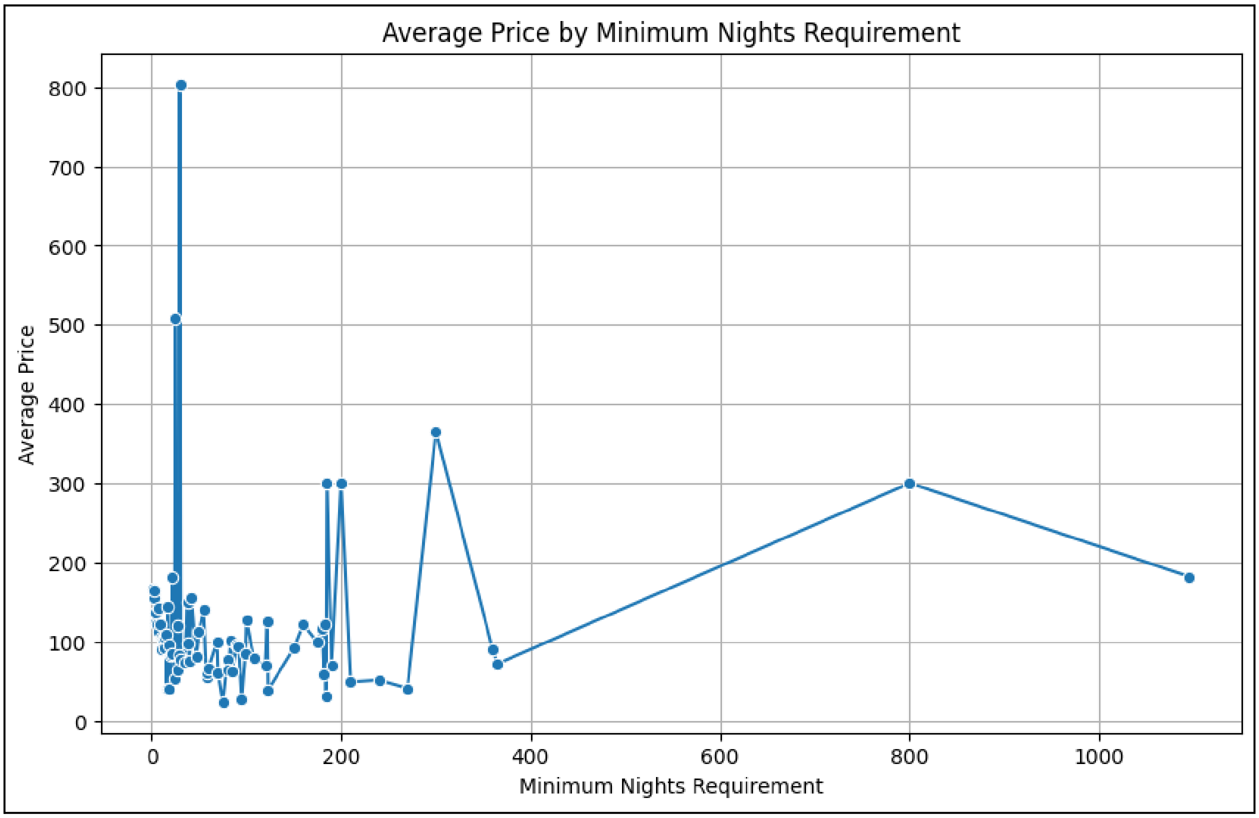
Below is the average property price per neighbourhood in Munich. We can see that there is a big price difference, where the average price in the most expensive neighbourhood is 412.64 and in the cheapest neighbourhood is 95.43.



Sparse areas or gaps on the scatter plot indicate regions in Munich with fewer Airbnb listings. These areas might be underrepresented or less popular for rentals, which could provide opportunities for market analysis or potential expansion of Airbnb services.



Above graph shows more specific locations with Airbnb activity. The graph provides a visual representation of the geographic distribution of Airbnb in Munich listings based on latitude and longitude coordinates. Each marker on the map represents a specific location. The density of markers indicates the concentration of Airbnb listings in different areas. We can observe clusters or hotspots where multiple markers are closely grouped together, indicating popular or densely populated areas for rentals. Sparse areas or gaps on the map with fewer markers indicate regions with lower Airbnb activity. These areas might be less popular for rentals, indicating potential gaps or opportunities for market analysis or expansion of Airbnb services. By examining the map, we can identify the proximity of Airbnb listings to specific points of interest such as tourist attractions, parks, transportation hubs, or commercial centers. This information can be useful for travelers or property owners in understanding the accessibility and desirability of different locations. The interactive nature of the map allows users to zoom in and out, as well as pan across different areas for a more detailed exploration of the distribution and patterns.



From this we can determine the optimal minimum nights for maximizing rental income or occupancy rates, we can look for points on the graph that correspond to higher average prices. These points indicate the minimum nights requirement that yields higher rental prices, which can contribute to maximizing rental income or occupancy rates.

There is no trend that we can see between the minimum nights requirement and rental prices. It is because rental prices are typically influenced by multiple factors, such as property size, location, amenities, and seasonal demand. The minimum nights requirement alone may not be the primary determinant of rental prices. We should consider analyzing the combined effect of multiple variables to gain a comprehensive understanding of the factors influencing rental prices.

5.0 Summary

The Airbnb project in Munich, Germany involved analyzing and exploring Airbnb listings data. The data was cleaned and processed to remove errors and prepare it for analysis. Descriptive analysis and visualizations were used to understand the data and identify patterns and trends. The project examined property types, rental prices, availability, and minimum nights requirement. Geospatial analysis helped visualize the distribution of listings across neighborhoods. Key insights were gained, including popular property types and neighborhoods, seasonal trends, and optimal minimum nights for rental income. The project aimed to provide recommendations for hosts and stakeholders in Munich to make informed decisions and maximize rental opportunities.

6.0 Reference

1) *Airbnb Inc. Report 2019*. (2020, August 8). Research-Methodology. https://research-methodology.net/airbnb-inc-report-2019/
2) *Airbnb research: an analysis in tourism and hospitality journal*. (2020). INTERNATIONAL JOURNAL OF CULTURE, TOURISM AND HOSPITALITY RESEARCH. https://www.emerald.com/insight/content/doi/10.1108/IJCTHR-06-2019-0113/full/pdf?title=airbnb-research-an-analysis-in-tourism-and-hospitality-journals
3) Kalender, S. (2021, December 15). Munich's Airbnb Data Analysis - Selmir Kalender - Medium. *Medium*. https://medium.com/@kalenderselmir/munichs-airbnb-data-analysis-fd815f2c918f

4) Daniel Adams Guttentag. (2016). *Why tourists choose Airbnb: A motivation-based segmentation study underpinned by innovation concepts.* degree of Doctor of Philosophy in Recreation and Leisure Studies.
https://uwspace.uwaterloo.ca/bitstream/handle/10012/10684/Guttentag_Daniel.pdf
5) *Airbnb Data Science Project*. (n.d.).
https://mohamedirfansh.github.io/Airbnb-Data-Science-Project/

7.0 Appendix

Link to the Google Collab:

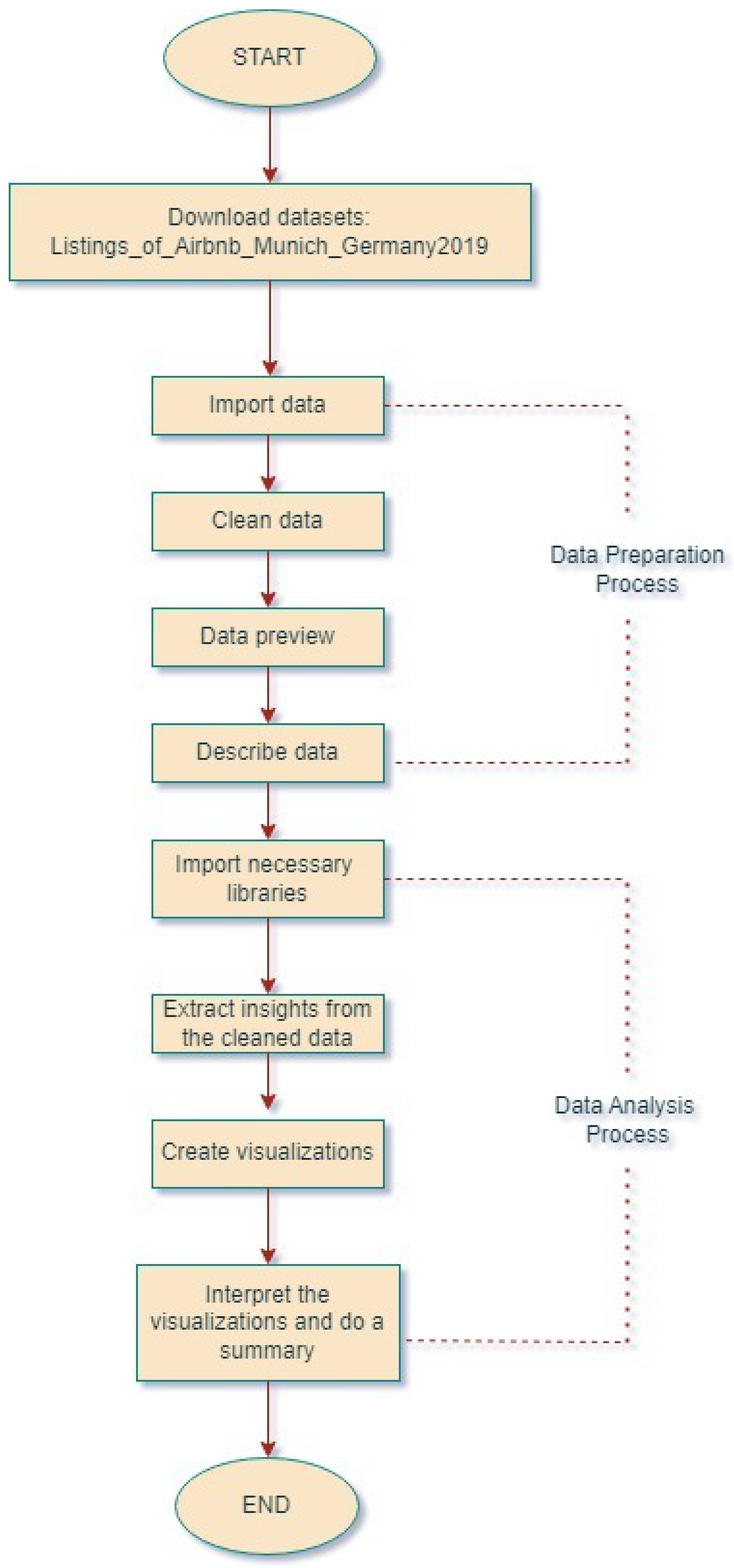
https://colab.research.google.com/drive/1RO-wm8MdKEHP6RJDKuxKqiWaQfSuQibp?usp=sharing

The data set were took from Kaggle and available in link below:

http://insideairbnb.com/get-the-data/

Data Preparation

```
from PIL import Image
img = Image.open("data preparation flowchart.jpeg")
display(img)
```



Data Import

```
import pandas as pd
data= pd.read_csv("listings_airbnb_munich.csv")
data
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_n
0	609851	Luxury 3 room flat close to Olympiapark	3024324	Christian	NaN	Maxvorstadt	48.15561	11.56736	Entire home/apt	200	
1	97945	Deluxw-Apartm. with roof terrace	517685	Angelika	NaN	Hadern	48.11492	11.48954	Entire home/apt	80	
2	114695	Apartment Munich/East with sundeck	581737	Stephan	NaN	Berg am Laim	48.12071	11.63758	Entire home/apt	95	
3	127383	City apartment next to Pinakothek	630556	Sonja	NaN	Maxvorstadt	48.15199	11.56482	Entire home/apt	120	
4	159634	Fancy, bright central roof top flat and homeof...	765694	Susana	NaN	Pasing-Obermenzing	48.13855	11.46586	Entire home/apt	60	
...	...	...	...	...	...	...	...	...	...	...	
5528	1560042	Frisch und freundlich in weiß-rot	1909709	Gabriele	NaN	Pasing-Obermenzing	48.15970	11.45158	Private room	38	
5529	1576125	Quiet and Relaxing Room with own Bath	8378646	Ralf	NaN	Allach-Untermenzing	48.17912	11.46728	Private room	38	
5530	1576417	Quiet and Relaxing Rooms with own Bath for 3Pe...	8378646	Ralf	NaN	Allach-Untermenzing	48.18063	11.46759	Private room	44	
5531	1583637	Comfortable & next to the Oktoberfest	3664843	Florian	NaN	Ludwigsvorstadt-Isarvorstadt	48.13296	11.55499	Private room	69	
5532	1585526	Tolle Altbauwohnung nahe Theresienwiese	7680306	Katharina	NaN	Schwanthalerhöhe	48.13850	11.52979	Private room	235	

5533 rows × 18 columns

```
num_entries = data.shape[0]
num_entries
```

```
5533
```

```
num_features = data.shape[1]
num_features
```

    18

```
data.dtypes
```

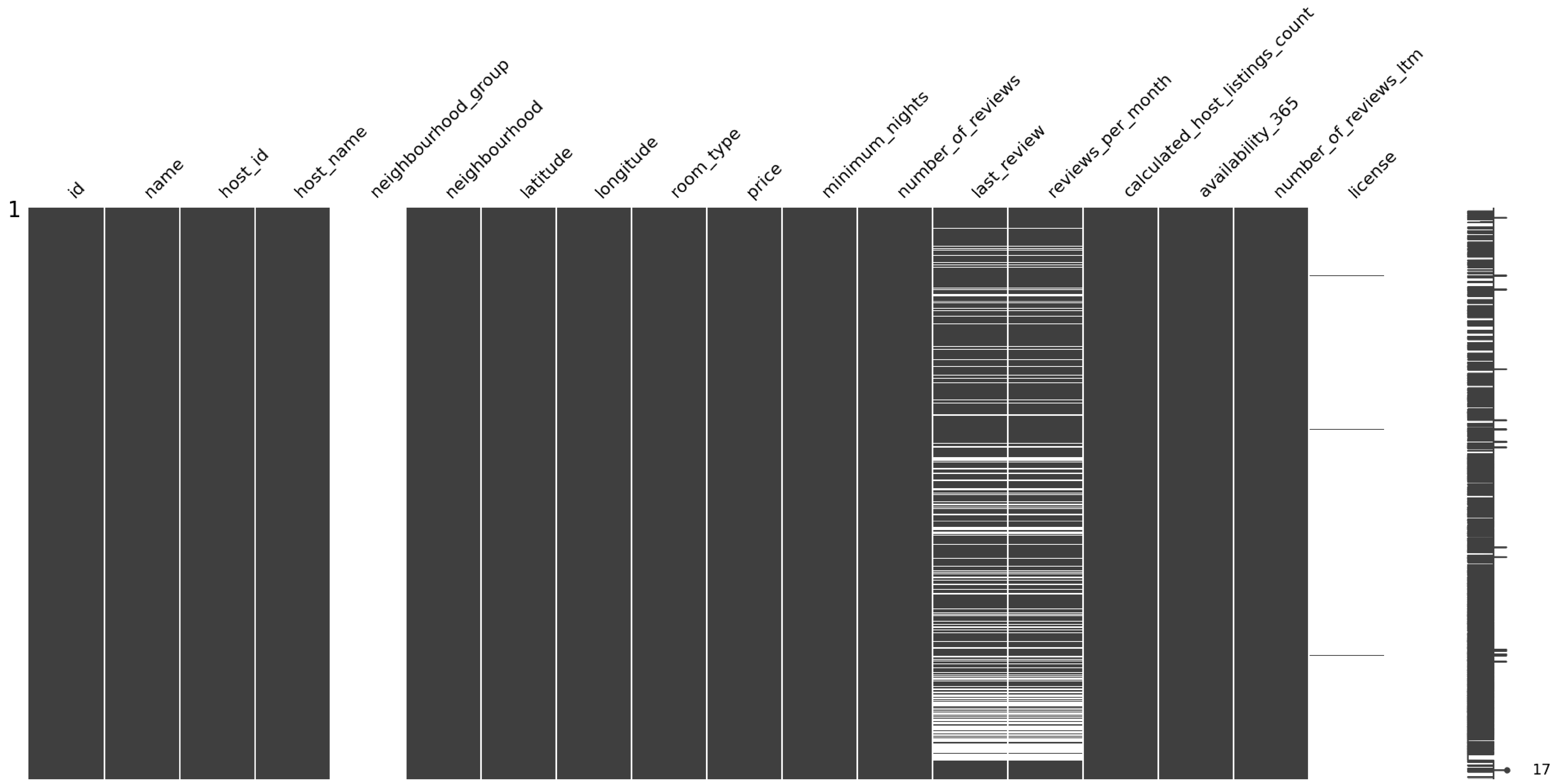
```
id                                int64
name                             object
host_id                           int64
host_name                        object
neighbourhood_group             float64
neighbourhood                    object
latitude                        float64
longitude                       float64
room_type                        object
price                             int64
minimum_nights                    int64
number_of_reviews                 int64
last_review                      object
reviews_per_month               float64
calculated_host_listings_count    int64
availability_365                  int64
number_of_reviews_ltm             int64
license                          object
dtype: object
```

Data Cleaning

Visualizing Missing Data

```
import missingno as msno
msno.matrix(data)
```

    <Axes: >




```
columns_to_drop = ['id', 'host_id', 'host_name', 'neighbourhood_group','number_of_reviews', 'last_review', 'reviews_per_month',
                   'calculated_host_listings_count', 'number_of_reviews_ltm', 'license']
new_data = data.drop(columns_to_drop, axis=1)
new_data
```

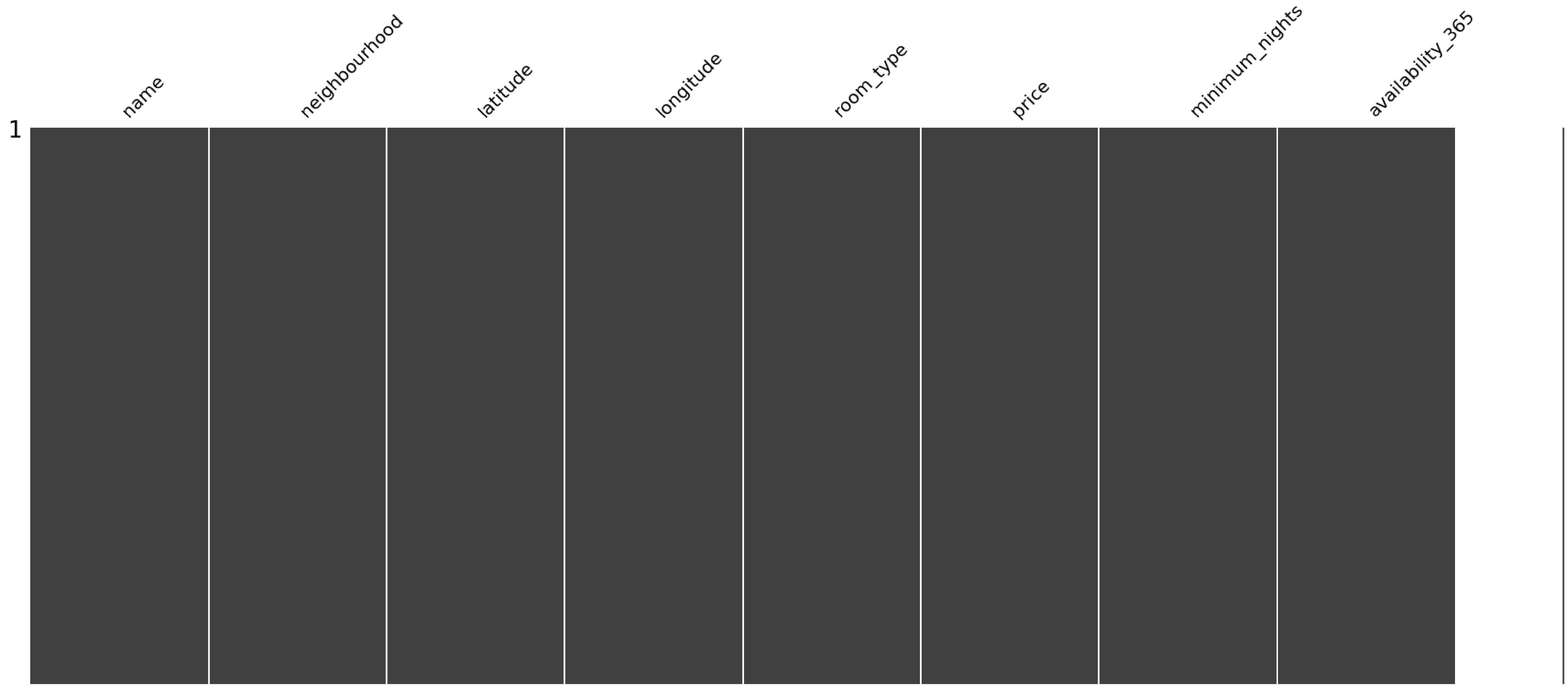
	name	neighbourhood	latitude	longitude	room_type	price	minimum_nights	availability_365
0	Luxury 3 room flat close to Olympiapark	Maxvorstadt	48.15561	11.56736	Entire home/apt	200	5	129
1	Deluxw-Apartm. with roof terrace	Hadern	48.11492	11.48954	Entire home/apt	80	2	86
2	Apartment Munich/East with sundeck	Berg am Laim	48.12071	11.63758	Entire home/apt	95	2	140
3	City apartment next to Pinakothek	Maxvorstadt	48.15199	11.56482	Entire home/apt	120	3	0
4	Fancy, bright central roof top flat and homeof...	Pasing-Obermenzing	48.13855	11.46586	Entire home/apt	60	2	1
...	...	...	...	...	...	...	...	...
5528	Frisch und freundlich in weiß-rot	Pasing-Obermenzing	48.15970	11.45158	Private room	38	1	117
5529	Quiet and Relaxing Room with own Bath	Allach-Untermenzing	48.17912	11.46728	Private room	38	1	0
5530	Quiet and Relaxing Rooms with own Bath for 3Pe...	Allach-Untermenzing	48.18063	11.46759	Private room	44	1	0
5531	Comfortable & next to the Oktoberfest	Ludwigsvorstadt-Isarvorstadt	48.13296	11.55499	Private room	69	2	0
5532	Tolle Altbauwohnung nahe Theresienwiese	Schwanthalerhöhe	48.13850	11.52979	Private room	235	3	365

5533 rows × 8 columns

For a cleaner and more objective analysis, these variables had been deleted since the variables that dropped will not be used in our analysis.

```
import missingno as msno
msno.matrix(new_data)
```

```
<Axes: >
```



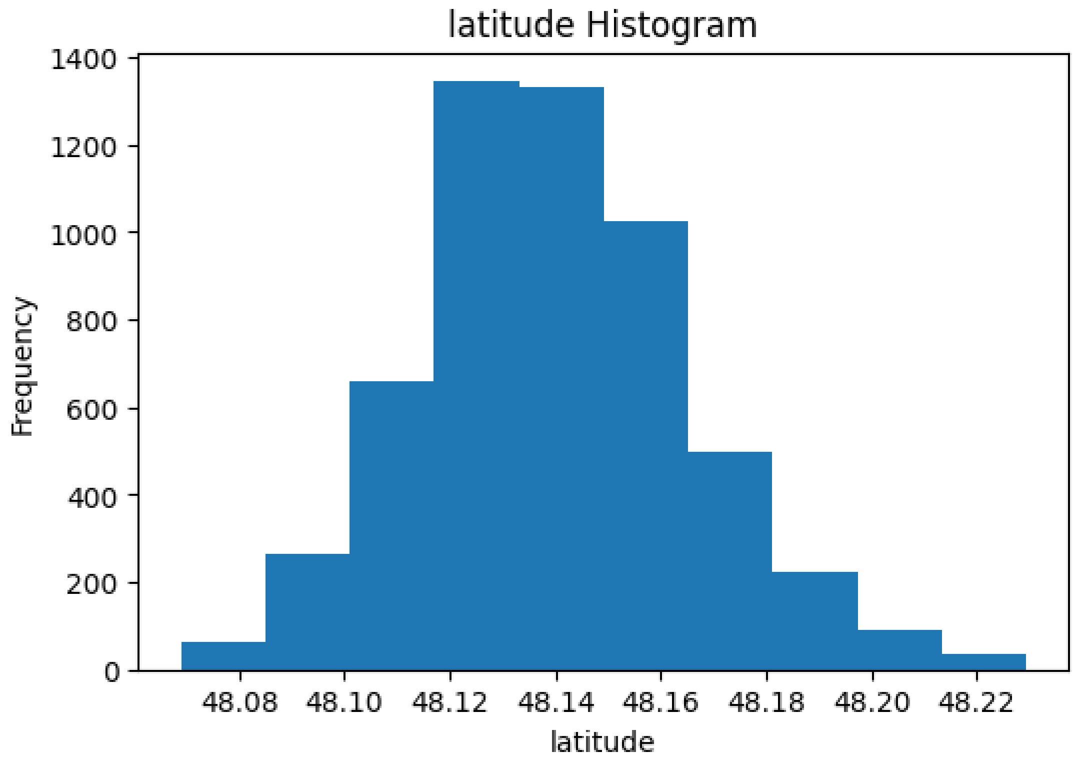

```
import pandas as pd
missing_percentage = new_data.isnull().mean()*100
missing_percentage
```

```
name                0.0
neighbourhood       0.0
latitude            0.0
longitude           0.0
room_type           0.0
price               0.0
minimum_nights      0.0
availability_365    0.0
dtype: float64
```

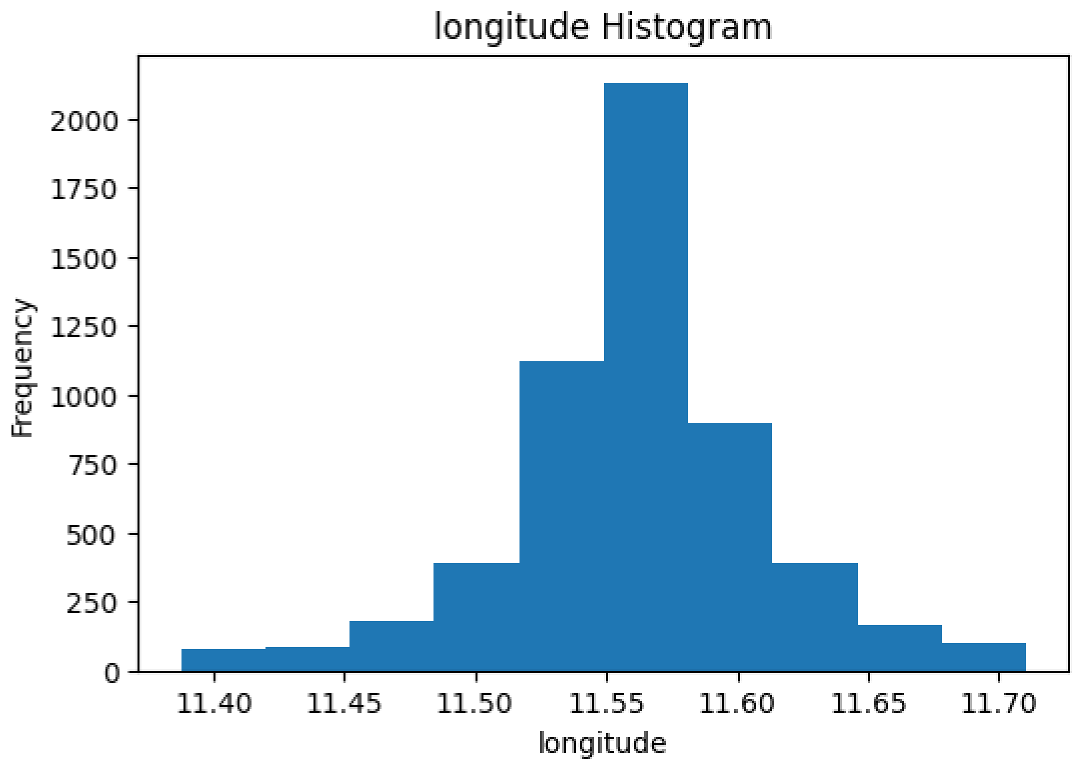
This dataset has practically no null values, therefore no null values treatment will be performed.

Outliers Detection and Treatment

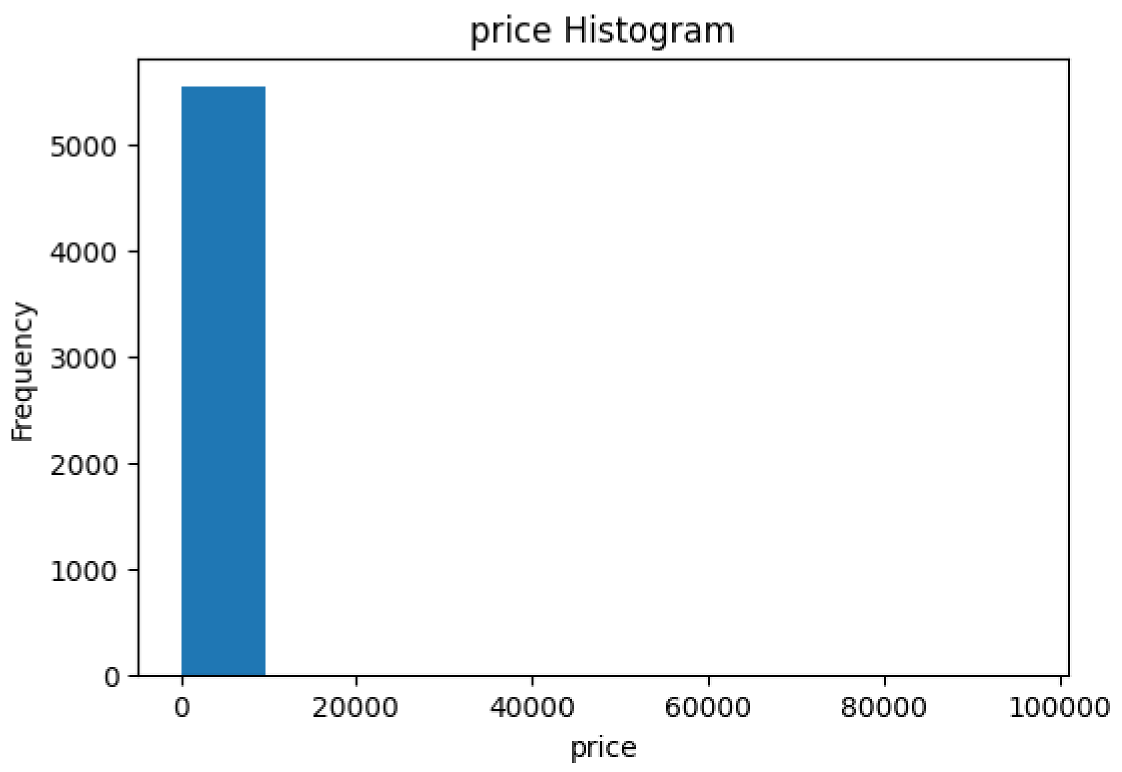
```
import pandas as pd
import matplotlib.pyplot as plt


plt.figure(figsize=(6, 4))
plt.hist(new_data['latitude'])
plt.title("latitude" + " Histogram")
plt.xlabel("latitude")
plt.ylabel("Frequency")
plt.show()
```



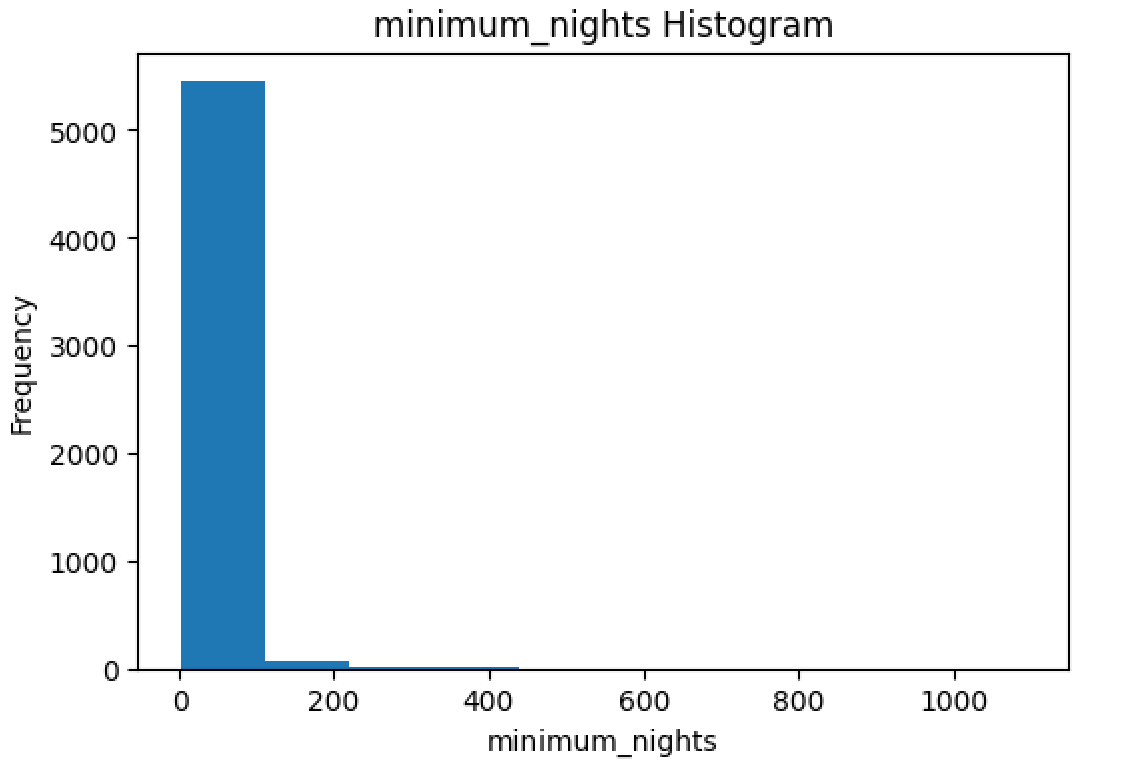

```
plt.figure(figsize=(6, 4))
plt.hist(new_data['longitude'])
plt.title("longitude" + " Histogram")
plt.xlabel("longitude")
plt.ylabel("Frequency")
plt.show()
```



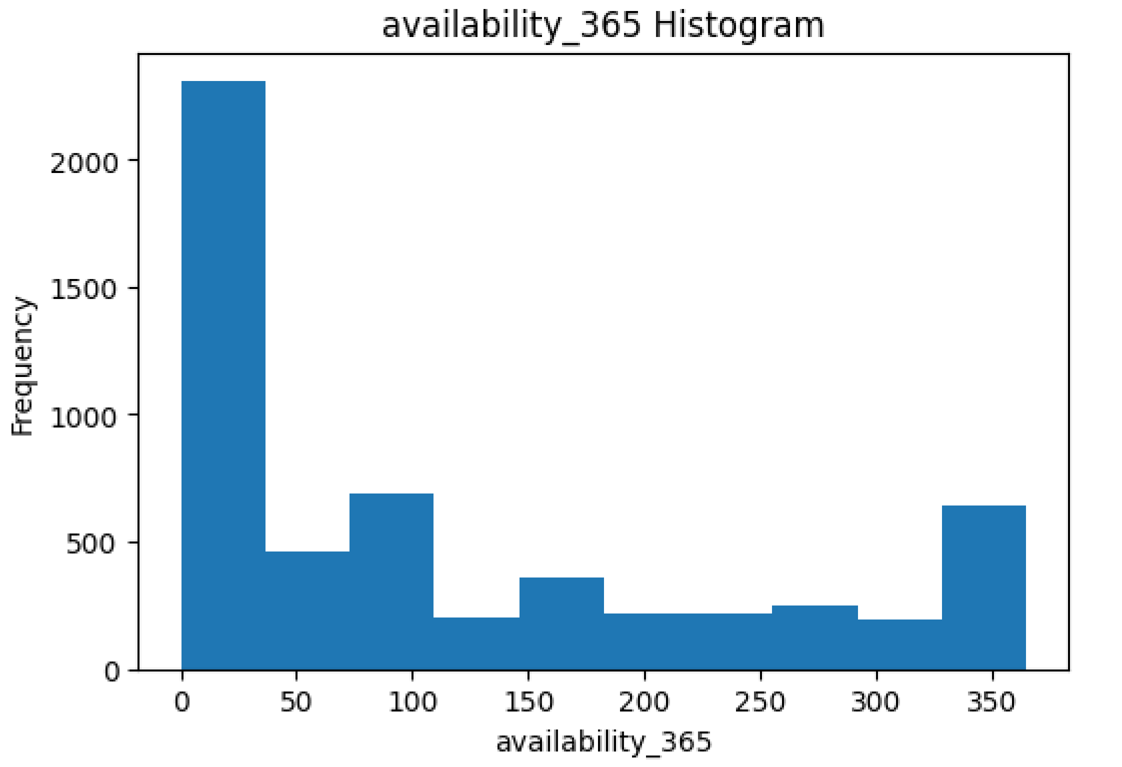
```
plt.figure(figsize=(6,4))
plt.hist(new_data['price'])
plt.title("price" + " Histogram")
plt.xlabel("price")
plt.ylabel("Frequency")

plt.show()
```




```
plt.figure(figsize=(6, 4))
plt.hist(new_data['minimum_nights'])
plt.title("minimum_nights" + " Histogram")
plt.xlabel("minimum_nights")
plt.ylabel("Frequency")
plt.show()
```




```
plt.figure(figsize=(6, 4))
plt.hist(new_data['availability_365'])
plt.title("availability_365" + " Histogram")
plt.xlabel("availability_365")
plt.ylabel("Frequency")
plt.show()
```



Through the five histogram, it is possible to verify the presence of outliers in the variables 'price' and 'minimum_nights'. The values do not follow a distribution and distort the entire graphical presentation.

```
new_data.describe()
```

	latitude	longitude	price	minimum_nights	availability_365
count	5533.000000	5533.000000	5533.000000	5533.000000	5533.000000
mean	48.139611	11.562330	170.836978	9.009398	114.627327
std	0.025659	0.048663	1308.536534	31.365482	125.219271
min	48.068870	11.387475	0.000000	1.000000	0.000000
25%	48.122560	11.538820	66.000000	1.000000	0.000000
50%	48.137080	11.564030	100.000000	2.000000	72.000000
75%	48.155700	11.585310	167.000000	4.000000	206.000000
max	48.229500	11.710610	96274.000000	1095.000000	365.000000

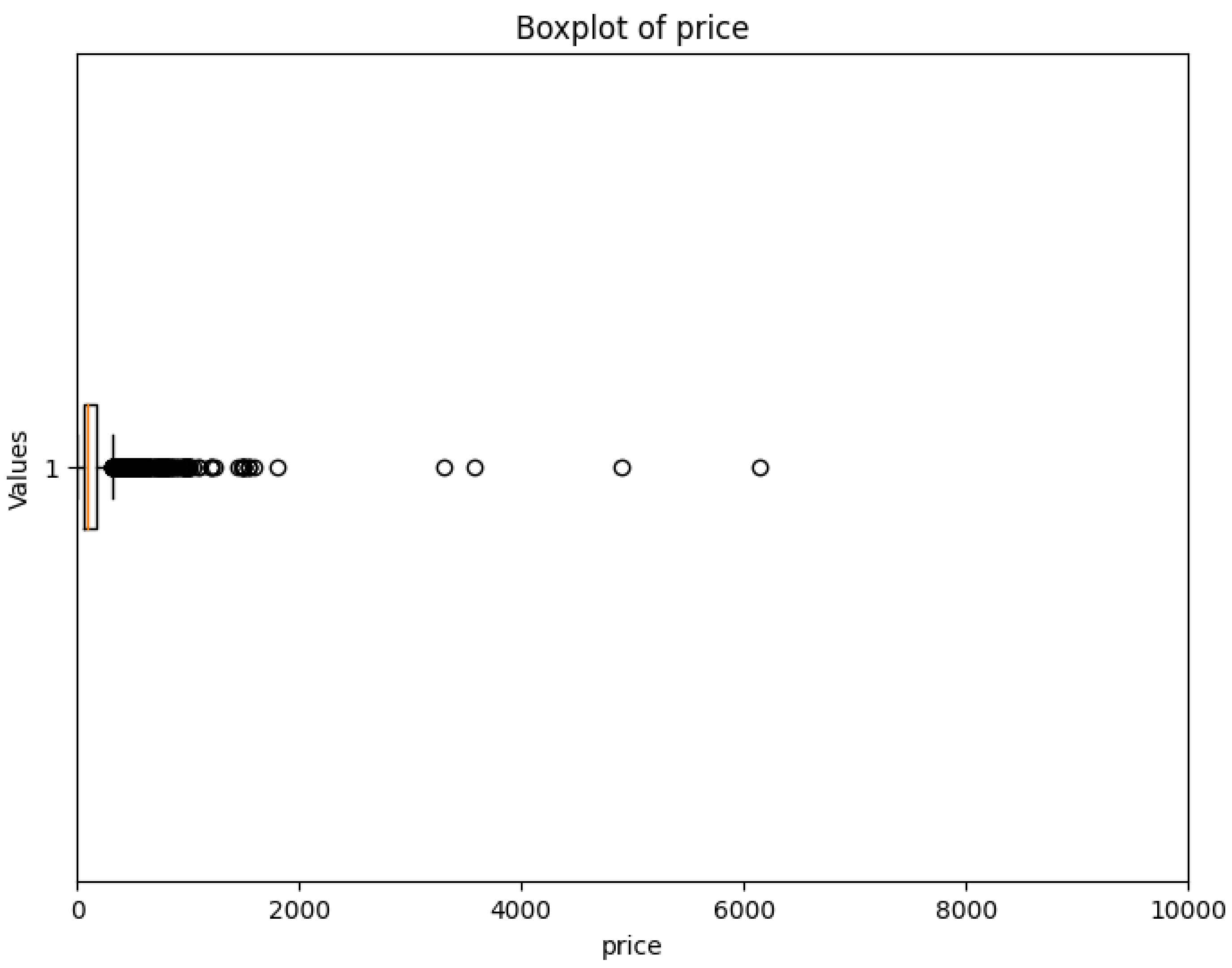
The variable 'price' has 75% of its values below 167, but its maximum value is 96274, which is well above the values obtained up to 75% of the dataset, which proves the presence of outliers.

The variable 'price' presents minimum values equal to 0. Understanding the Airbnb business, it is known that no one rents any property on Airbnb for free.

The variable 'minimum_nights' has 75% of its values below 4, but its maximum value is 1095, which is well above the values obtained up to 75% of the dataset, which proves the presence of outliers.

Boxplot for "Price"

```
plt.figure(figsize=(8, 6))
plt.boxplot(new_data['price'], vert=False)
plt.title('Boxplot of ' + 'price')
plt.xlabel('price')
plt.ylabel('Values')
plt.xlim(0,10000)
plt.show()
```




```
import numpy as np
import pandas as pd

outliers = new_data[new_data['price'] > 1000]
outliers_count = np.sum(new_data['price'] > 1000)
outliers_ratio = round(outliers_count / len(new_data), 3)

print("Outliers:")
print(outliers)
print("\nQuantity of Outliers:", outliers_count)
print("Outliers Ratio:", outliers_ratio)
```

```
5217      Great flat for Oktoberfest - for couples or 4

                                    neighbourhood   latitude  longitude  \
251                                  Tudering-Riem  48.130710  11.695030
453                               Schwanthalerhöhe  48.130360  11.543080
1722                                Altstadt-Lehel  48.134160  11.576950
1864                              Sendling-Westpark  48.108340  11.526620
1875                                 Au-Haidhausen  48.120650  11.579480
2558                                   Maxvorstadt  48.150850  11.572580
3029                         Neuhausen-Nymphenburg  48.151790  11.530910
3182                                      Sendling  48.123690  11.546420
3382                  Ludwigsvorstadt-Isarvorstadt  48.133950  11.553670
3416                                 Schwabing-West  48.168130  11.581460
3520                                  Berg am Laim  48.124082  11.605817
3583                                 Au-Haidhausen  48.128353  11.596788
3646                  Ludwigsvorstadt-Isarvorstadt  48.132094  11.566024
3999                  Ludwigsvorstadt-Isarvorstadt  48.136090  11.558870
4114                  Ludwigsvorstadt-Isarvorstadt  48.126840  11.558060
4133                                   Maxvorstadt  48.146160  11.546230
4616                         Milbertshofen-Am Hart  48.188591  11.558877
4679  Thalkirchen-Obersendling-Forstenried-Fürstenri...  48.077360  11.516680
4826  Thalkirchen-Obersendling-Forstenried-Fürstenri...  48.075393  11.517271
4858                         Neuhausen-Nymphenburg  48.158954  11.555540
5017                                       Moosach  48.170482  11.513101
5018                                       Moosach  48.170607  11.514607
5019                                       Moosach  48.170540  11.513202
5072                                Altstadt-Lehel  48.137896  11.576529
5217                                   Obergiesing  48.116220  11.582400

            room_type  price  minimum_nights  availability_365
251   Entire home/apt   1050               5               192
```

```
    Outliers Ratio: 0.005
```

```
import numpy as np
import pandas as pd

zero_price_values = new_data[new_data['price'] == 0]

print("\nValues where 'price' is equal to 0:")
print(zero_price_values)

outliers_count = np.sum(new_data['price'] == 0)
print("\nQuantity of Outliers:", outliers_count)

outliers_ratio = round(outliers_count / len(new_data), 3)
print("Outliers Ratio:", outliers_ratio)
```

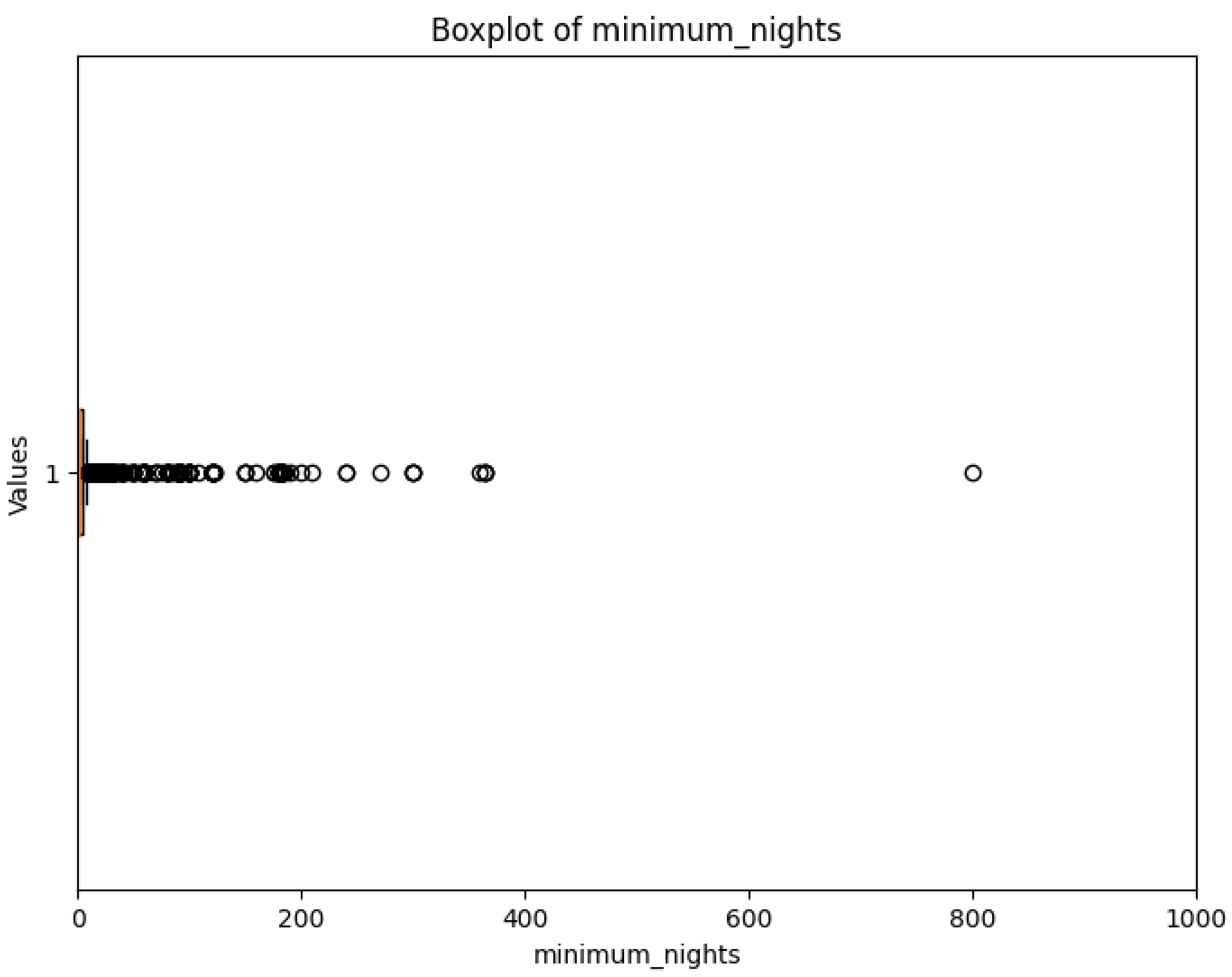
```
    Values where 'price' is equal to 0:
                          name     neighbourhood  latitude  longitude   room_type  \
    2123  Boutique Hotel Krone  Schwanthalerhöhe  48.13547   11.54717  Hotel room

          price  minimum_nights  availability_365
    2123      0               1                 0

    Quantity of Outliers: 1
    Outliers Ratio: 0.0
```

Above, the boxplot for the variable 'price' visually shows the information contained in the summary statistics.

As a parameter, all data greater than 1000 will be considered an outlier in this analysis. Above also, we will see the quantity and ratio of these outliers and the values where 'price' is equal to 0.

Boxplot for "minimum_nights"

```
plt.figure(figsize=(8, 6))
plt.boxplot(new_data['minimum_nights'], vert=False)
plt.title('Boxplot of ' + 'minimum_nights')
plt.xlabel('minimum_nights')
plt.ylabel('Values')
plt.xlim(0,1000)

plt.show()
```




```
import numpy as np
import pandas as pd

outliers = new_data[new_data['price'] > 30]

outliers_count = np.sum(new_data['price'] > 30)
outliers_ratio = round(outliers_count / len(new_data), 3)

print("Outliers:")
print(outliers)
print("\nQuantity of Outliers:", outliers_count)
print("Outliers Ratio:", outliers_ratio)
```

```
    Outliers:
                                                       name  \
    0               Luxury 3 room flat close to Olympiapark
    1                      Deluxw-Apartm. with roof terrace
    2                   Apartment Munich/East with sundeck
    3                     City apartment next to Pinakothek
    4     Fancy, bright central roof top flat and homeof...
    ...                                                 ...
    5528                  Frisch und freundlich in weiß-rot
    5529              Quiet and Relaxing Room with own Bath
    5530  Quiet and Relaxing Rooms with own Bath for 3Pe...
    5531              Comfortable & next to the Oktoberfest
    5532            Tolle Altbauwohnung nahe Theresienwiese

                      neighbourhood  latitude  longitude        room_type  \
    0                   Maxvorstadt  48.15561   11.56736  Entire home/apt
    1                        Hadern  48.11492   11.48954  Entire home/apt
    2                  Berg am Laim  48.12071   11.63758  Entire home/apt
    3                   Maxvorstadt  48.15199   11.56482  Entire home/apt
    4             Pasing-Obermenzing  48.13855   11.46586  Entire home/apt
    ...                         ...       ...        ...              ...
    5528          Pasing-Obermenzing  48.15970   11.45158     Private room
    5529         Allach-Untermenzing  48.17912   11.46728     Private room
    5530         Allach-Untermenzing  48.18063   11.46759     Private room
    5531  Ludwigsvorstadt-Isarvorstadt  48.13296   11.55499     Private room
    5532            Schwanthalerhöhe  48.13850   11.52979     Private room

          price  minimum_nights  availability_365
    0       200               5               129
```

```
1        80               2                86
2        95               2               140
3       120               3                 0
4        60               2                 1
...     ...             ...               ...
5528     38               1               117
5529     38               1                 0
5530     44               1                 0
5531     69               2                 0
5532    235               3               365

[5420 rows x 8 columns]

Quantity of Outliers: 5420
Outliers Ratio: 0.98
```

Above, the boxplot for the variable 'minimum_nights' visually shows the information contained in the summary statistics.

As a parameter, all data greater than 30 will be considered an outlier in this analysis. Above also, we will see the quantity and ratio of these outliers.

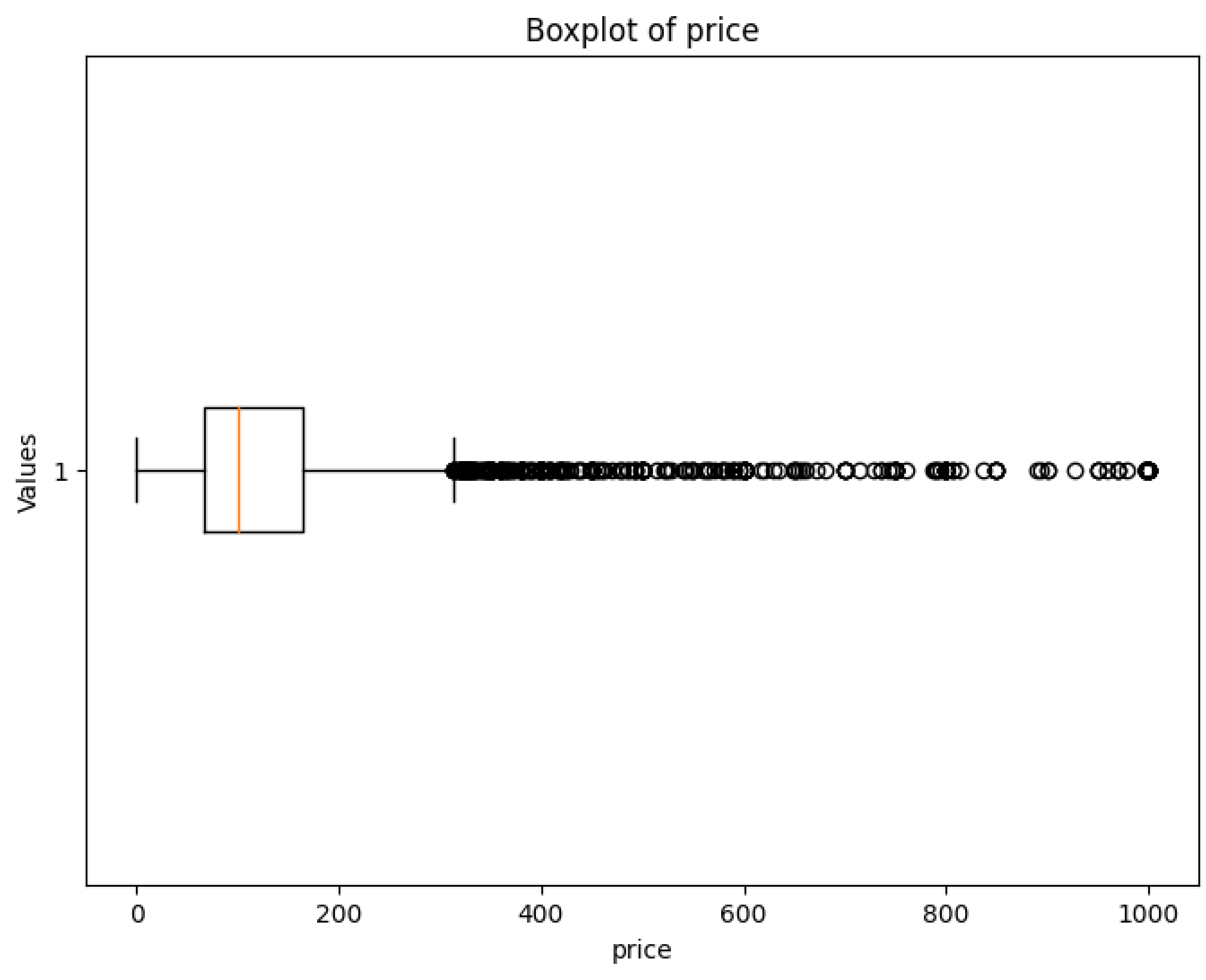
We had set parameters for both of them in which will be considered outliers since the amount of outliers are too many.

Removing Outliers and Creating Data Frame for Analysis

```
outliers = new_data[new_data['price'] > 1000]

cleaned_data = new_data.drop(outliers.index)

print("Cleaned Data:")
print(cleaned_data)
```

```
Cleaned Data:
                                                  name  \
0                Luxury 3 room flat close to Olympiapark
1                      Deluxw-Apartm. with roof terrace
2                    Apartment Munich/East with sundeck
3                     City apartment next to Pinakothek
4     Fancy, bright central roof top flat and homeof...
...                                                 ...
5528                    Frisch und freundlich in weiß-rot
5529                Quiet and Relaxing Room with own Bath
5530  Quiet and Relaxing Rooms with own Bath for 3Pe...
5531                Comfortable & next to the Oktoberfest
5532              Tolle Altbauwohnung nahe Theresienwiese

                   neighbourhood  latitude  longitude        room_type  \
0                    Maxvorstadt  48.15561   11.56736  Entire home/apt
1                         Hadern  48.11492   11.48954  Entire home/apt
2                   Berg am Laim  48.12071   11.63758  Entire home/apt
3                    Maxvorstadt  48.15199   11.56482  Entire home/apt
4              Pasing-Obermenzing  48.13855   11.46586  Entire home/apt
...                          ...       ...        ...              ...
5528           Pasing-Obermenzing  48.15970   11.45158     Private room
5529          Allach-Untermenzing  48.17912   11.46728     Private room
5530          Allach-Untermenzing  48.18063   11.46759     Private room
5531  Ludwigsvorstadt-Isarvorstadt  48.13296   11.55499     Private room
5532             Schwanthalerhöhe  48.13850   11.52979     Private room

      price  minimum_nights  availability_365
0       200               5               129
1        80               2                86
2        95               2               140
3       120               3                 0
4        60               2                 1
...     ...             ...               ...
5528     38               1               117
5529     38               1                 0
5530     44               1                 0
5531     69               2                 0
5532    235               3               365

[5508 rows x 8 columns]
```

```
plt.figure(figsize=(8, 6))
plt.boxplot(cleaned_data['price'], vert=False)
plt.title('Boxplot of ' + 'price')
plt.xlabel('price')
plt.ylabel('Values')
plt.show()
```

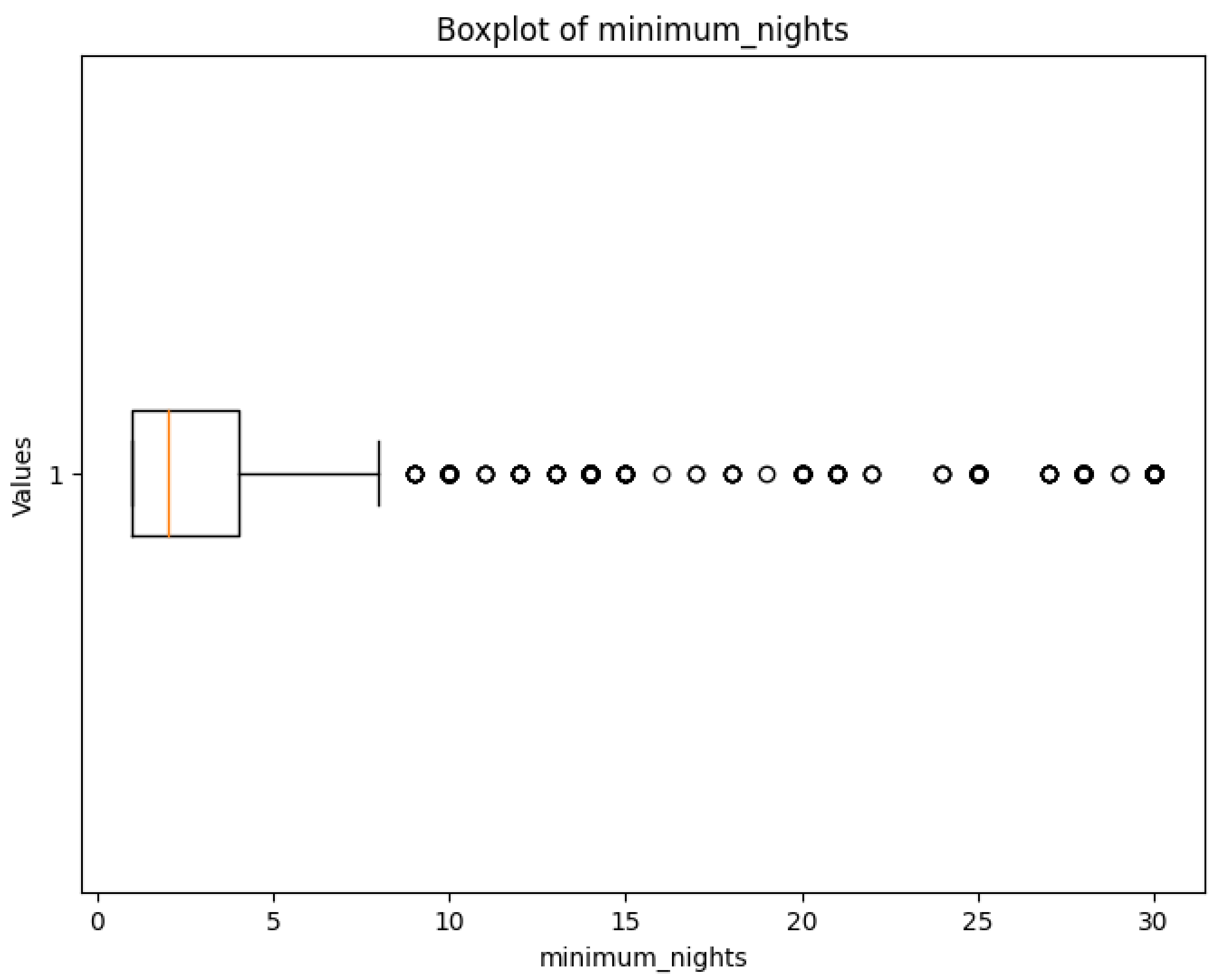



```
outliers = new_data[new_data['minimum_nights'] > 30]

# Remove outliers from the DataFrame
cleaned_data = new_data.drop(outliers.index)
```

```
# Display the cleaned data
print("Cleaned Data:")
print(cleaned_data)
```

```
Cleaned Data:
                                                 name  \
0               Luxury 3 room flat close to Olympiapark
1                    Deluxw-Apartm. with roof terrace
2                 Apartment Munich/East with sundeck
3                   City apartment next to Pinakothek
4     Fancy, bright central roof top flat and homeof...
...                                                 ...
5528                    Frisch und freundlich in weiß-rot
5529              Quiet and Relaxing Room with own Bath
5530  Quiet and Relaxing Rooms with own Bath for 3Pe...
5531              Comfortable & next to the Oktoberfest
5532            Tolle Altbauwohnung nahe Theresienwiese

                  neighbourhood  latitude  longitude        room_type  \
0                   Maxvorstadt  48.15561   11.56736  Entire home/apt
1                        Hadern  48.11492   11.48954  Entire home/apt
2                  Berg am Laim  48.12071   11.63758  Entire home/apt
3                   Maxvorstadt  48.15199   11.56482  Entire home/apt
4             Pasing-Obermenzing  48.13855   11.46586  Entire home/apt
...                         ...       ...        ...              ...
5528          Pasing-Obermenzing  48.15970   11.45158     Private room
5529          Allach-Untermenzing  48.17912   11.46728     Private room
5530          Allach-Untermenzing  48.18063   11.46759     Private room
5531  Ludwigsvorstadt-Isarvorstadt  48.13296   11.55499     Private room
5532              Schwanthalerhöhe  48.13850   11.52979     Private room

      price  minimum_nights  availability_365
0       200               5               129
1        80               2                86
2        95               2               140
3       120               3                 0
4        60               2                 1
...     ...             ...               ...
5528     38               1               117
5529     38               1                 0
5530     44               1                 0
5531     69               2                 0
5532    235               3               365

[5319 rows x 8 columns]
```

```
plt.figure(figsize=(8, 6))
plt.boxplot(cleaned_data['minimum_nights'], vert=False)
plt.title('Boxplot of ' + 'minimum_nights')
plt.xlabel('minimum_nights')
plt.ylabel('Values')
plt.show()
```



Finally, with the clean data frame created and treated, the analysis begins.

```
cleaned_data.describe()
```

	latitude	longitude	price	minimum_nights	availability_365
count	5319.000000	5319.000000	5319.000000	5319.000000	5319.000000
mean	48.139530	11.562366	173.774206	4.480165	112.395187
std	0.025685	0.048759	1334.399578	6.571928	124.739760
min	48.068870	11.387475	0.000000	1.000000	0.000000
25%	48.122391	11.538725	68.000000	1.000000	0.000000
50%	48.137030	11.563940	100.000000	2.000000	67.000000
75%	48.155565	11.585520	170.000000	4.000000	202.000000
max	48.229500	11.710610	96274.000000	30.000000	365.000000

Exploratory Data Analysis

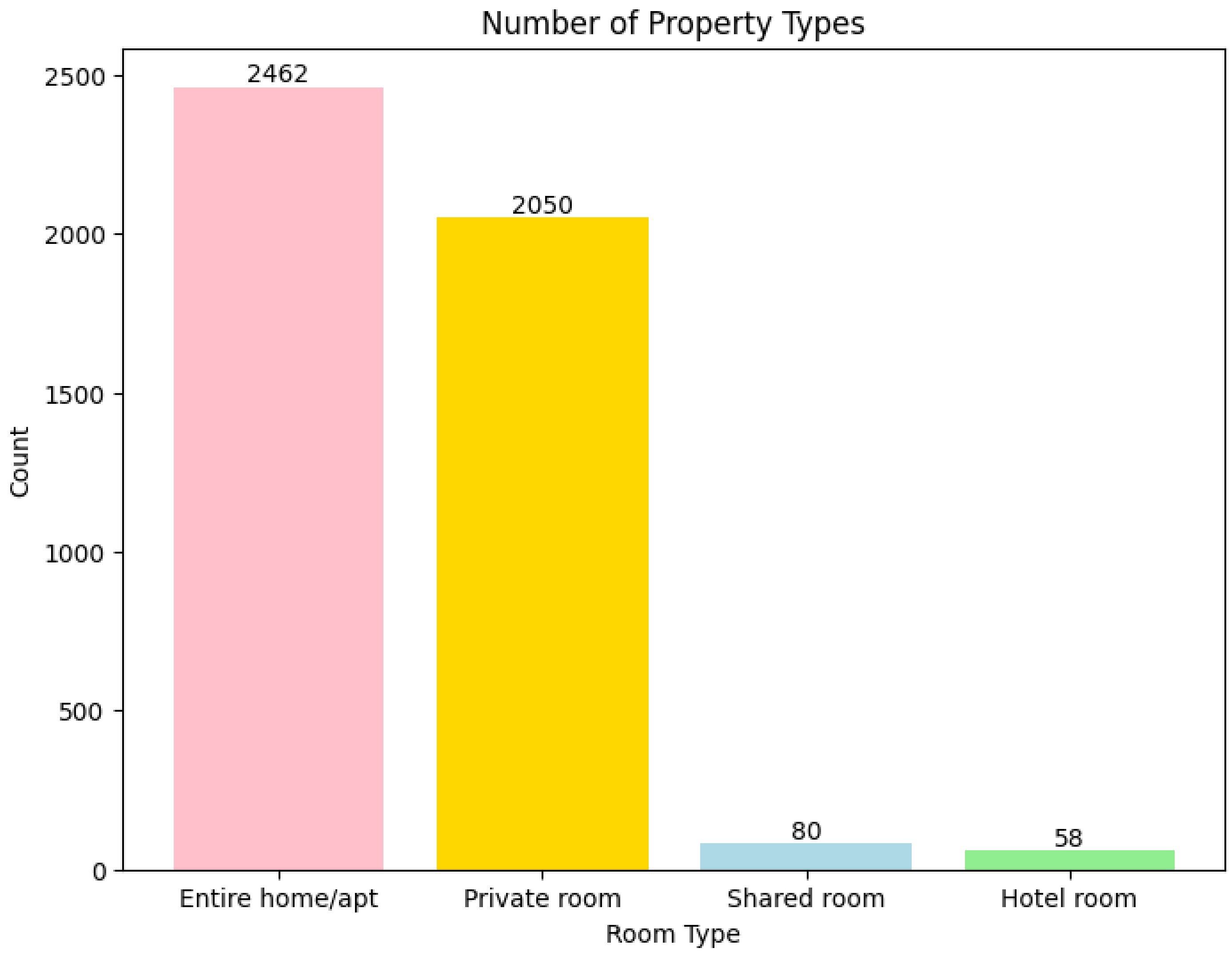
Objective 1:

```
import pandas as pd
property_counts = new_data['room_type'].value_counts()
property_ratios = (property_counts / len(new_data) * 100).round(2).astype(str) + '%'

print("Number of Property Types:")
print(property_counts)
```

```
print()

print("Ratio of Property Types:")
print(property_ratios)
```

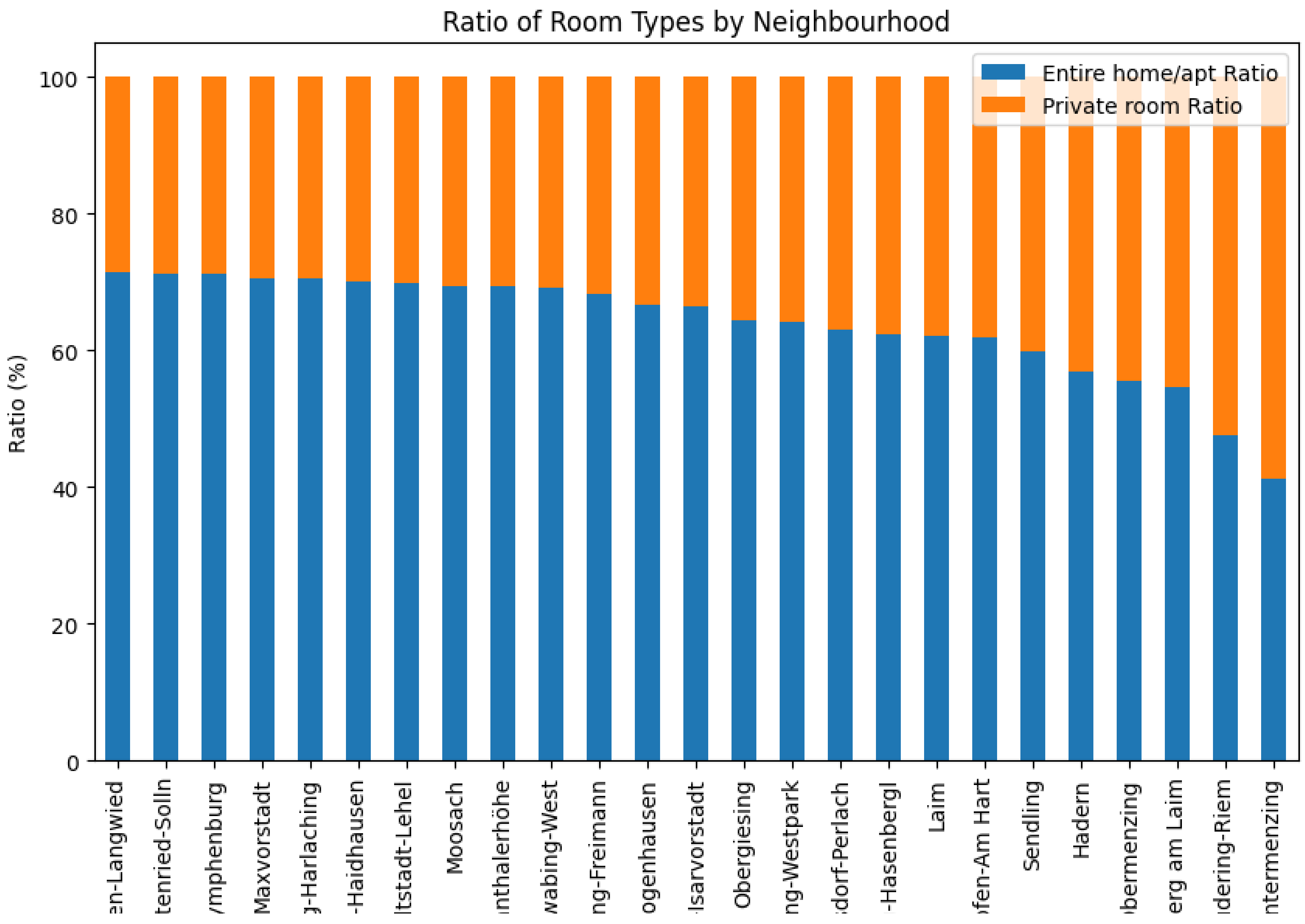
```
Number of Property Types:
Entire home/apt    3591
Private room       1853
Shared room          52
Hotel room           37
Name: room_type, dtype: int64

Ratio of Property Types:
Entire home/apt     64.9%
Private room       33.49%
Shared room         0.94%
Hotel room          0.67%
Name: room_type, dtype: object
```

```
import matplotlib.pyplot as plt

room_types = ['Entire home/apt', 'Private room', 'Shared room', 'Hotel room']
counts = [2462, 2050, 80, 58]
colors = ['#FFC0CB', '#FFD700', '#ADD8E6', '#90EE90']

plt.figure(figsize=(8, 6))
plt.bar(room_types, counts, color=colors)

for i, count in enumerate(counts):
    plt.text(i, count, str(count), ha='center', va='bottom')

plt.title("Number of Property Types")
plt.xlabel("Room Type")
plt.ylabel("Count")

plt.show()
```



As the property types 'Shared room' and 'Hotel room' are not relevant to the number of properties being rented on Airbnb in the city of Munich, we will continue this analysis in the neighbourhoods using only the property types 'Entire home/apt' and 'Private room'. In Munich, Germany, hotel rooms and shared rooms might not be as common as other property types for a variety of reasons. First off, shared rooms often offer little in the way of personal space or privacy, which may not be what most travellers like. Munich is a well-liked travel destination, drawing a variety of tourists who frequently seek out more secluded and pleasant lodgings. Additionally, Munich has a large supply of hotels with a variety of alternatives and amenities, which could reduce the demand for hotels with listings on Airbnb. Additionally, Munich's cultural tastes and accepted travel practices may have an impact on how popular shared rooms and hotel rooms are, with a stronger desire for complete homes or private rooms that offer a more individualised and opulent experience.

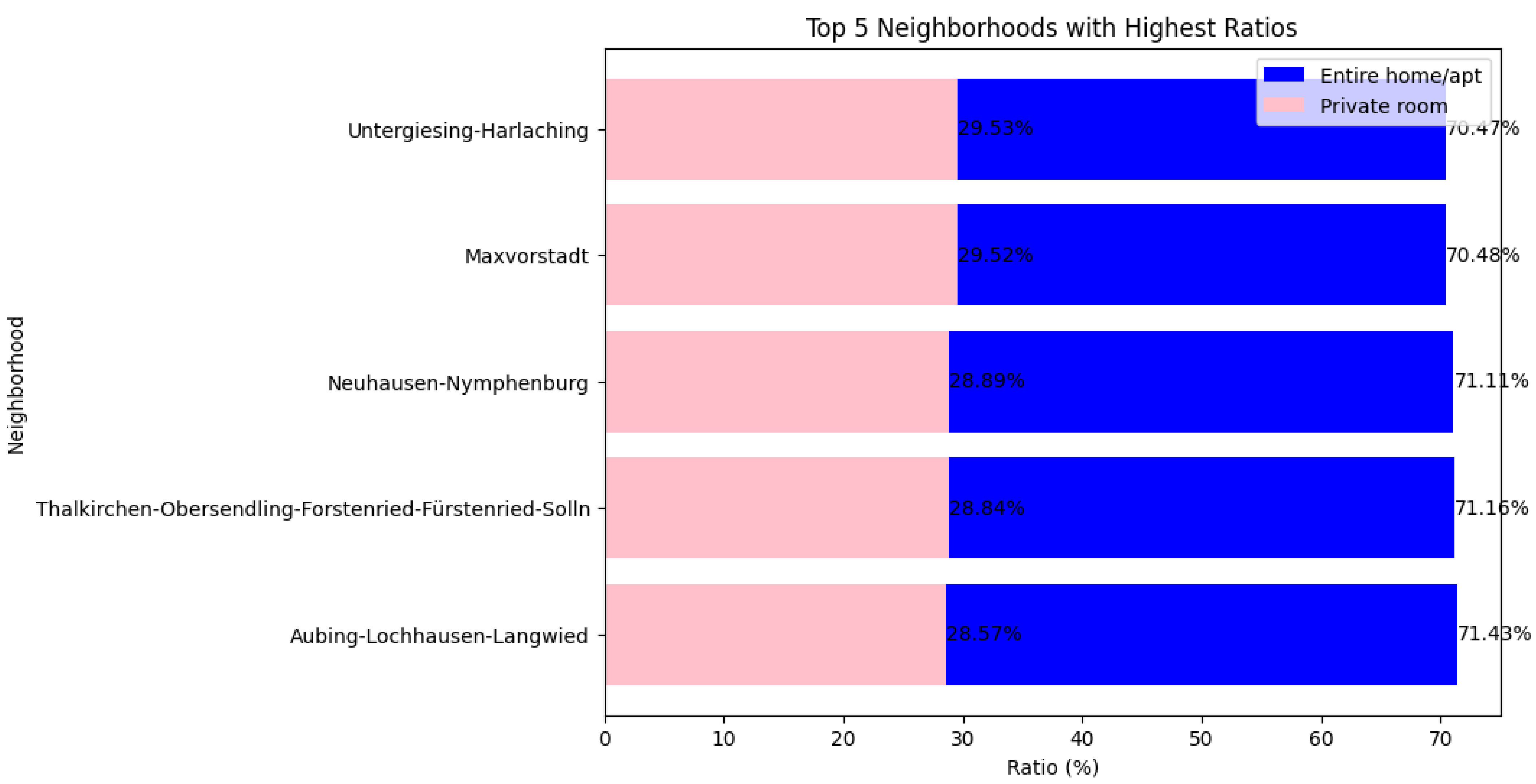
```
import matplotlib.pyplot as plt

grouped_data = grouped_data.sort_values(by='Entire home/apt Ratio', ascending=False)

fig, ax = plt.subplots(figsize=(10, 6))
grouped_data[['Entire home/apt Ratio', 'Private room Ratio']].plot(kind='bar', stacked=True, ax=ax)

plt.title('Ratio of Room Types by Neighbourhood')
plt.xlabel('Neighbourhood')
plt.ylabel('Ratio (%)')

plt.legend()

plt.xticks(rotation=90)

plt.show()
```



It can be seen that the distribution of property types in Munich is well balanced in most neighbourhoods. This indicates that both types of accommodations are available and in demand, providing a diverse range of options for travelers. A well-balanced distribution of property types can be beneficial for both hosts and guests. Hosts have the flexibility to offer different types of accommodations based on their property and preferences, while guests have the opportunity to choose the type of accommodation that suits their needs and preferences.

Below we can see the neighbourhoods where the proportion of property type 'Entire home/apt' is higher, so if the Airbnb user wants to stay in one of the neighbourhoods listed below, there is a greater chance that this user will find offers of this type property available.

```
import matplotlib.pyplot as plt


top_5_neighborhoods = grouped_data.head(5)


plt.figure(figsize=(8, 6))
plt.barh(top_5_neighborhoods.index, top_5_neighborhoods['Entire home/apt Ratio'], color='blue', label='Entire home/apt')
plt.barh(top_5_neighborhoods.index, top_5_neighborhoods['Private room Ratio'], color='pink', label='Private room')


for i, neighborhood in enumerate(top_5_neighborhoods.index):
    ratio_entire = top_5_neighborhoods.loc[neighborhood, 'Entire home/apt Ratio']
    ratio_private = top_5_neighborhoods.loc[neighborhood, 'Private room Ratio']
    plt.text(ratio_entire, i, f'{ratio_entire}%', va='center', color='black')
    plt.text(ratio_private, i, f'{ratio_private}%', va='center', color='black')


plt.title("Top 5 Neighborhoods with Highest Ratios")
plt.xlabel("Ratio (%)")
plt.ylabel("Neighborhood")
plt.legend()


plt.show()
```

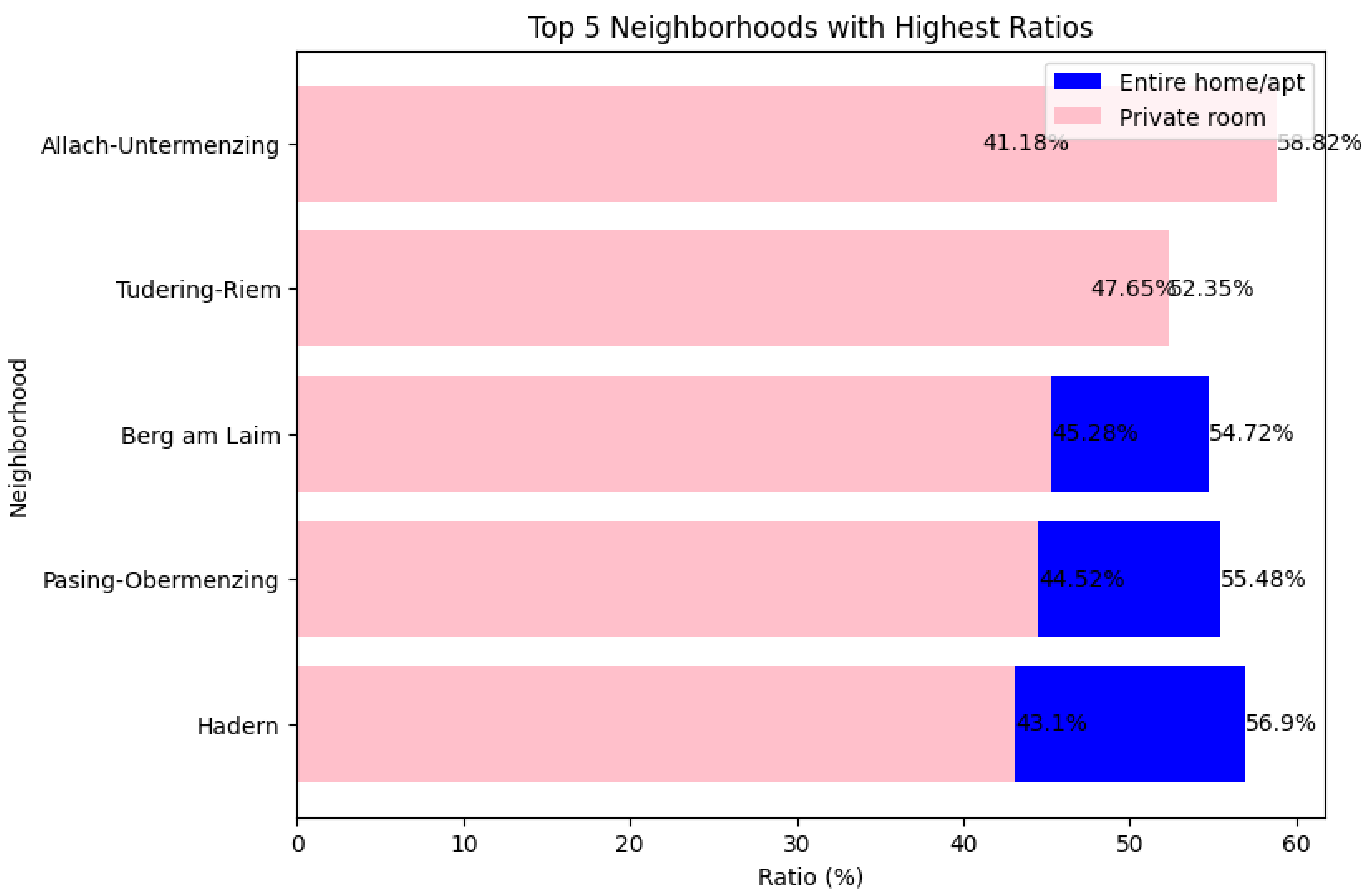


These neighbourhoods in Munich may have a higher concentration of residential properties or property owners who are more inclined to rent out their entire homes/apartments rather than individual rooms.

Now, we can see the neighbourhoods where the proportion of property type 'Private room' is higher, so if the Airbnb user wants to stay in one of the neighbourhoods listed below, there is a greater chance that this user will find offers for this type of property available.

```
import matplotlib.pyplot as plt

top_5_neighborhoods = grouped_data.tail(5)
```

```
plt.figure(figsize=(8, 6))
plt.barh(top_5_neighborhoods.index, top_5_neighborhoods['Entire home/apt Ratio'], color='blue', label='Entire home/apt')
plt.barh(top_5_neighborhoods.index, top_5_neighborhoods['Private room Ratio'], color='pink', label='Private room')

for i, neighborhood in enumerate(top_5_neighborhoods.index):
    ratio_entire = top_5_neighborhoods.loc[neighborhood, 'Entire home/apt Ratio']
    ratio_private = top_5_neighborhoods.loc[neighborhood, 'Private room Ratio']
    plt.text(ratio_entire, i, f'{ratio_entire}%', va='center', color='black')
    plt.text(ratio_private, i, f'{ratio_private}%', va='center', color='black')

plt.title("Top 5 Neighborhoods with Highest Ratios")
plt.xlabel("Ratio (%)")
plt.ylabel("Neighborhood")
plt.legend()

plt.show()
```



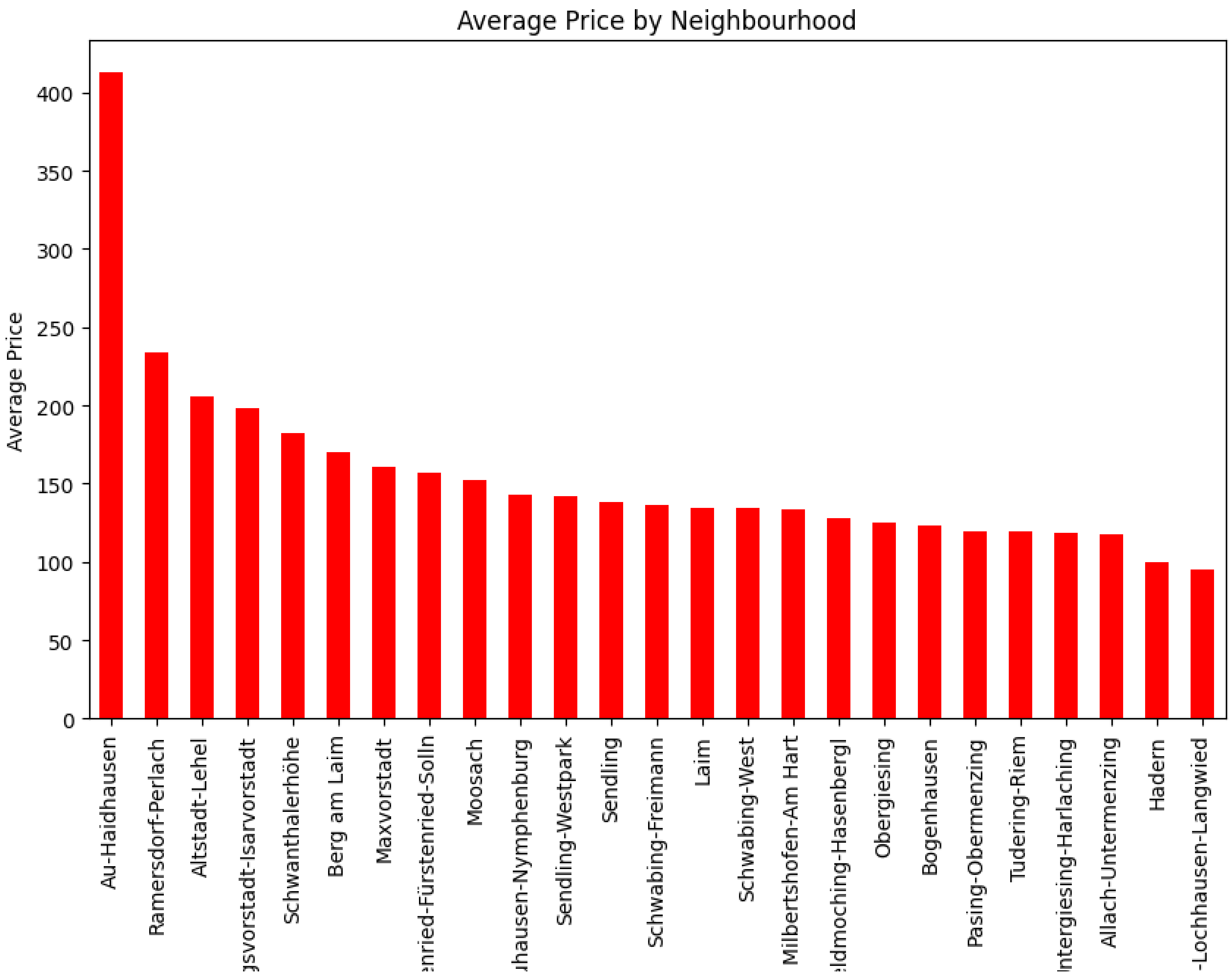
The unique characteristics of the neighborhoods, such as their location, amenities, atmosphere, or target audience, might make them more attractive for travelers or tenants seeking private room accommodations. For example, neighborhoods near universities or popular tourist destinations often have a higher demand for private rooms.

Objective 2:

```
selected_columns = ['neighbourhood', 'price']
selected_data = data[selected_columns]
average_price_by_neighbourhood = round(selected_data.groupby('neighbourhood')['price'].mean().sort_values(ascending=False),2)
average_price_by_neighbourhood
```

```
neighbourhood
Au-Haidhausen                                          412.64
Ramersdorf-Perlach                                     233.79
Altstadt-Lehel                                         206.05
Ludwigsvorstadt-Isarvorstadt                           198.49
Schwanthalerhöhe                                       182.61
Berg am Laim                                           170.39
Maxvorstadt                                            160.45
Thalkirchen-Obersendling-Forstenried-Fürstenried-Solln 156.63
Moosach                                                152.23
Neuhausen-Nymphenburg                                  142.49
Sendling-Westpark                                      142.08
Sendling                                               138.30
Schwabing-Freimann                                     136.52
Laim                                                   134.71
Schwabing-West                                         134.69
Milbertshofen-Am Hart                                  133.69
Feldmoching-Hasenbergl                                 127.97
Obergiesing                                            125.38
Bogenhausen                                            123.26
Pasing-Obermenzing                                     119.39
Tudering-Riem                                          119.02
Untergiesing-Harlaching                                118.72
Allach-Untermenzing                                    117.96
Hadern                                                 100.17
Aubing-Lochhausen-Langwied                              95.43
Name: price, dtype: float64
```

```
import matplotlib.pyplot as plt

neighbourhood_prices = new_data.groupby('neighbourhood')['price'].mean().sort_values(ascending=False)

plt.figure(figsize=(10, 6))
neighbourhood_prices.plot(kind='bar', color='red')
plt.title('Average Price by Neighbourhood')
plt.xlabel('Neighbourhood')
plt.ylabel('Average Price')
plt.xticks(rotation=90)
plt.show()
```



Above is the average property price per neighbourhood in Munich. We can see that there is a big price difference, where the average price in the most expensive neighbourhood is 412.64 and in the cheapest neighbourhood is 95.43.

```
highest_prices = round(new_data.groupby('neighbourhood')['price'].mean().nlargest(2),2)

lowest_prices = round(new_data.groupby('neighbourhood')['price'].mean().nsmallest(2),2)

print("Neighbourhoods with the Highest Prices:")
print(highest_prices)

print("\nNeighbourhoods with the Lowest Prices:")
print(lowest_prices)
```

```
Neighbourhoods with the Highest Prices:
neighbourhood
Au-Haidhausen         412.64
Ramersdorf-Perlach    233.79
Name: price, dtype: float64

Neighbourhoods with the Lowest Prices:
neighbourhood
Aubing-Lochhausen-Langwied     95.43
Hadern                        100.17
Name: price, dtype: float64
```

Objective 3:

```
!pip install folium
```

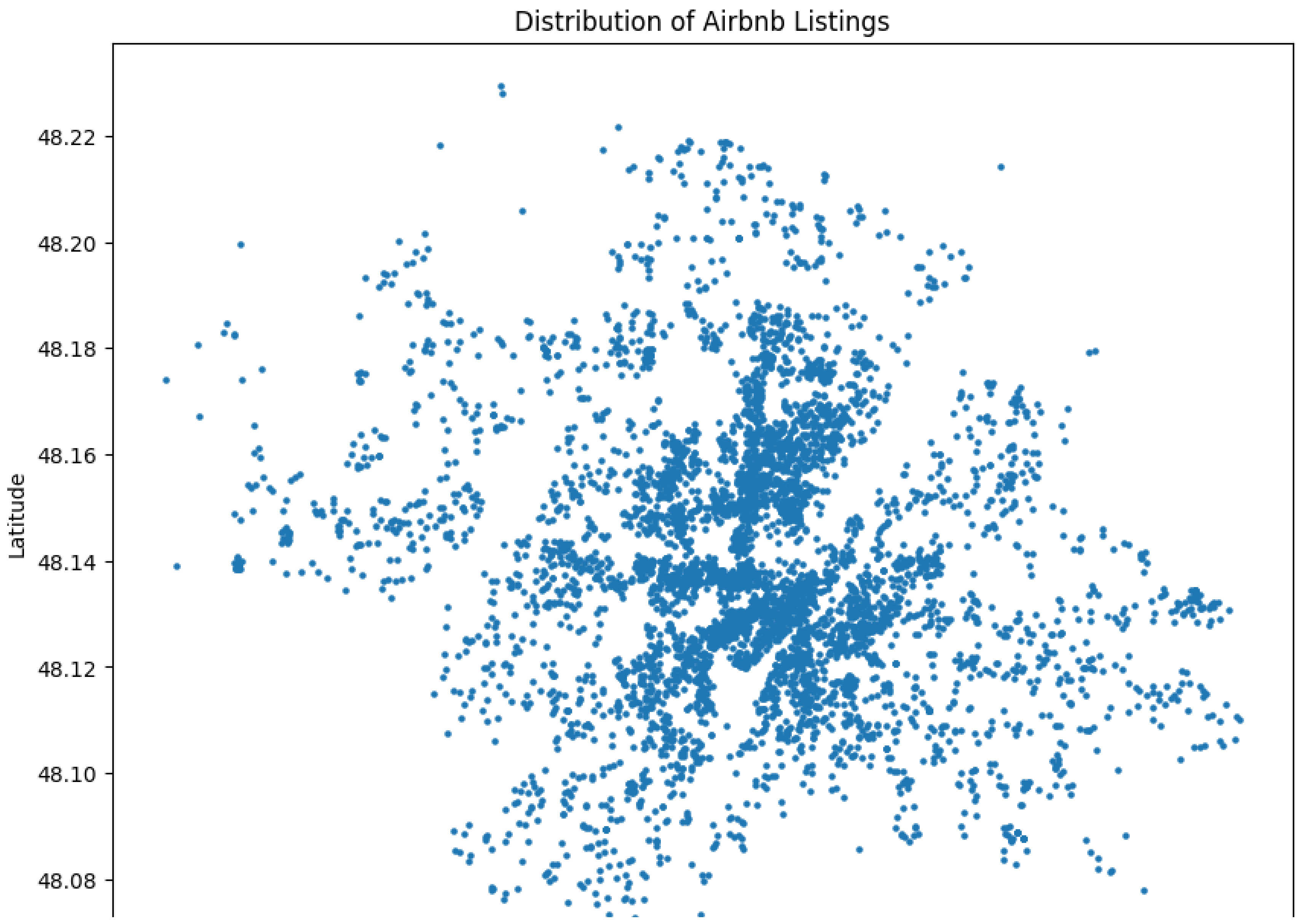
```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: folium in /usr/local/lib/python3.10/dist-packages (0.14.0)
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from folium) (0.6.0)
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from folium) (1.22.4)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from folium) (2.27.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2>=2.9->folium) (2.1.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.4)
```

```
!pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: geopandas in /usr/local/lib/python3.10/dist-packages (0.13.0)
Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.9.4.post1)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas) (23.1)
Requirement already satisfied: pandas>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.5.3)
Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (3.5.0)
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (2.0.1)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (23.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (2022.12.7)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (8.1.3)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (0.7.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopandas) (1.22.4)
```

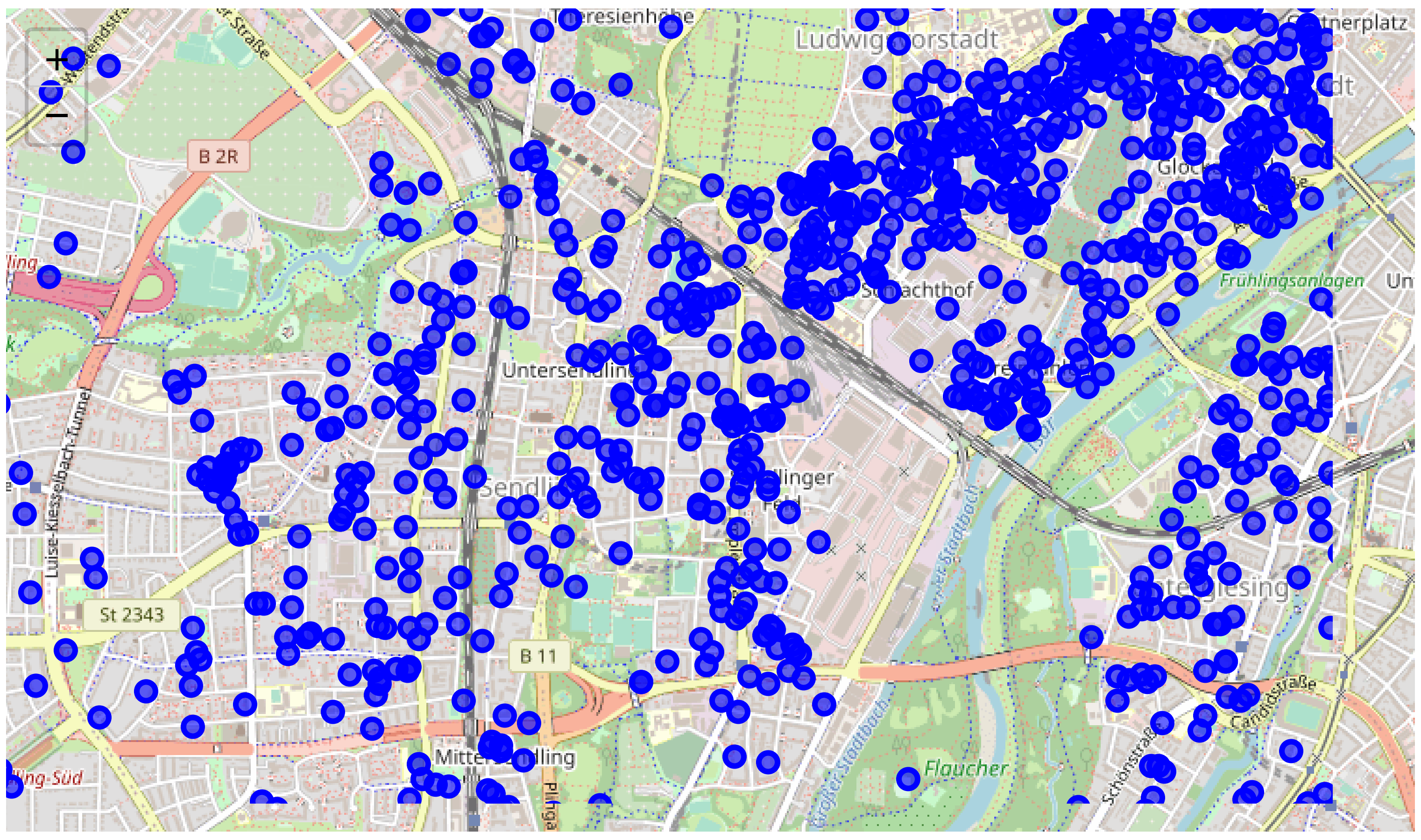
```
import geopandas as gpd
```

```
selected_columns = ['latitude', 'longitude']
selected_data = data[selected_columns]

plt.figure(figsize=(10, 8))
plt.scatter(selected_data['longitude'], selected_data['latitude'], s=5, alpha=1.0)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Distribution of Airbnb Listings')
plt.show()
```



Sparse areas or gaps on the scatter plot indicate regions in Munich with fewer Airbnb listings. These areas might be underrepresented or less popular for rentals, which could provide opportunities for market analysis or potential expansion of Airbnb services.

```
import folium

selected_columns = ['latitude', 'longitude']
selected_data = data[selected_columns]

m = folium.Map(selected_data=['latitude', 'longitude'], zoom_start=12)

for index, row in data.iterrows():
    folium.CircleMarker(location=[row['latitude'], row['longitude']],
                        radius=5,
                        color='blue',
                        fill=True,
                        fill_color='blue',
                        fill_opacity=0.6,
                        popup=row['neighbourhood']).add_to(m)

m
```



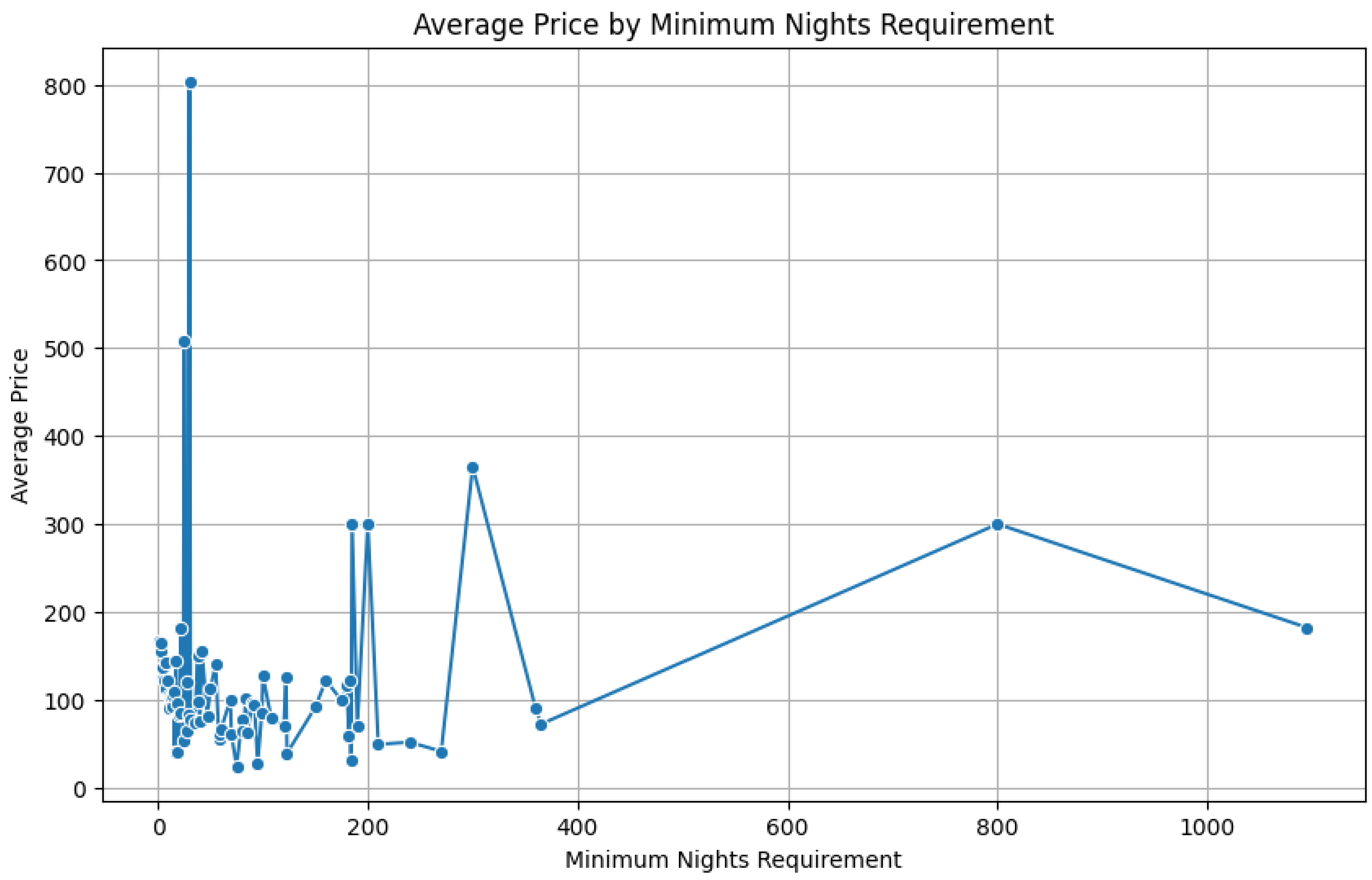
Above graph show more specific for the location with Airbnb activity. The graph provides a visual representation of the geographic distribution of Airbnb in Munich listings based on latitude and longitude coordinates. Each marker on the map represents a specific location. The density of markers indicates the concentration of Airbnb listings in different areas. We can observe clusters or hotspots where multiple markers are closely grouped together, indicating popular or densely populated areas for rentals. Sparse areas or gaps on the map with fewer markers indicate regions with lower Airbnb activity. These areas might be less popular for rentals, indicating potential gaps or opportunities for market analysis or expansion of Airbnb services. By examining the map, we can identify the proximity of Airbnb listings to specific points of interest such as tourist attractions, parks, transportation hubs, or commercial centers. This information can be useful for travelers or property owners in understanding the accessibility and desirability of different locations. The interactive nature of the map allows users to zoom in and out, as well as pan across different areas for a more detailed exploration of the distribution and patterns.

Objective 4:

```
selected_columns = ['minimum_nights', 'price']
selected_data = data[selected_columns]
round(selected_data.groupby('minimum_nights')['price'].mean().sort_values(ascending=False),2)
```

```
minimum_nights
30     803.81
25     508.47
300    365.33
800    300.00
200    300.00
        ...
16      40.00
123     39.00
184     30.00
95      27.00
75      24.00
Name: price, Length: 74, dtype: float64
```

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

selected_columns = ['minimum_nights', 'price']
selected_data = data[selected_columns]

average_price_by_minimum_nights = selected_data.groupby('minimum_nights')['price'].mean().reset_index()

plt.figure(figsize=(10, 6))
sns.lineplot(data=average_price_by_minimum_nights, x='minimum_nights', y='price', marker='o')
plt.title('Average Price by Minimum Nights Requirement')
plt.xlabel('Minimum Nights Requirement')
plt.ylabel('Average Price')
plt.grid(True)
plt.show()
```



From this we can determine the optimal minimum nights for maximizing rental income or occupancy rates, we can look for points on the graph that correspond to higher average prices. These points indicate the minimum nights requirement that yields higher rental prices, which can contribute to maximizing rental income or occupancy rates.

There is no trend that we can see between the minimum nights requirement and rental prices. It is because of rental prices are typically influenced by multiple factors, such as property size, location, amenities, and seasonal demand. The minimum nights requirement alone may not be the primary determinant of rental prices. We should consider analyzing the combined effect of multiple variables to gain a comprehensive understanding of the factors influencing rental prices.