Airbnb Business Analysis Report: Is there any noticeable difference in price and the neighbourhood of an Airbnb listing?

Introduction

This Airbnb business analysis focused on establishing if there is a relationship between price difference and the neighbourhood of an Airbnb listing (Dudás, 2017). By leveraging data-driven insights into price variations among neighbourhoods, Airbnb can gain a deeper understanding of market demand, optimise pricing strategies, differentiate its offerings, and ultimately enhance the overall success of the business (Luo et al., 2019). Analysed with the use of the 2019 Airbnb dataset (Appendix 1) (Kaggle, 2019).

Methodology

Data Preprocessing

Handle missing values, outliers, and inconsistencies in the dataset. Clean and normalise the data for accurate analysis. Perform data transformation and feature engineering if necessary.

Exploratory Data Analysis (EDA)

Conduct descriptive statistics to understand the central tendencies and distributions of the price data across different neighbourhoods.

Visualise the price differences among neighbourhood groups (boroughs)

Using box plots, histograms, or bar charts. Explore potential correlations between price and other relevant features such as property type, room type, and availability.

K-means Clustering

Use the K-means clustering algorithm to group the neighbourhoods based on the property listing prices.

Analysis and Interpretation

Analyse the clusters to identify distinct groups of neighbourhoods with similar pricing patterns.

Evaluate the characteristics of each cluster to understand the price differences and similarities among neighbourhood groups.

Exploratory Data Analysis

We underwent EDA to assist us with the interpretation of the data. EDA is defined as "performing initial investigations on data to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations" (Patil, 2022). This allows us to grasp an understanding of data before delving deeper into the analysis of the dataset. This permits more ease when discovering the answer of if there is a relationship between price and neighbourhood.

We first determined the structure, size and the type of data. Once there was an understanding of the structure of the data, null data and outliers were then removed from the dataset, this was beneficial as there was now "a lesser chance of errors and biases when analysing the data" (Kwak & Kim, 2017). Histograms seen in Figures 1 & 2 not only displayed the distribution of the data but by analysing the shape of the histograms, outliers could also be detected and removed. Since the aim of this report is to find out if the neighbourhood group of an Airbnb has an impact on the listed price, we determined that outliers applied to Airbnbs listed at more than $1000.

Figure 1: Histogram of AirBnB prices in New York

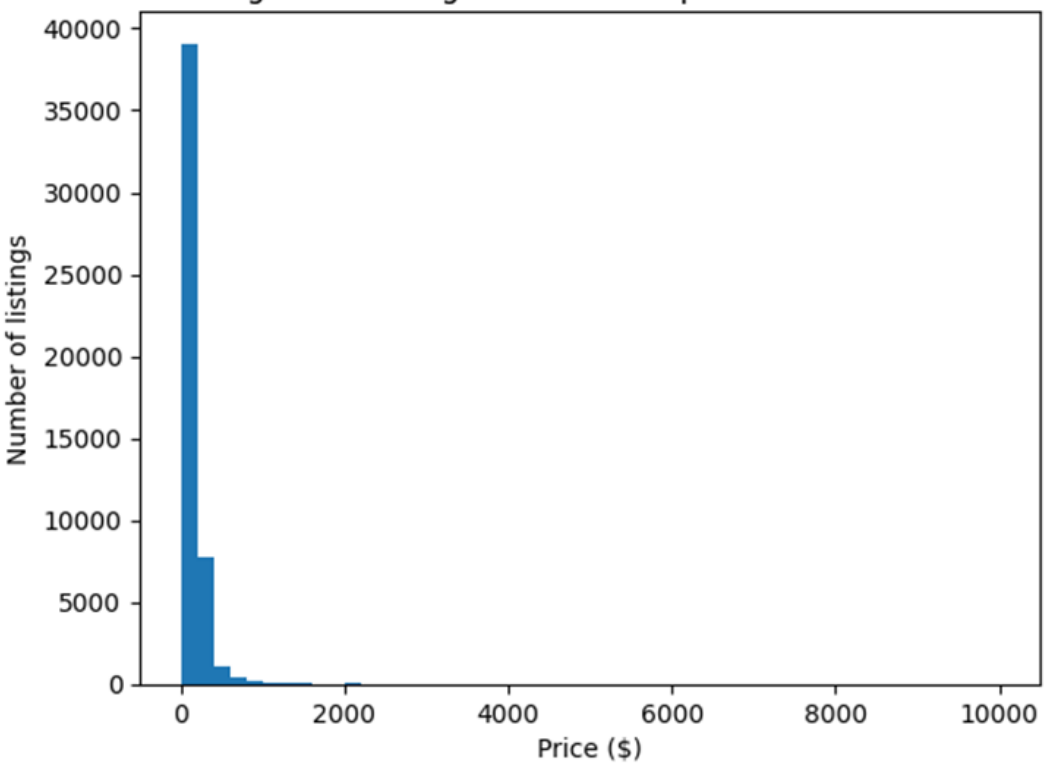
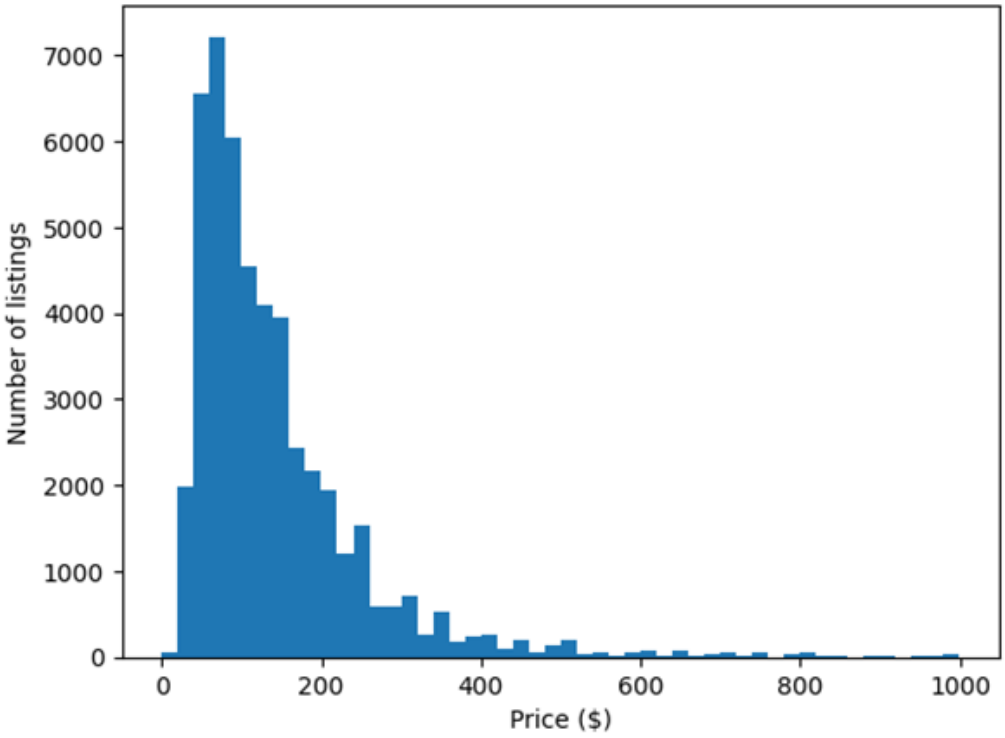
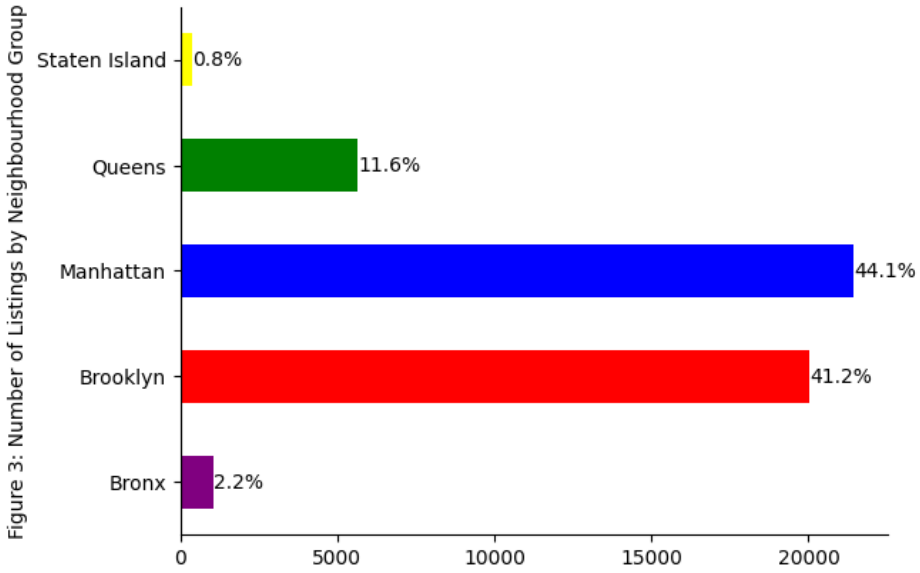


Figure 2: Histogram of AirBnB prices in New York after outliers have been removed



Data Visualisation

New York City has plenty of places to stay on Airbnb and in Figure 3 we see the number of listings in each neighbourhood showing that most of the listings are in Manhattan and Brooklyn making up 85.3%. Showing to travellers that they are likely to stay somewhere in either Brooklyn or Manhattan whilst Queens, Bronx and Staten Island are less popular for listings.



Going onto Figure 4 we are able to see the distribution of price in each borough with the boroughs with a higher number of listings having a higher median price compared to those with a lower number of listings having a lower median price showing a noticeable difference in price between the 5 boroughs based on median price as Manhattan costs $149 per night and for a slightly less expensive stay Brooklyn is another option at $90. Whilst cheaper options in the other 3 boroughs range from $75 to $65 median price per night (Kaggle, 2019).

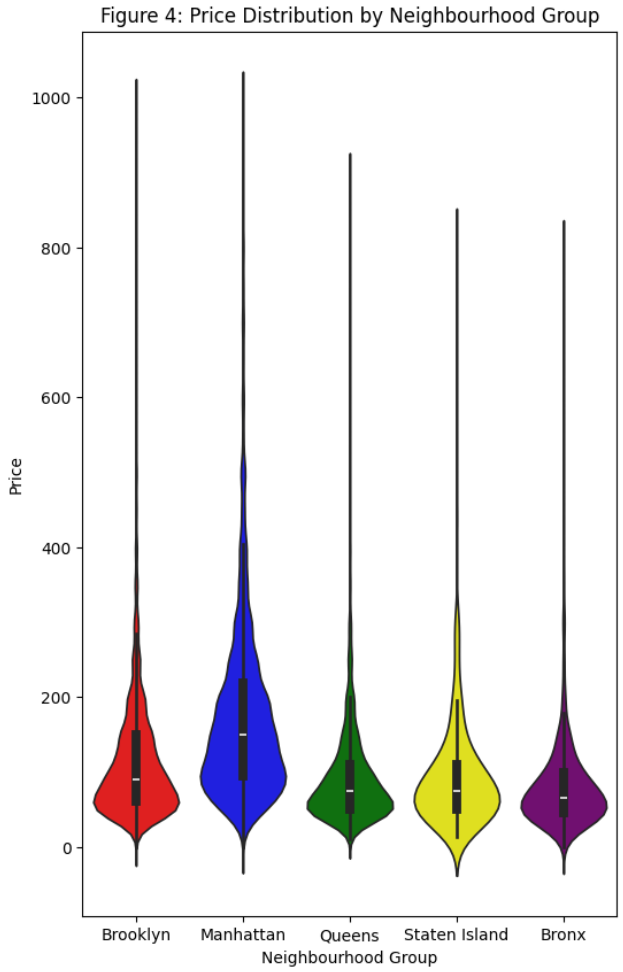
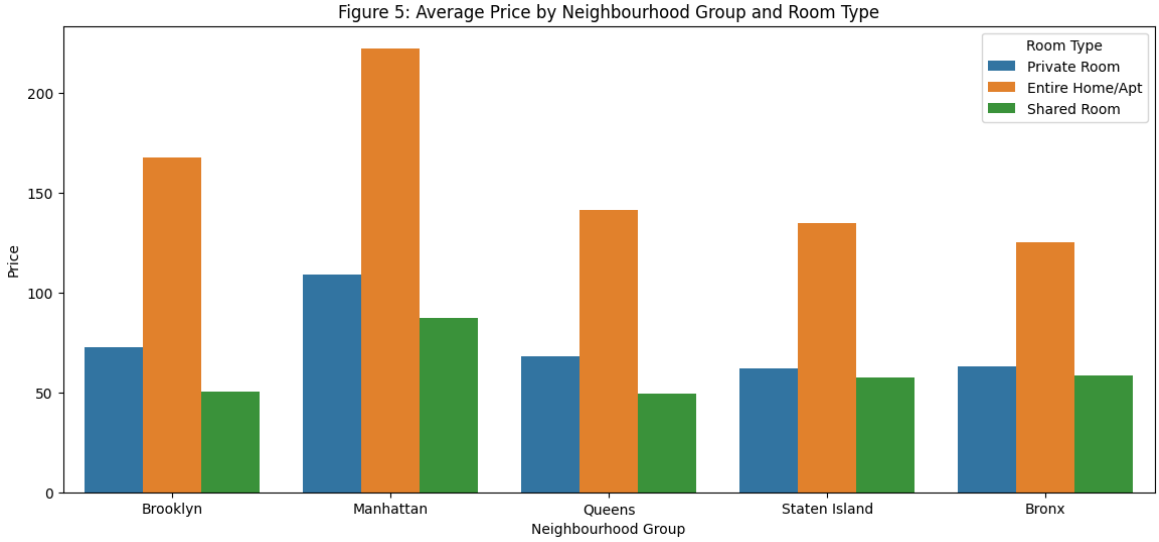


Figure 5 we further breakdown the average price by neighbourhood group and the types of rooms on offer which are either a private room, entire home/apartment or a shared room. Showing that it is cheaper to stay in Queens, Staten Island and the Bronx. However, as we know from Figure 1 there is far lower availability and for someone visiting New York City there are less tourist attractions in this area so a visitor may choose Manhattan or Brooklyn due to there being more tourist attractions and being closer to the centre of the city (Rogers, 2016).

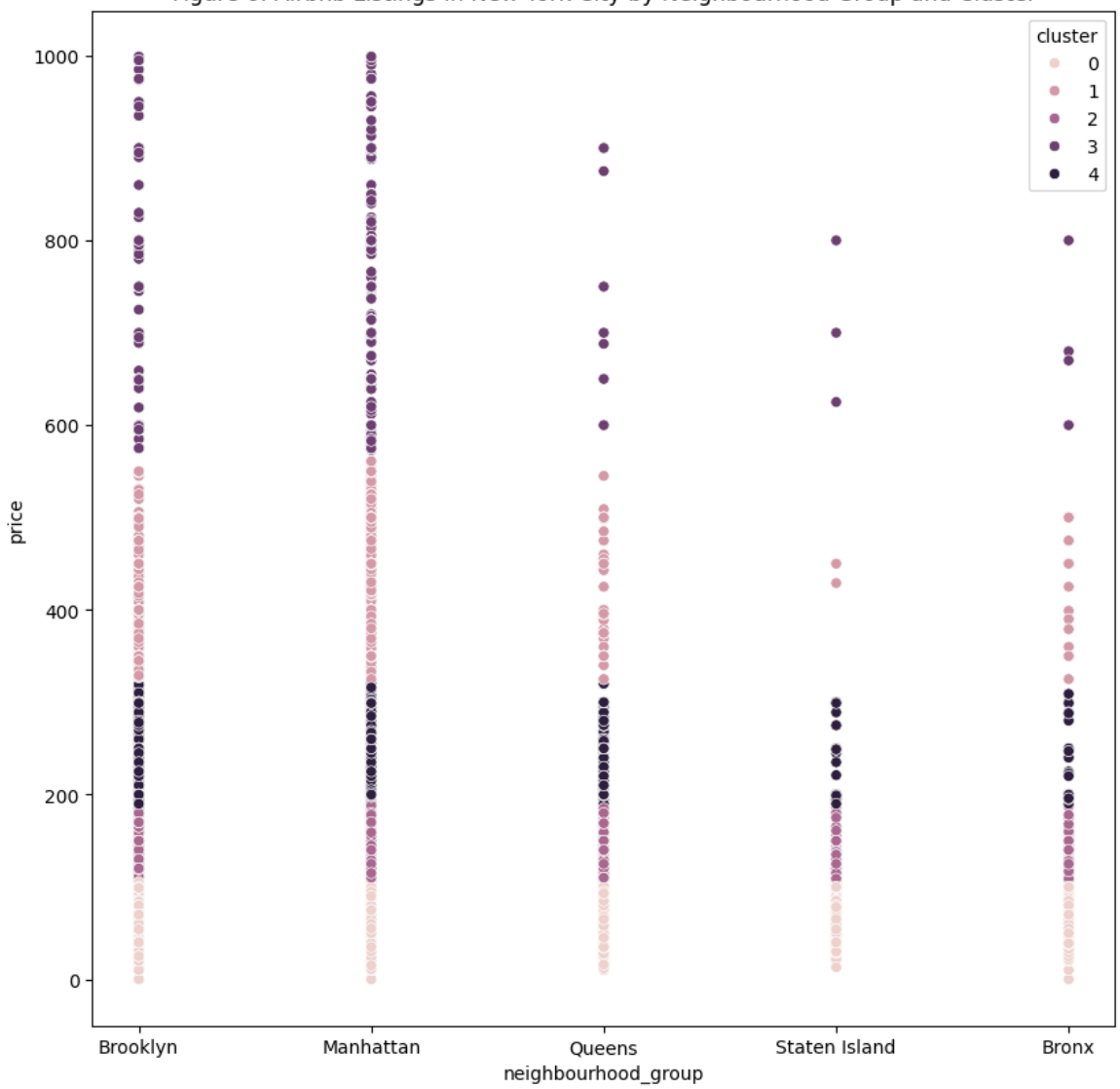
Figure 5: Average Price by Neighbourhood Group and Room Type



Unsupervised Machine Learning

As gathered from the data analysis and visualisation when measuring price against neighbourhood groups, there is a clear relationship where more expensive listings are located in Manhattan and Brooklyn. To further ensure this, the unsupervised machine learning algorithm of K-means clustering was applied. “K-means relies on the average (mean) of the cluster to measure the similarity of the group, to group similar values in clusters” (Thinsungnoen et al., 2015). Despite the data being linear between price and neighbourhood group, the K-means clustering process (Figure.6) gives more of an in-depth visualisation of the relationship between the two variables. When analysing the results for cluster 2, it is further cemented that Manhattan and Brooklyn clearly possess listings with a higher average price, this is also shown in cluster number 4 where this cluster possesses listings that are in a group one level below. Justification for these more expensive listings in the aforementioned boroughs may be tied to these districts having “the highest cost of living in the country, with the perception of higher costs deriving from wealthy individuals driving inflation” (Rutkoff, 2011).

Figure 6: Airbnb Listings in New York City by Neighbourhood Group and Cluster



Findings and Insights

The analysis revealed significant variations in listing prices across the boroughs. Manhattan emerged as the borough with the highest average listing prices, indicating a premium pricing trend in this area (Glaeser, 2015). In contrast, Bronx, Brooklyn, Queens, and Staten Island displayed relatively even distributions of listing prices, suggesting a more uniform pricing landscape across these boroughs. What's more, the statistical test resulted in a p-value of 0.0000, which is less than the conventional significance level of 0.05. This statistically significant

result confirms that there is indeed a substantial difference in listing prices among the boroughs in New York City (Andrade, 2019)

Conclusion & Recommendations

The high prices in Manhattan create a chance to tap into the wealthy market and boost earnings from luxury listings. Meanwhile, the consistent pricing in other boroughs indicates an opportunity for competitive pricing approaches, appealing to a wider range of customers.

Airbnb can improve business and better serve its different customers by customising prices based on the unique characteristics of each borough (Ahmad et al., 2021).

In order to enhance targeted marketing and growth, further research could look at specific factors driving high prices in Manhattan (Cheng and Jin, 2019). It's also critical to investigate methods for optimising pricing strategies in this profitable business (Sirgy,2014).

References

Ahmad Bakri, A., Rosman, S.H. and Ismail, S., 2021. Success factors of marketing strategy in real estate business. ASEAN Entrepreneurship Journal (AEJ), 7(1), pp.20-26.

Andrade, C., 2019. The P value and statistical significance: misunderstandings, explanations, challenges, and alternatives. Indian journal of psychological medicine, 41(3), pp.210-215.

Cheng, M. and Jin, X., 2019. What do Airbnb users care about? An analysis of online review comments. International Journal of Hospitality Management, 76, pp.58-70.

Dudás, G., Vida, G., Kovalcsik, T. and Boros, L., 2017. A socio-economic analysis of Airbnb in New York City. Regional Statistics, 7(1), pp.135-151.

Glaeser, E.L., Gyourko, J. and Saks, R., 2005. Why is Manhattan so expensive? Regulation and the rise in housing prices. The Journal of Law and Economics, 48(2), pp.331-369.

Kaggle. (2019). Airbnb dataset. Available from: https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data . [Accessed on 08 March 2024]

Kwak, S.K. and Kim, J.H. (2017) 'Statistical Data Preparation: Management of missing values and outliers', *Korean Journal of Anesthesiology*, 70(4). doi:10.4097/kjae.2017.70.4.407.

Luo, Y., Zhou, X. and Zhou, Y., 2019. Predicting airbnb listing price across different cities.

Patil, P. (2022) *What is exploratory data analysis?*, *Medium*. Available at: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15#:~:text=EDA%20is%20all%20about%20making,getting%20them%20dirty%20with%20it. (Accessed: 18 April 2024).

Rogers, D. (2016) *Where to stay in New York: Hotels by district*, *The Telegraph*. Available at: https://www.telegraph.co.uk/travel/destinations/north-america/united-states/new-york/articles/where-to-stay-in-new-york-hotels-by-district-nyc/ (Accessed: 19 April 2024).

Rutkoff, A. (2011) *America's most expensive places to live - Manhattan, Brooklyn, queens - WSJ*. Available at: https://www.wsj.com/articles/BL-METROB-14386 (Accessed: 20 April 2024).

Sirgy, M.J., 2014. Real estate marketing: Strategy, personal selling, negotiation, management, and ethics. Routledge.

Thinsungnoen, T. *et al.* (2015) 'The clustering validity with silhouette and sum of squared errors', *The Proceedings of the 2nd International Conference on Industrial Application Engineering 2015* [Preprint]. doi:10.12792/iciae2015.012.

Wang, C.R. and Jeong, M., 2018. What makes you choose Airbnb again? An examination of users' perceptions toward the website and their stay. International Journal of Hospitality Management, 74, pp.162-170.

Appendices

Appendix 1: Dataset

The dataset, named AB_NYC_2019.csv encompasses information regarding listing activity and metrics in NYC, NY for the year 2019 related Airbnb listings. This dataset consists of 16 columns and 48,895 rows which contain essential details about hosts, geographical availability, and pertinent metrics, enabling the generation of predictions and conclusions related to Airbnb operations

Appendix 2: Tools and algorithms employed:

- Google Colab
- python
- numpy
- pandas
- matplotlib
- seaborn
- scipy
- sklearn
- missingno
- pylab

Appendix 3: Code

Loading Data and Packages

```
import numpy as np

import pylab as pl

import matplotlib.pyplot as plt

import seaborn as sns

import os

import numpy as np # linear algebra

import pandas as pd # data processing

import matplotlib.pyplot as plt

import seaborn as sns

import scipy.stats as st

from sklearn import ensemble, tree, linear_model

import missingno as msno

from scipy.stats import f_oneway

from matplotlib import pyplot as plt



#importing the required visulaisation tools
```

df = pd.read_csv('AB_NYC_2019.csv')

#reading the dataset

Viewing Data and Removing Nulls

df.shape

(48895, 16)

df.head(5)

#checking the data has been imported by looking at the first 5 rows

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_host_listings_count	availability_365
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	2018-10-19	0.21	6	365
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	2019-05-21	0.38	2	355
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	NaN	NaN	1	365
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	2019-07-05	4.64	1	194
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	2018-11-19	0.10	1	0

df.info()

```
<class 'pandas.core.frame.DataFrame'>
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
 15  availability_365                48895 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

df.isnull().sum()

checking if data is null in each field

```
id                                0
name                             16
host_id                           0
host_name                        21
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                   10052
reviews_per_month             10052
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

df.loc[df.number_of_reviews==0, 'reviews_per_month'] = 0

```
df.loc[df.number_of_reviews==0, 'last_review'] = 0

#changing the nulls to 0

df = df[pd.notnull(df['name'])]

df = df[pd.notnull(df['host_name'])]

#filtering out null values
```

Observaton made about the missing data

name is missing 16 entries

host_name has 21 entries missing

last_review column missing 10052

reviews_per_month 10052

```
df.isnull().sum()
```

re-checking if the data is null

```
id                                0
name                              0
host_id                           0
host_name                         0
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

df.describe()

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	4.885800e+04	4.885800e+04	48858.000000	48858.000000	48858.000000	48858.000000	48858.000000	48858.000000	48858.000000	48858.000000
mean	1.902335e+07	6.763169e+07	40.728941	-73.952170	152.740309	7.012444	23.273098	1.091124	7.148369	112.801425
std	1.098289e+07	7.862389e+07	0.054528	0.046159	240.232386	20.019757	44.549898	1.597270	32.964600	131.610962
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
25%	9.475980e+06	7.818669e+06	40.690090	-73.983070	69.000000	1.000000	1.000000	0.040000	1.000000	0.000000
50%	1.969114e+07	3.079133e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.370000	1.000000	45.000000
75%	2.915765e+07	1.074344e+08	40.763107	-73.936280	175.000000	5.000000	24.000000	1.580000	2.000000	227.000000
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

function that calculates summary statistics for price, this helps determine outliers

```
def price_summary_stats(df):
  price_stats = df.groupby('neighbourhood_group')['price'].describe()
  return price_stats
```

price_summary_stats(df)

neighbourhood_group	count	mean	std	min	25%	50%	75%	max
Bronx	1089.0	87.469238	106.798933	0.0	45.0	65.0	99.0	2500.0
Brooklyn	20089.0	124.410523	186.936694	0.0	60.0	90.0	150.0	10000.0
Manhattan	21643.0	196.897473	291.489822	0.0	95.0	150.0	220.0	10000.0
Queens	5664.0	99.536017	167.128794	10.0	50.0	75.0	110.0	10000.0
Staten Island	373.0	114.812332	277.620403	13.0	50.0	75.0	110.0	5000.0

np.mean(df.price)

152.74030864955586

import numpy as np

```
def price_interquartile_range(df):
  price_IQR = df.groupby('neighbourhood_group')['price'].agg(lambda x: np.percentile(x, 75) - np.percentile(x, 25))
  return price_IQR
```

price_interquartile_range(df)

```
neighbourhood_group
Bronx             54.0
Brooklyn          90.0
Manhattan        125.0
Queens            60.0
Staten Island     60.0
Name: price, dtype: float64
```

Observation:

Manhattan has inter-quartile range of 125 indicating the widest spread of price for property listing.

IQR 60 for Queens and Staten Island suggests prices are clustered tightly, sharing similar listings.

Using median compare the price

median_price = df.groupby('neighbourhood_group')['price'].median()

print(median_price)

```
neighbourhood_group
Bronx             65.0
Brooklyn          90.0
Manhattan        150.0
Queens            75.0
Staten Island     75.0
Name: price, dtype: float64
```

A higher median of 150 indicates that Manhattan listings are priced on average higher than other NY boroughs.

Queens and Staten Island have the lowest median of 75, this suggests that their listings are priced lowest on average.

To determine if there is a significant difference in prices among New York City boroughs, we perform an ANOVA (Analysis of Variance) test

Our null hypothesis (H0) is that there is no significant difference in prices among the boroughs.

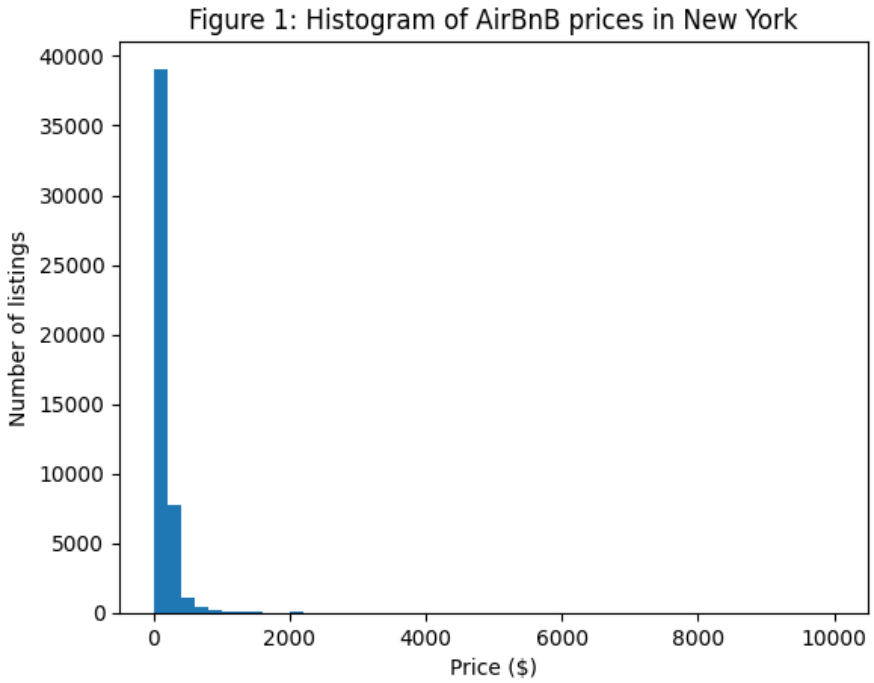
The alternative hypothesis (Ha) is that there is a significant difference in prices among the boroughs

```
# Drop unnecessary columns

nyc = df.drop(["latitude", "longitude", "last_review", "host_name", "id", "host_id", "name"], axis=1)

# Create dummy variables for borough and room type

dummies_neighbourhood = pd.get_dummies(nyc["neighbourhood_group"])

dummies_room = pd.get_dummies(nyc["room_type"])

nyc = pd.concat([nyc, dummies_neighbourhood, dummies_room], axis=1)

# Drop rows with missing values

nyc = nyc.dropna()

# Perform ANOVA test

boroughs = ["Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island"]

f_statistic, p_value = f_oneway(*[nyc[nyc[borough] == 1]["price"] for borough in boroughs])

print(f"p-value = {p_value:.4f}")
```

p-value = 0.0000

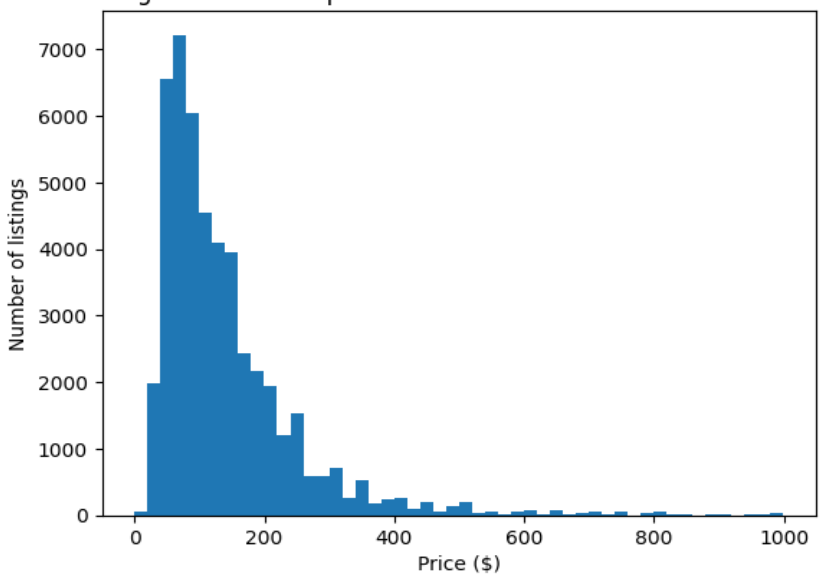
Since p-value(0.0000) is less than 0.05, We reject null hypothesis and say that there is a significant difference in prices among boroughs

```
plt.hist(df.price, bins=50)

plt.title('Figure 1: Histogram of AirBnB prices in New York')

plt.xlabel('Price ($)')  # Label for the x-axis

plt.ylabel('Number of listings')  # Label for the y-axis

#histogram of price to check the distribution of the data
```



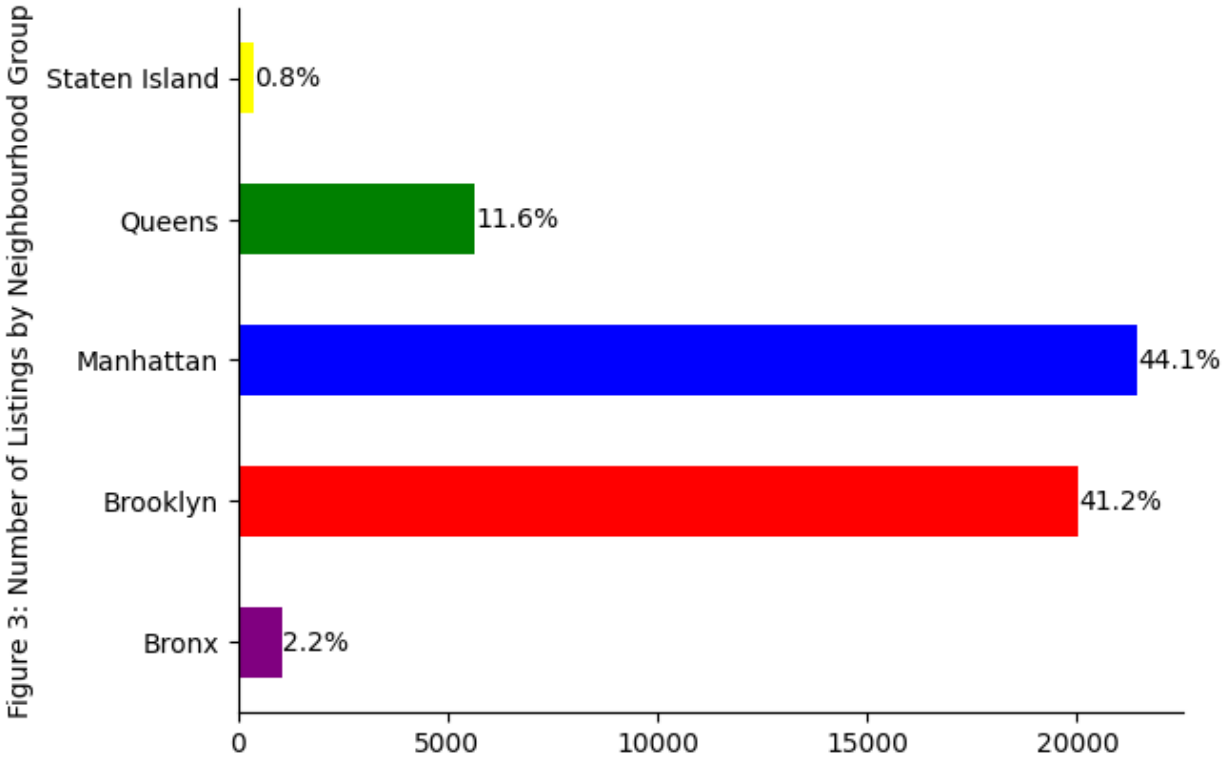
```
len(df[df.price > 1000])

#counting the entries that are higher than 1000

239

df = df[df.price < 1000]

#removing outliers over 1000



plt.hist(df.price, bins=50)

plt.title('Figure 2: Histogram of AirBnB prices in New York after outliers have been removed')

plt.xlabel('Price ($)')  # Label for the x-axis

plt.ylabel('Number of listings')  # Label for the y-axis

#checking the distribution again
```

Figure 2: Histogram of AirBnB prices in New York after outliers have been removed



```
# Calculate counts per group

group_counts = df.groupby('neighbourhood_group').size()

# Calculate total for percentage computation

total = group_counts.sum()

# Define colors

colors2 = ['purple', 'red', 'blue', 'green', 'yellow']

# Create bar plot

ax = group_counts.plot(kind='barh', color=colors2)

# Remove unnecessary spines

plt.gca().spines['top'].set_visible(False)

plt.gca().spines['right'].set_visible(False)

# Set the new y-axis label

plt.ylabel('Figure 3: Number of Listings by Neighbourhood Group')

# Annotate percentages on the bars

for i in ax.patches:

    ax.text(i.get_width()+0.3, i.get_y() + i.get_height()/2, '{:.1%}'.format(i.get_width()/total),
va='center')
```

plt.show()



We can see that there is a vast majority of airbnbs in the tourist locations of Manhattan and Brooklyn

plt.figure(figsize=(6, 10))

Define color palette

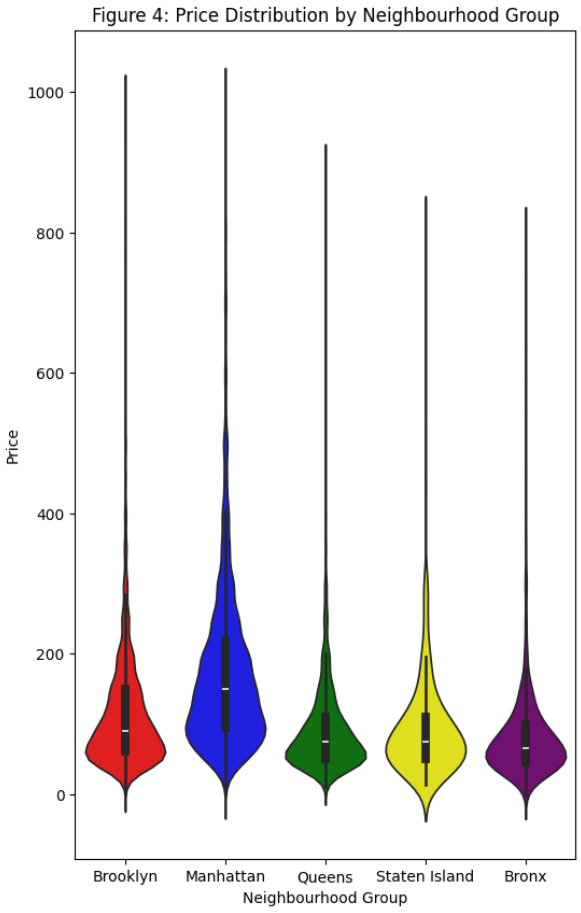
colors = ['red', 'blue', 'green', 'yellow', 'purple']

Create a violin plot

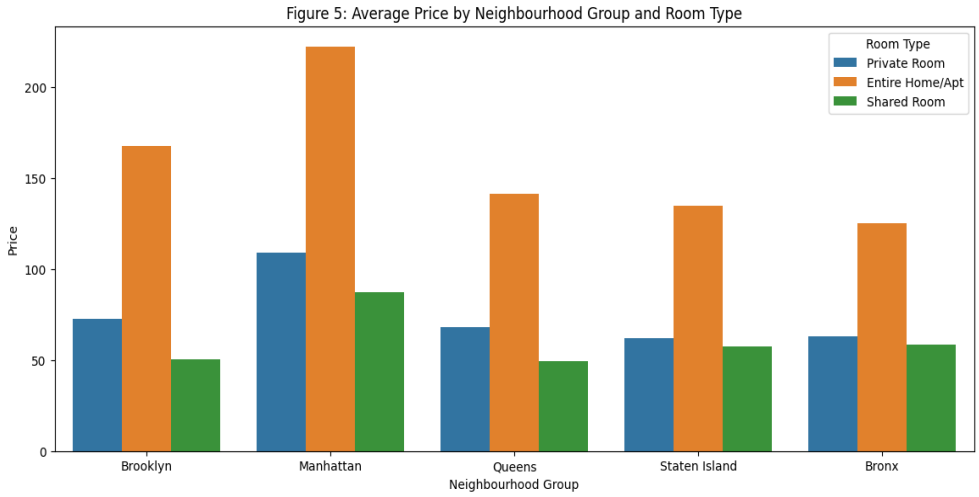
ax = sns.violinplot(x="neighbourhood_group", y="price", data=df, palette=colors)

Set title

```
ax.set_title('Figure 4: Price Distribution by Neighbourhood Group')

# Set x and y labels

ax.set_xlabel('Neighbourhood Group')

ax.set_ylabel('Price')

plt.show()
```



```
plt.figure(figsize=(14, 6))

# Create a bar plot

ax = sns.barplot(x="neighbourhood_group", y="price", hue="room_type", data=df, ci=None)

# Set title

plt.title('Figure 5: Average Price by Neighbourhood Group and Room Type')

# Set x and y labels

ax.set_xlabel('Neighbourhood Group')

ax.set_ylabel('Price')

# Rename the legend title

handles, labels = ax.get_legend_handles_labels()

ax.legend(handles, ["Private Room", "Entire Home/Apt", "Shared Room"], title='Room Type')

plt.show()
```



A clearer plot showing the average price especially for entire homes is much higher for Manhattan than it is for other boroughs with Brooklyn coming second.

```
import pandas as pd

df['price_quantiles'] = pd.qcut(df['price'], 4)

# Check for missing values and address them

print(df.isnull().sum())

df.dropna(subset=['price', 'neighbourhood_group'], inplace=True)  # Example of dropping rows with missing values

# Encode 'Neighbourhood_Group' as a categorical variable

df['Neighbourhood_Group_Code'] = pd.Categorical(df['neighbourhood_group']).codes

# Verify changes

print(df[['neighbourhood_group', 'Neighbourhood_Group_Code']].drop_duplicates())
```

```
id                                0
name                             16
host_id                           0
host_name                        21
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                   10052
reviews_per_month             10052
calculated_host_listings_count    0
availability_365                  0
dtype: int64
    neighbourhood_group  Neighbourhood_Group_Code
0              Brooklyn                         1
1             Manhattan                         2
46               Queens                         3
169       Staten Island                         4
171               Bronx                         0
```

```
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Create a KMeans model with 5 clusters

kmeans = KMeans(n_clusters=5)

# Fit the model to the price data

kmeans.fit(df['price'].values.reshape(-1, 1))

# Get the cluster labels for each data point

labels = kmeans.labels_

# Add the cluster labels to the DataFrame

df['cluster'] = labels

# Create a new DataFrame with only the neighborhood_group, latitude, longitude, and cluster
columns

df_map = df[['neighbourhood_group', 'price', 'cluster']]

# Create a figure and axes

fig, ax = plt.subplots(figsize=(10, 10))

# Plot the points on the map using Seaborn, grouped by cluster

sns.scatterplot(x='neighbourhood_group', y='price', data=df_map, hue='cluster', ax=ax)

# Set the title and show the plot
```

```
plt.title('Figure 6: Airbnb Listings in New York City by Neighbourhood Group and Cluster')

plt.show()
```