# Code

February 2, 2024

What factors influence the pricing of Airbnb listings in New York City?

Introduction

With the rapid rise of Airbnb, it is important for stakeholders to understand the pricing of listings. Such research has numerous applications; in improving the efficiency of Airbnb and short term rental markets, for policy makers to better understand and develop regulations around the market, and for other participants such as tourist services and investors to better understand the platform.

New York City, a major financial center and economic powerhouse, with a bustling tourism sector, and unfolding housing crisis, presents an interesting case study for our question.

Analyzing the market however is no easy task, and will require us to examine endogenous factors (such as airbnb supply) as welll as exogenous ones such as socioeconomic characteristics across New York's bouroughs. One big first step in analyzing the pricing is by examining Airbnb's own dataset. We'll naturally observe price to be our dependent variable, and look at it's relation with the variables 'reviews_per_month', 'room_type', 'neighbourhood_group', 'minimum_nights', and 'availability_365'.

'reviews_per_month' is a natural choice to be looking at its correlation with price, we can speculate more reviews correlated with higher price. We can also speculate that lower 'availability_365' correlates with a higher price because of supply and demand, however lower 'availability_365' may also be indicative of a listing's lower time on the market. We can speculate larger 'room_type's might be more in demand to capitalize on tourist's higher elasticity and command a higher price, however the opposite may also be true because of New York's constrained housing supply. We can also hypothesize that hosts will charge different prices for each 'neighbourhood_group', if true this will be an excellent entry point for further research.

Data Loading and Cleaning (New York Airbnb Dataset)

First we load in our dataset. The dataset is already pretty clean so there is no need to drop any rows.

[ ]:
```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
```

```python
import warnings
warnings.filterwarnings('ignore')

np.random.seed(0)

# Load in data
df = pd.read_csv("../Data/AB_NYC_2019.csv")

# Get snapshot of data
df.head()
```

```
[ ]:      id                                              name  host_id  \
    0  2539                Clean & quiet apt home by the park     2787
    1  2595                             Skylit Midtown Castle     2845
    2  3647               THE VILLAGE OF HARLEM…NEW YORK !     4632
    3  3831                   Cozy Entire Floor of Brownstone     4869
    4  5022  Entire Apt: Spacious Studio/Loft by central park     7192

         host_name neighbourhood_group neighbourhood  latitude  longitude  \
    0         John            Brooklyn    Kensington  40.64749  -73.97237
    1     Jennifer           Manhattan       Midtown  40.75362  -73.98377
    2    Elisabeth           Manhattan        Harlem  40.80902  -73.94190
    3  LisaRoxanne            Brooklyn  Clinton Hill  40.68514  -73.95976
    4        Laura           Manhattan   East Harlem  40.79851  -73.94399

             room_type  price  minimum_nights  number_of_reviews last_review  \
    0     Private room    149               1                  9  2018-10-19
    1  Entire home/apt    225               1                 45  2019-05-21
    2     Private room    150               3                  0         NaN
    3  Entire home/apt     89               1                270  2019-07-05
    4  Entire home/apt     80              10                  9  2018-11-19

       reviews_per_month  calculated_host_listings_count  availability_365
    0               0.21                               6               365
    1               0.38                               2               355
    2                NaN                               1               365
    3               4.64                               1               194
    4               0.10                               1                 0
```

```python
[ ]: # Summary Statistics for dependent and independent variables
df[['reviews_per_month', 'room_type', 'minimum_nights', 'availability_365',␣
 ↪'calculated_host_listings_count', 'price']].describe(include = 'all')
```

```
[ ]:        reviews_per_month        room_type  minimum_nights  availability_365  \
    count        38843.000000            48895    48895.000000      48895.000000
    unique                NaN                3             NaN               NaN
    top                   NaN  Entire home/apt             NaN               NaN
```

```
freq                NaN          25409           NaN            NaN
mean           1.373221            NaN      7.029962     112.781327
std            1.680442            NaN     20.510550     131.622289
min            0.010000            NaN      1.000000       0.000000
25%            0.190000            NaN      1.000000       0.000000
50%            0.720000            NaN      3.000000      45.000000
75%            2.020000            NaN      5.000000     227.000000
max           58.500000            NaN   1250.000000     365.000000

        calculated_host_listings_count         price
count                     48895.000000  48895.000000
unique                             NaN           NaN
top                                NaN           NaN
freq                               NaN           NaN
mean                          7.143982    152.720687
std                          32.952519    240.154170
min                           1.000000      0.000000
25%                           1.000000     69.000000
50%                           1.000000    106.000000
75%                           2.000000    175.000000
max                         327.000000  10000.000000
```

[ ]:

```
# Checking for missing values - seems all good, missing last reviews and
↪reviews_per_month are not unusual
# missing names are not a concern since we can use host_id should we want other
↪airbnb data
df.isnull().sum()
```

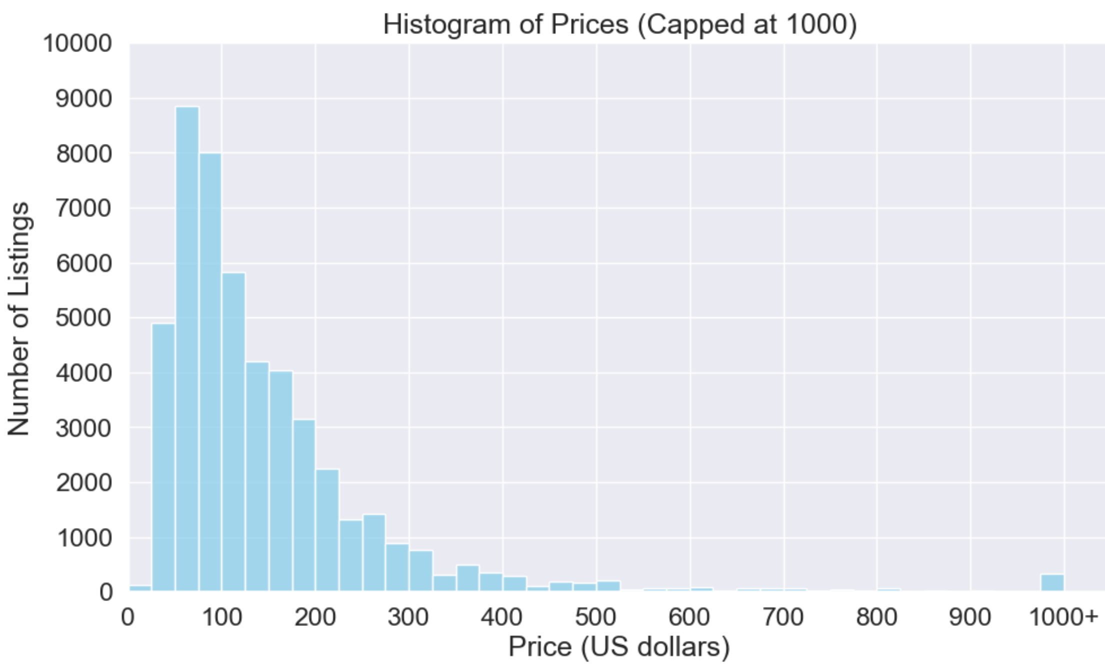
[ ]:

```
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
```

Plots, Histograms, Figures

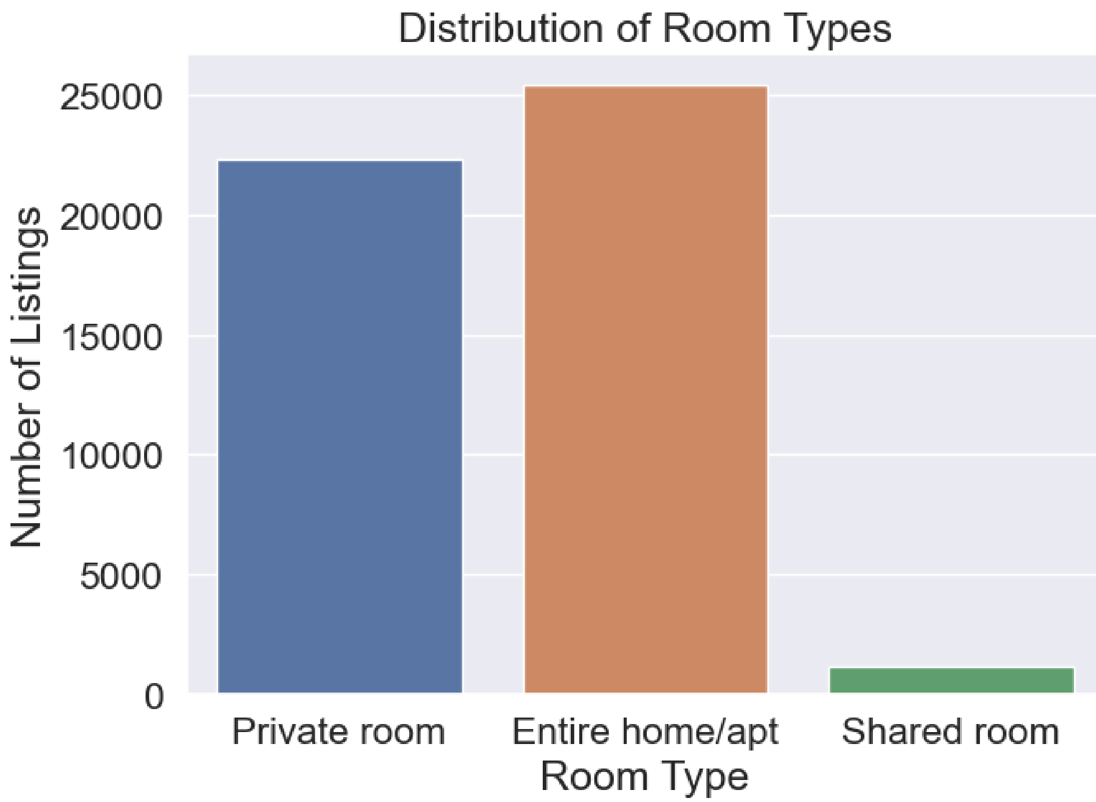
We now plot some relevant graphs starting with the required histograms of our variables.

[ ]:
```python
# In order to produce a more appealing graph we cap the x axis to 1000, note
↪that there are 239 listings priced at over $1000 a night
sns.set(font_scale=1.5)

df_copy = df.copy()
df_copy['price_capped'] = df_copy['price'].apply(lambda x: min(x, 1000))

plt.figure(figsize=(10, 6))
sns.histplot(df_copy['price_capped'], bins=range(0, 1025, 25), color='skyblue')

plt.title('Histogram of Prices (Capped at 1000)')
plt.xlabel('Price (US dollars)')
plt.ylabel('Number of Listings')

xticks = range(0, 1025, 100)
xtick_labels = [str(x) for x in xticks[:-1]] + ['1000+']
plt.xticks(xticks, xtick_labels)
y_ticks = range(0, 10000 + 1, 1000)
plt.yticks(y_ticks)

plt.xlim(left=0)
plt.ylim(bottom=0)

plt.show()
number_of_nights_over_30 = df[df['price'] > 1000].shape[0]
number_of_nights_over_30
```

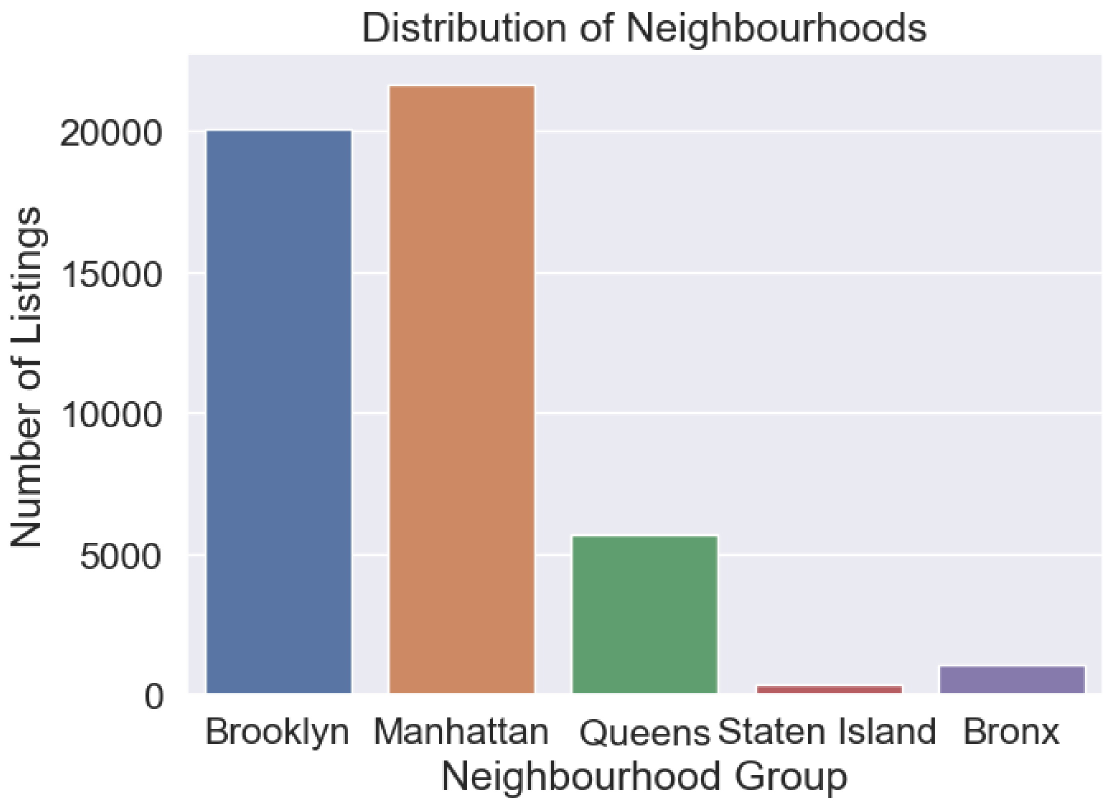


[ ]: 239

[ ]:
```
sns.countplot(x='room_type',data=df)
plt.title("Distribution of Room Types")
plt.xlabel('Room Type')
plt.ylabel('Number of Listings')
plt.show()
```

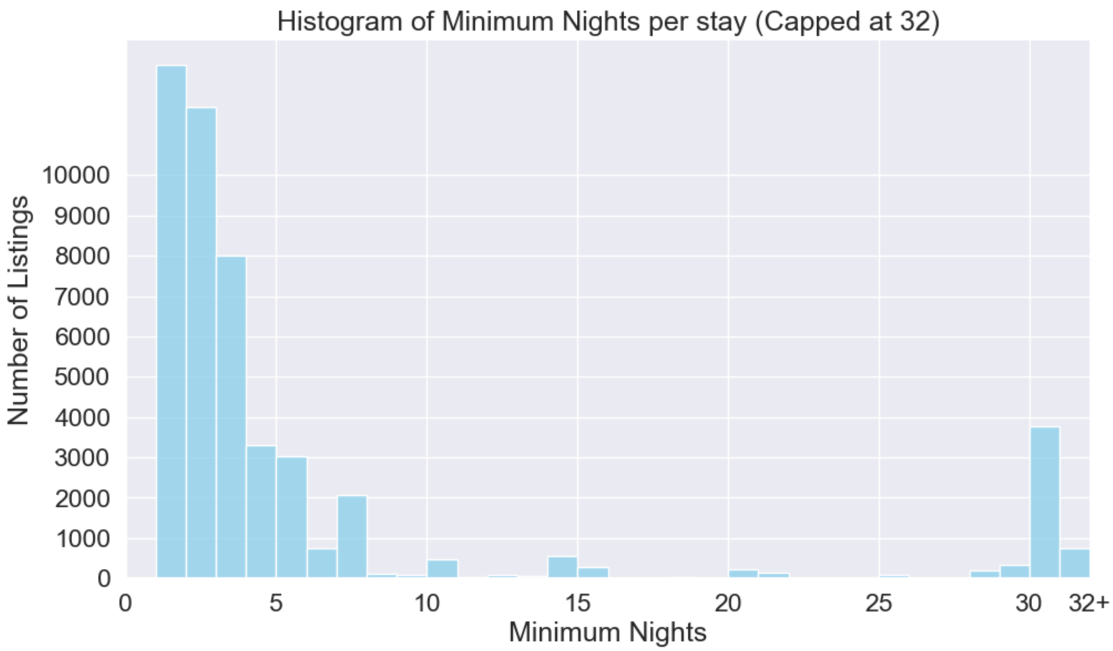



[ ]:
```
sns.countplot(x='neighbourhood_group',data=df)
plt.title("Distribution of Neighbourhoods")
plt.xlabel('Neighbourhood Group')
plt.ylabel('Number of Listings')
plt.show()
```



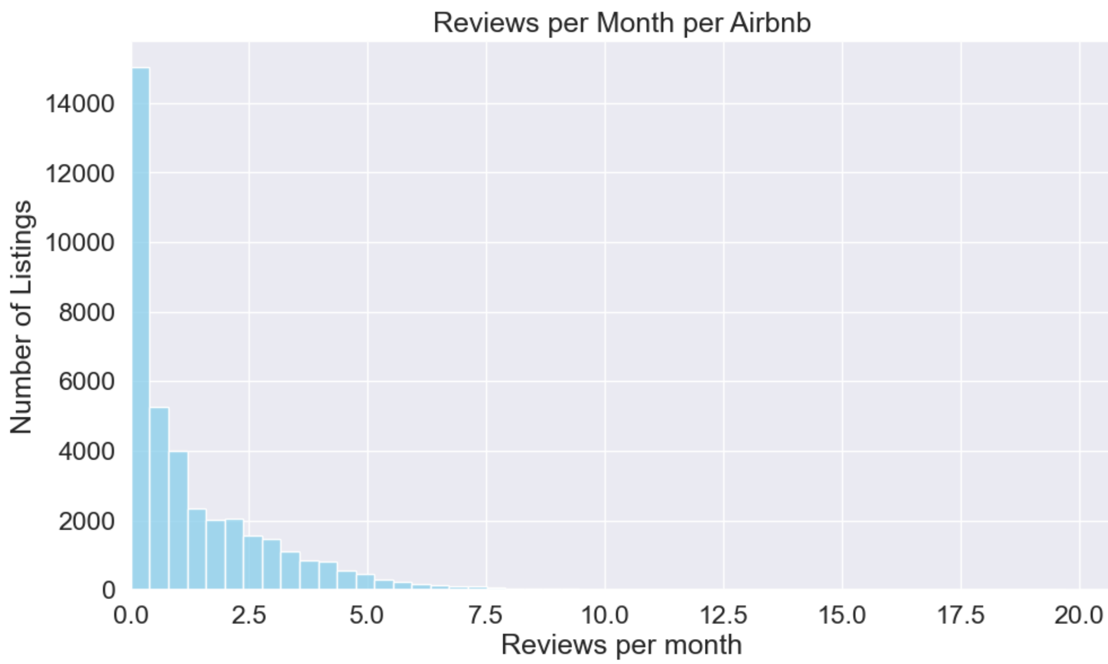

[ ]:
```
# In order to produce an appealing graph we bin values over 32 nights, it is␣
↪worth noting there are 538 listings with over 32 minimum nights per stay
df_pcopy = df.copy()
df_pcopy['minimum_nights_capped'] = df_pcopy['minimum_nights'].apply(lambda x:␣
↪min(x, 32))

plt.figure(figsize=(10, 6))
sns.histplot(df_pcopy['minimum_nights_capped'], bins=range(0, 33, 1),␣
↪color='skyblue')

plt.title('Histogram of Minimum Nights per stay (Capped at 32)')
plt.xlabel('Minimum Nights')
plt.ylabel('Number of Listings')

xticks = list(range(0, 33, 5)) + [32]
xtick_labels = [str(x) for x in xticks[:-1]] + ['32+']
plt.xticks(xticks, xtick_labels)

plt.xlim(0, 32)
plt.ylim(0)
y_ticks = range(0, 10000 + 1, 1000)
```

```
plt.yticks(y_ticks)

plt.show()
number_of_nights_over_32 = df_pcopy[df_pcopy['minimum_nights'] > 32].shape[0]
number_of_nights_over_32
```




[ ]: 538

[ ]:
```
# we drop significant outliers to produce a better histogram as there are only␣
↪3 units with more than 20 reviews a month
filtered_df = df[df['reviews_per_month'] <= 20]
plt.figure(figsize=(10, 6))
sns.histplot(filtered_df['reviews_per_month'], bins=50, kde=False,␣
↪color='skyblue')
plt.title('Reviews per Month per Airbnb')
plt.xlabel('Reviews per month')
plt.ylabel('Number of Listings')
plt.xlim(0)
plt.ylim(0)
plt.show()

number_of_reviews_over_30 = df[df['reviews_per_month'] > 20].shape[0]
number_of_reviews_over_30
```



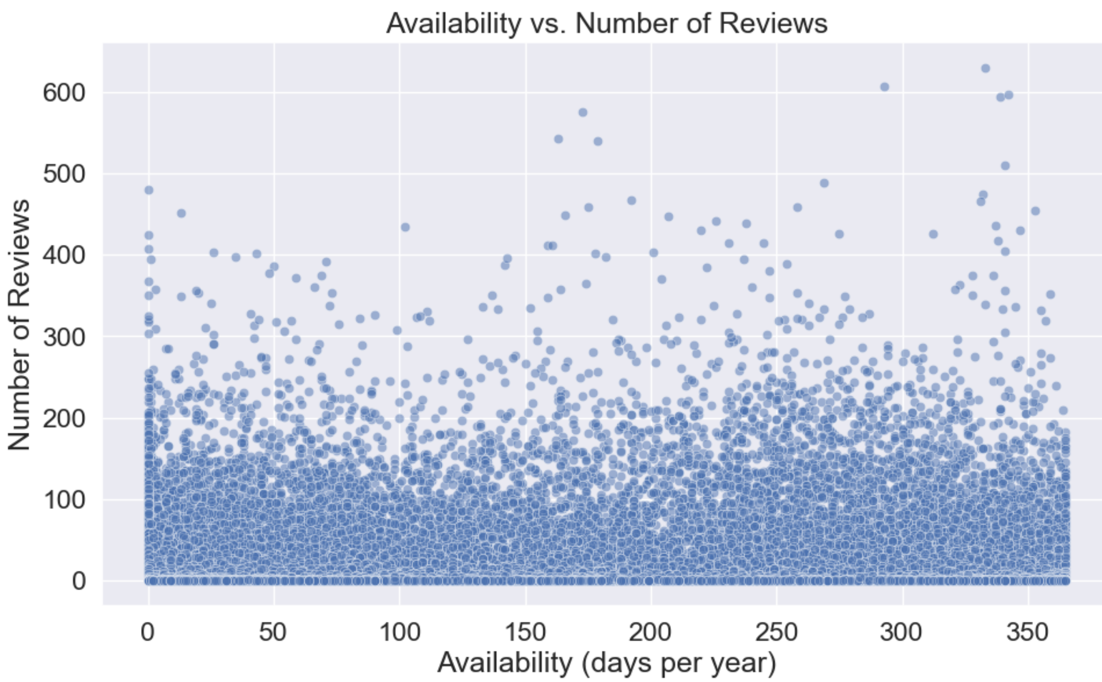

[ ]: 3

[ ]:
```
sns.histplot(filtered_df['availability_365'], bins=50, color='skyblue')
plt.title('Available Days Per Airbnb Listing')
plt.xlabel('Availability')
plt.ylabel('Number of Listings')
plt.xlim(0, 365)
plt.ylim(0)
plt.show()

availble0 = df[df['availability_365'] == 0].shape[0]
availble0
```


[ ]: 17533

We have finished plotting our required histograms. One thing that immediately sticks out is that there are an unusually high number of Airbnb's with 0 available days, wether or not this is because they are fully booked or unavailable or cant sell is unclear, so we can plot it against reviews per month to help see. Identifying and eliminating units which are completely unavailable for some reason will help our research question.

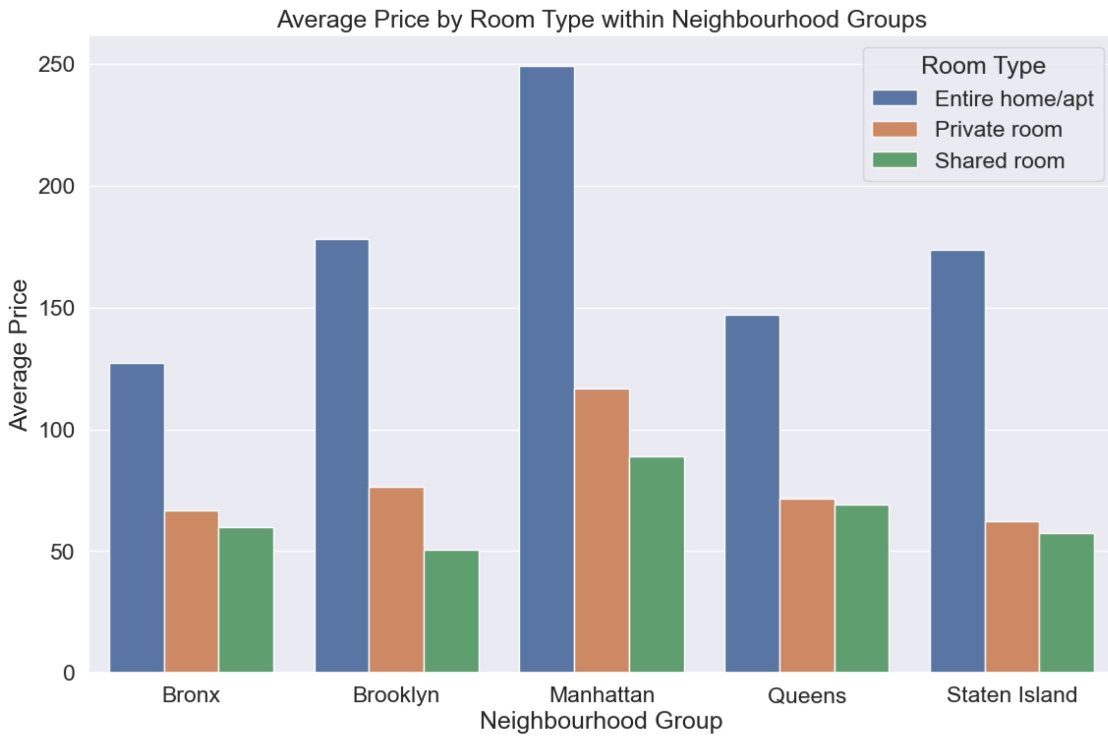
[ ]:
```
plt.figure(figsize=(10, 6))

sns.scatterplot(data=df, x='availability_365', y='number_of_reviews', alpha=0.5)

plt.title('Availability vs. Number of Reviews')
plt.xlabel('Availability (days per year)')
plt.ylabel('Number of Reviews')

plt.show()
```




There appears to be no correlation between availability and number of reviews. Many units with 0 available days still have a normal amount of reviews so its likely they are selling fine. Lets now plot price against some of our independent variables.

From the following graph we can see that Entire home/apt listings, as well listings in Brooklyn and Mahhattan command higher prices than other room types and boroughs. This is an anomaly and examining the reasons why will be a key part of future analysis.

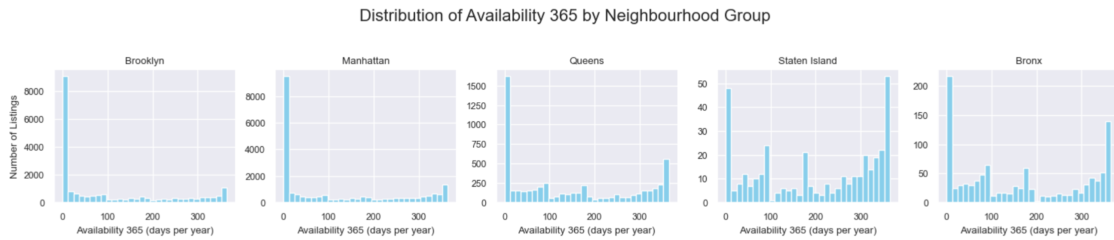
[ ]:
```
# Assuming 'df' is your DataFrame with the Airbnb data
# Calculate the average price for each room type within each neighbourhood group
grouped_data = df.groupby(['neighbourhood_group', 'room_type'])['price'].mean().
    ↪reset_index()

# Create a bar plot with Seaborn
plt.figure(figsize=(12, 8))
sns.barplot(data=grouped_data, x='neighbourhood_group', y='price',␣
    ↪hue='room_type')
plt.title('Average Price by Room Type within Neighbourhood Groups')
plt.xlabel('Neighbourhood Group')
plt.ylabel('Average Price')
plt.legend(title='Room Type')
plt.show()
```




Plotting the availbility_365 by each neighbourhood we can see that the Queens, Staten Island and the Bronx have an unusually high Airbnb vacancy rate compared to their amount of listings, where as Brooklyn and Manhattan have very low availability. This could suggest a further path for research, seeing what socioeconomic differences there are between Brooklyn and Manhattan, vs Queens, Staten Island and the Bronx.

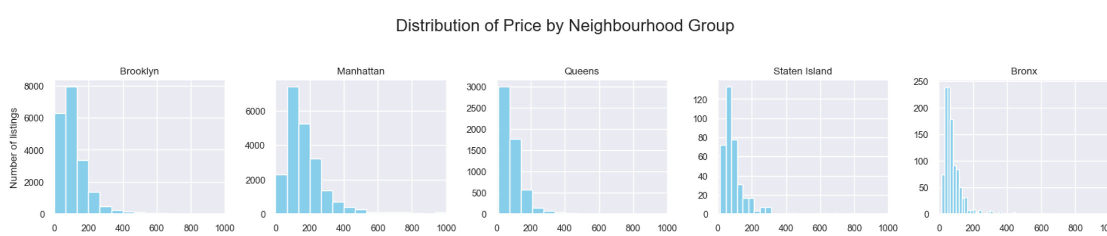
[ ]:
```
current_context = sns.plotting_context()

with sns.plotting_context('notebook', font_scale=0.8):
    g = sns.FacetGrid(df, col='neighbourhood_group', sharex=False, sharey=False)

    g.map(plt.hist, 'availability_365', bins=30, color='skyblue')

    g.set_axis_labels('Availability 365 (days per year)', 'Number of Listings')
    g.set_titles('{col_name}')
    for ax in g.axes.flat:
        ax.title.set_position([0.5, 1.05])

    if g.fig._suptitle is not None:
        g.fig._suptitle.set_visible(False)
    g.fig.suptitle('Distribution of Availability 365 by Neighbourhood Group',␣
 ↪fontsize=16, y=1.05)

    g.tight_layout(w_pad=1, h_pad=1)
```

```
    plt.show()

sns.set_context(current_context)
```



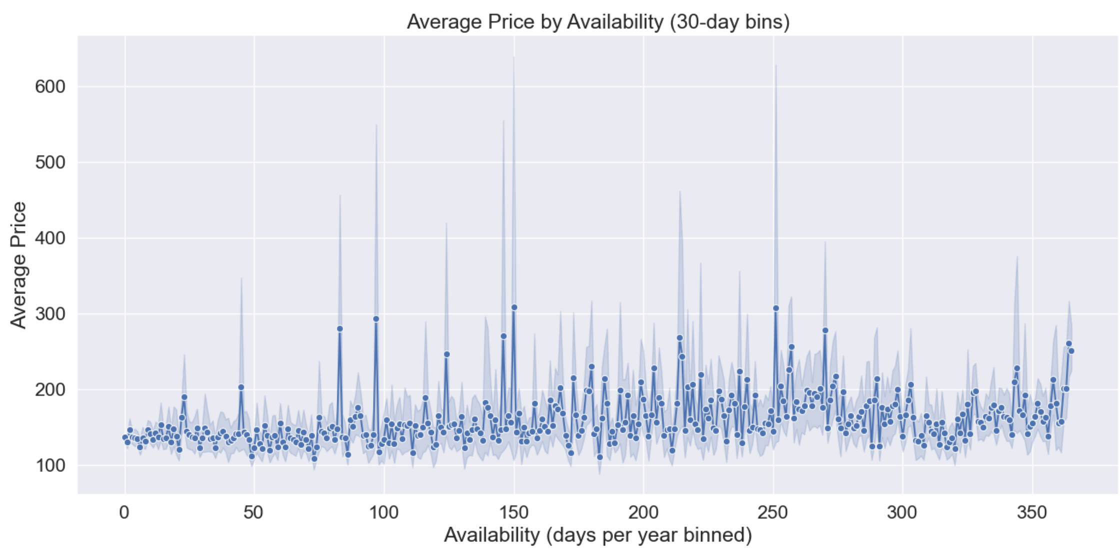

Plotting the distribution of price by each neighbourhood group shows that Manhattan has a higher share of units which command higher prices.

```
[ ]: current_context = sns.plotting_context()

with sns.plotting_context('notebook', font_scale=0.8):
    g = sns.FacetGrid(df, col='neighbourhood_group', sharex=False, sharey=False)

    g.map(plt.hist, 'price', bins=150, color='skyblue')

    g.set_axis_labels('Price', 'Number of listings')
    g.set_titles('{col_name}')
    for ax in g.axes.flat:
        ax.title.set_position([0.5, 1.05])

    if g.fig._suptitle is not None:
        g.fig._suptitle.set_visible(False)
    g.fig.suptitle('Distribution of Price by Neighbourhood Group', fontsize=16,␣
 ↪y=1.05)
    g.set(xlim=(0, 1000))
    g.tight_layout(w_pad=1, h_pad=1)

    plt.show()

sns.set_context(current_context)
```



We can speculate that units with lower availabillity 365 may have a higher price because of higher demand and create a line graph to see if thats true. However the graph does not conclusively support that theory.

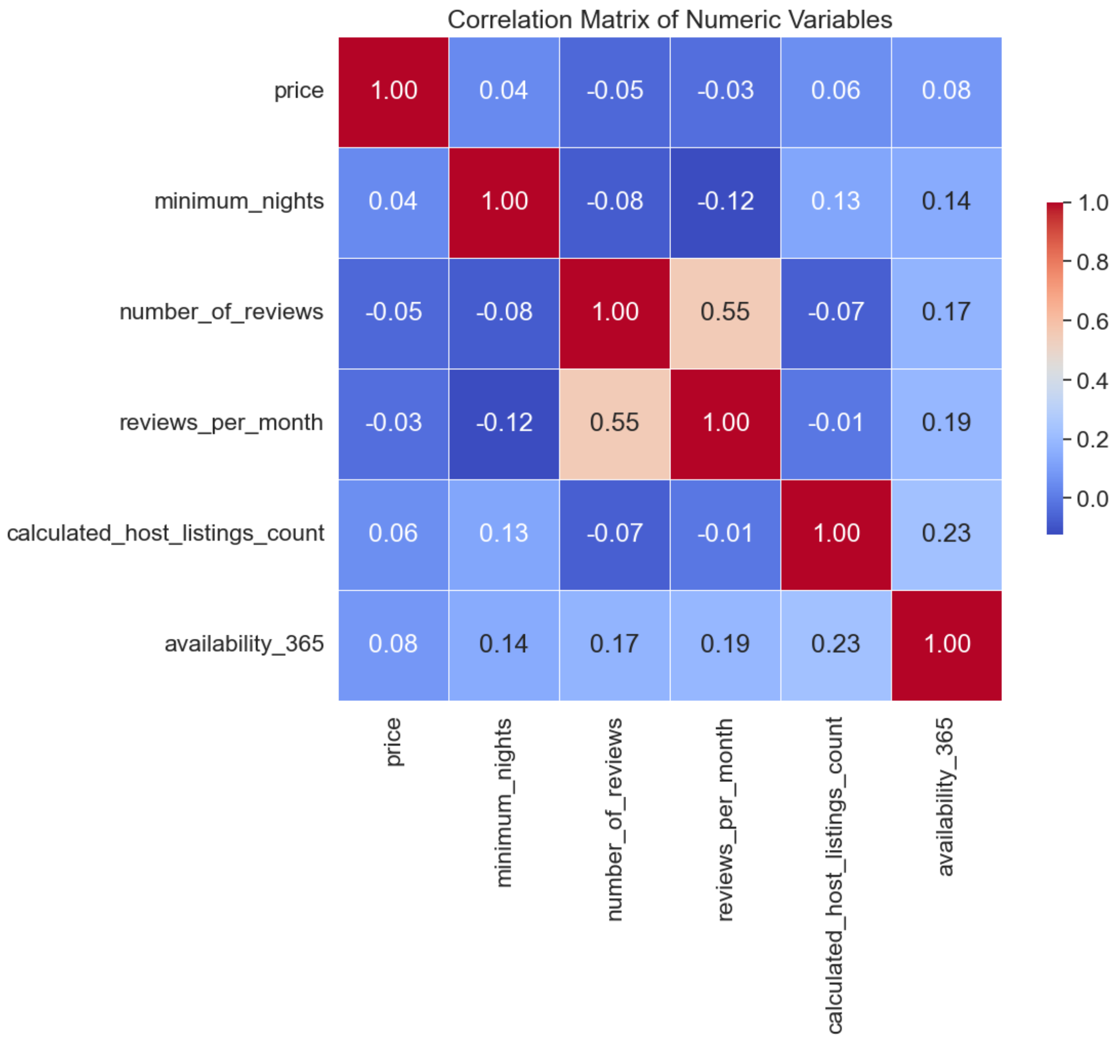
```
[ ]: df['availability_binned'] = pd.cut(df['availability_365'], bins=np.arange(0,␣
     ↪365, 30), include_lowest=True)
     plt.figure(figsize=(14, 7))
     sns.lineplot(data=df, x='availability_365', y='price', marker='o')
     plt.title('Average Price by Availability (30-day bins)')
     plt.xlabel('Availability (days per year binned)')
     plt.ylabel('Average Price')
     plt.tight_layout()
     plt.show()
```




Lastly creating a correlation matrix may allow us to see any relation we may have missed.

```
[ ]: numeric_df = df.select_dtypes(include=[np.number])
     numeric_df = numeric_df.drop(['id','host_id', 'latitude', 'longitude'], axis=1)

     # Calculate the correlation matrix
     corr_matrix = numeric_df.corr()

     # Set up the matplotlib figure
     plt.figure(figsize=(10, 8))

     # Draw the heatmap with the mask and correct aspect ratio
     sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True,
```

```
            linewidths=.5, cbar_kws={"shrink": .5})

# Add title and labels
plt.title('Correlation Matrix of Numeric Variables')

# Show plot
plt.show()
```




There is little interesting data to glean from the correlation matrix.

Conclusion

In conclusion our exploration has uncovered several anomalies which can serve to narrow down our research path. We have noted that Manhattan has the highest price in New York City for Airbnb listing's and that Entire home/apartment listings also have higher prices. We have also learned that Brooklyn and Manhattan have a significantly larger portion of Airbnb's compared to Queens, the Bronx, and Staten Island, and that Airbnb's in those areas have a higher availability (indicating that they may be rented less). Understanding what underlying characteristics and socioeconomic factors may be causing this disparity will advance our research question.

The distribution of 'minimum_nights' also suggested that short stays are the norm, with most hosts preferring rentals that do not exceed 30 nights, which is indicative of the short-term nature of Airbnb rentals. However the portion of longer 'minimium_nights' rentals were not insignificant.

The analysis of pricing showed a wide range of accommodation costs, with a small fraction of listings priced significantly higher than the median. These listings did not conform to the general trend and often represented luxury accommodations or properties in high-demand areas. Our look into higher-priced listings revealed that these properties do not significantly differ in their availability throughout the year compared to more moderately priced options, suggesting that pricing strategies are varied and do not necessarily correlate with increased occupancy.

Overall, our biggest takeaway is in observing the disparity between Brooklyn and Manhattan, vs Queens, the Bronx, and Staten Island. This presents a viable research path, as we potentially merge in other data.

Identify price outliers by controlling for population