Project Kick-off: Airbnb Listings Analysis

Project Overview

This project focuses on analyzing Airbnb listings data, including listing prices, room types, locations, and guest reviews. The dataset provides a detailed view of properties across different cities, their pricing strategies, and customer preferences. This makes it ideal for extracting insights on property performance, pricing trends, and guest satisfaction.

Objective

The main goal of this project is to explore Airbnb listings data to answer key business questions:

- Which room types or neighborhoods generate the most bookings or revenue?
- How do prices vary across cities and property types?
- Are there seasonal trends in bookings or reviews?
- How do guest ratings correlate with price, location, or property type?

Tools & Approach

- **Pandas:** Load, clean, and explore the dataset
- **Matplotlib & Seaborn:** Create static visualizations for trends and comparisons
- **Plotly:** Generate interactive charts for deeper insights
- **Jupyter/Colab Notebook:** Structured workflow with explanations and charts

Deliverables

The final notebook will include:

1. Dataset preparation and cleaning
2. Exploratory data analysis (EDA) with summary statistics
3. Visualizations: bar charts, line charts, pie charts, heatmaps, and histograms
4. Dashboard-like presentation with key insights

DataSet Preparation

In [19]:
```
# Import libraries and load dataset
from google.colab import files
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Upload the dataset
uploaded = files.upload()
# Load the Airbnb dataset
df = pd.read_csv("Airbnb_Open_Data.csv",low_memory=False)
df.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Airbnb_Open_Data.csv to Airbnb_Open_Data (1).csv
```

Out[19]:

	id	NAME	host id	host_identity_verified	host name	neigh
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	
3	1002755	NaN	85098326012	unconfirmed	Garry	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	

5 rows × 26 columns

In [18]:
```
# Basic dataset overview
print("Shape:", df.shape)
print("Columns:", df.columns.tolist())
```

```python
df.info()
df.describe(include="all").transpose()
```

```
Shape: (102599, 27)
Columns: ['id', 'name', 'host_id', 'host_identity_verified', 'host_name', 'n
eighbourhood_group', 'neighbourhood', 'lat', 'long', 'country', 'country_cod
e', 'instant_bookable', 'cancellation_policy', 'room_type', 'construction_ye
ar', 'price', 'service_fee', 'minimum_nights', 'number_of_reviews', 'last_re
view', 'reviews_per_month', 'review_rate_number', 'calculated_host_listings_
count', 'availability_365', 'house_rules', 'license', 'month']
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 27 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   name                            102349 non-null  object
 2   host_id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host_name                       102193 non-null  object
 5   neighbourhood_group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country_code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room_type                       102599 non-null  object
 14  construction_year               102385 non-null  float64
 15  price                           102352 non-null  float64
 16  service_fee                     102326 non-null  object
 17  minimum_nights                  102190 non-null  float64
 18  number_of_reviews               102416 non-null  float64
 19  last_review                     86706 non-null   datetime64[ns]
 20  reviews_per_month               86720 non-null   float64
 21  review_rate_number              102273 non-null  float64
 22  calculated_host_listings_count  102280 non-null  float64
 23  availability_365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
 26  month                           86706 non-null   period[M]
dtypes: datetime64[ns](1), float64(10), int64(2), object(13), period[M](1)
memory usage: 21.1+ MB
```

Out[18]:

	count	unique	top	freq	
id	102599.0	NaN	NaN	NaN	2914
name	102349	61281	Home away from home	33	
host_id	102599.0	NaN	NaN	NaN	4925411
host_identity_verified	102310	2	unconfirmed	51200	
host_name	102193	13190	Michael	881	
neighbourhood_group	102570	7	Manhattan	43792	
neighbourhood	102583	224	Bedford-Stuyvesant	7937	
lat	102591.0	NaN	NaN	NaN	
long	102591.0	NaN	NaN	NaN	
country	102067	1	United States	102067	
country_code	102468	1	US	102468	
instant_bookable	102494	2	False	51474	
cancellation_policy	102523	3	moderate	34343	
room_type	102599	4	Entire home/apt	53701	
construction_year	102385.0	NaN	NaN	NaN	2
price	102352.0	NaN	NaN	NaN	
service_fee	102326	231	$41	526	
minimum_nights	102190.0	NaN	NaN	NaN	
number_of_reviews	102416.0	NaN	NaN	NaN	
last_review	86706	NaN	NaN	NaN	03:40:5
reviews_per_month	86720.0	NaN	NaN	NaN	
review_rate_number	102273.0	NaN	NaN	NaN	
calculated_host_listings_count	102280.0	NaN	NaN	NaN	
availability_365	102151.0	NaN	NaN	NaN	
house_rules	50468	1976	#NAME?	2712	
license	2	1	41662/AL	2	
month	86706	116	2019-06	21497	

In [ ]:

```
# Clean dataset
df = df.drop_duplicates()
df = df.dropna(how="all")
```

```python
df = df.fillna({
    "price": 0,
    "reviews_per_month": 0,
    "number_of_reviews": 0,
    "minimum_nights": 0
})
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")
if 'last_review' in df.columns:
    df['last_review'] = pd.to_datetime(df['last_review'], errors="coerce")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 102058 entries, 0 to 102057
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102058 non-null  int64
 1   name                            101808 non-null  object
 2   host_id                         102058 non-null  int64
 3   host_identity_verified          101769 non-null  object
 4   host_name                       101654 non-null  object
 5   neighbourhood_group             102029 non-null  object
 6   neighbourhood                   102042 non-null  object
 7   lat                             102050 non-null  float64
 8   long                            102050 non-null  float64
 9   country                         101526 non-null  object
 10  country_code                    101927 non-null  object
 11  instant_bookable                101953 non-null  object
 12  cancellation_policy             101982 non-null  object
 13  room_type                       102058 non-null  object
 14  construction_year               101844 non-null  float64
 15  price                           102058 non-null  object
 16  service_fee                     101785 non-null  object
 17  minimum_nights                  101658 non-null  float64
 18  number_of_reviews               101875 non-null  float64
 19  last_review                     86226 non-null   datetime64[ns]
 20  reviews_per_month               86240 non-null   float64
 21  review_rate_number              101739 non-null  float64
 22  calculated_host_listings_count  101739 non-null  float64
 23  availability_365                101610 non-null  float64
 24  house_rules                     50216 non-null   object
 25  license                         2 non-null       object
dtypes: datetime64[ns](1), float64(9), int64(2), object(14)
memory usage: 21.0+ MB
```

In [ ]:

```python
# Convert price and service_fee to numeric
df['price'] = pd.to_numeric(df['price'], errors='coerce')
df['service_fee'] = pd.to_numeric(df['service_fee'], errors='coerce')

# Fill remaining missing numerical values
df['minimum_nights'] = df['minimum_nights'].fillna(0)
df['number_of_reviews'] = df['number_of_reviews'].fillna(0)
df['reviews_per_month'] = df['reviews_per_month'].fillna(0)
df['review_rate_number'] = df['review_rate_number'].fillna(0)
df['calculated_host_listings_count'] = df['calculated_host_listings_count'].
df['availability_365'] = df['availability_365'].fillna(0)
```

```python
# Verify the cleaned dataset
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 102058 entries, 0 to 102057
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102058 non-null  int64
 1   name                            101808 non-null  object
 2   host_id                         102058 non-null  int64
 3   host_identity_verified          101769 non-null  object
 4   host_name                       101654 non-null  object
 5   neighbourhood_group             102029 non-null  object
 6   neighbourhood                   102042 non-null  object
 7   lat                             102050 non-null  float64
 8   long                            102050 non-null  float64
 9   country                         101526 non-null  object
 10  country_code                    101927 non-null  object
 11  instant_bookable                101953 non-null  object
 12  cancellation_policy             101982 non-null  object
 13  room_type                       102058 non-null  object
 14  construction_year               101844 non-null  float64
 15  price                           247 non-null     float64
 16  service_fee                     0 non-null       float64
 17  minimum_nights                  102058 non-null  float64
 18  number_of_reviews               102058 non-null  float64
 19  last_review                     86226 non-null   datetime64[ns]
 20  reviews_per_month               102058 non-null  float64
 21  review_rate_number              102058 non-null  float64
 22  calculated_host_listings_count  102058 non-null  float64
 23  availability_365                102058 non-null  float64
 24  house_rules                     50216 non-null   object
 25  license                         2 non-null       object
dtypes: datetime64[ns](1), float64(11), int64(2), object(12)
memory usage: 21.0+ MB
```

Out[ ]:

	id	name	host_id	host_identity_verified	host_name	ne
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	
3	1002755	NaN	85098326012	unconfirmed	Garry	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	

5 rows × 26 columns

Exploratory Data Analysis (EDA)

In [ ]:

```
# Step 3: Exploratory Data Analysis (EDA)

# Summary statistics and missing values
print("Summary statistics:")
print(df.describe().transpose())
print("\nMissing values per column:")
print(df.isnull().sum())

# Check unique values for key categorical columns
categorical_cols = ['host_identity_verified', 'neighbourhood', 'country',
                    'instant_bookable', 'cancellation_policy', 'room_type']
print("\nUnique values for key categorical columns:")
for col in categorical_cols:
    if col in df.columns:
        print(col, ":", df[col].nunique())

# Correlation between numerical columns
numerical_cols = ['price', 'minimum_nights', 'number_of_reviews',
                  'reviews_per_month', 'review_rate_number',
                  'calculated_host_listings_count',
                  'availability_365', 'service_fee']
numerical_cols = [col for col in numerical_cols if col in df.columns]
print("\nCorrelation between numerical columns:")
print(df[numerical_cols].corr())
```

```
# Top 10 hosts by number of listings
if 'host_name' in df.columns:
    top_hosts = df.groupby('host_name')['id'].count().sort_values(ascending=
    print("\nTop 10 hosts by number of listings:")
    print(top_hosts)

# Monthly reviews trends
if 'last_review' in df.columns:
    df['month'] = df['last_review'].dt.to_period('M')
    monthly_reviews = df.groupby('month')['id'].count()
    print("\nMonthly reviews count (first 12 months):")
    print(monthly_reviews.head(12))
```

```
Summary statistics:
                                  count                           mean  \
id                             102599.0                 29146234.52213
host_id                        102599.0            49254111474.328667
lat                            102591.0                      40.728094
long                           102591.0                     -73.949644
construction_year              102385.0                    2012.487464
price                          102352.0                     625.293536
service_fee                    102326.0                     125.026924
minimum_nights                 102599.0                       8.103412
number_of_reviews              102599.0                      27.434722
last_review                       86706  2019-06-12 03:40:52.065601024
reviews_per_month              102599.0                       1.161368
review_rate_number             102599.0                       3.268687
calculated_host_listings_count 102599.0                       7.911929
availability_365               102599.0                     140.516993

                                               min                  25%  \
id                                       1001254.0           15085814.5
host_id                                123600518.0        24583328475.0
lat                                       40.49979             40.68874
long                                     -74.24984            -73.98258
construction_year                           2003.0               2007.0
price                                         50.0                340.0
service_fee                                   10.0                 68.0
minimum_nights                             -1223.0                  1.0
number_of_reviews                              0.0                  1.0
last_review                    2012-07-11 00:00:00  2018-10-28 00:00:00
reviews_per_month                              0.0                 0.09
review_rate_number                             0.0                  2.0
calculated_host_listings_count                 0.0                  1.0
availability_365                             -10.0                  2.0

                                               50%                  75%  \
id                                      29136603.0           43201198.0
host_id                              49117739352.0        73996495817.0
lat                                       40.72229             40.76276
long                                     -73.95444            -73.93235
construction_year                           2012.0               2017.0
price                                        624.0                913.0
service_fee                                  125.0                183.0
minimum_nights                                 3.0                  5.0
number_of_reviews                              7.0                 30.0
last_review                    2019-06-14 00:00:00  2019-07-05 00:00:00
reviews_per_month                             0.48                 1.71
review_rate_number                             3.0                  4.0
calculated_host_listings_count                 1.0                  2.0
availability_365                              95.0                268.0

                                               max                 std
id                                      57367417.0     16257505.607309
host_id                              98763129024.0  28538996644.374817
lat                                       40.91697            0.055857
long                                     -73.70522            0.049521
construction_year                           2022.0            5.765556
price                                       1200.0          331.671614
```

```
service_fee                                  240.0          66.325739
minimum_nights                              5645.0          30.497129
number_of_reviews                           1024.0          49.478373
last_review                    2058-06-16 00:00:00                NaN
reviews_per_month                             90.0           1.680924
review_rate_number                             5.0           1.295823
calculated_host_listings_count               332.0          32.171688
availability_365                            3677.0         135.459024

Missing values per column:
id                                     0
name                                 250
host_id                                0
host_identity_verified               289
host_name                            406
neighbourhood_group                   29
neighbourhood                         16
lat                                    8
long                                   8
country                              532
country_code                         131
instant_bookable                     105
cancellation_policy                   76
room_type                              0
construction_year                    214
price                                247
service_fee                          273
minimum_nights                         0
number_of_reviews                      0
last_review                        15893
reviews_per_month                      0
review_rate_number                     0
calculated_host_listings_count         0
availability_365                       0
house_rules                        52131
license                           102597
dtype: int64

Unique values for key categorical columns:
host_identity_verified : 2
neighbourhood : 224
country : 1
instant_bookable : 2
cancellation_policy : 3
room_type : 4

Correlation between numerical columns:
                                   price  minimum_nights  number_of_reviews  \
price                           1.000000       -0.003417           0.005324
minimum_nights                 -0.003417        1.000000          -0.049915
number_of_reviews               0.005324       -0.049915           1.000000
reviews_per_month               0.005260       -0.092127           0.617857
review_rate_number             -0.003905       -0.002115          -0.021413
calculated_host_listings_count -0.000111        0.084608          -0.080657
availability_365               -0.002720        0.057903           0.097681
```

```
service_fee                        0.999991       -0.003597          0.005216

                                   reviews_per_month  review_rate_number  \
price                                       0.005260           -0.003905
minimum_nights                             -0.092127           -0.002115
number_of_reviews                           0.617857           -0.021413
reviews_per_month                           1.000000            0.031001
review_rate_number                          0.031001            1.000000
calculated_host_listings_count             -0.039924            0.024666
availability_365                            0.073163           -0.008357
service_fee                                 0.005102           -0.003723

                                   calculated_host_listings_count  \
price                                                   -0.000111
minimum_nights                                           0.084608
number_of_reviews                                       -0.080657
reviews_per_month                                       -0.039924
review_rate_number                                       0.024666
calculated_host_listings_count                           1.000000
availability_365                                         0.158912
service_fee                                              0.000035

                                   availability_365  service_fee
price                                     -0.002720     0.999991
minimum_nights                             0.057903    -0.003597
number_of_reviews                          0.097681     0.005216
reviews_per_month                          0.073163     0.005102
review_rate_number                        -0.008357    -0.003723
calculated_host_listings_count             0.158912     0.000035
availability_365                           1.000000    -0.003061
service_fee                               -0.003061     1.000000

Top 10 hosts by number of listings:
host_name
Michael         881
David           764
John            581
Alex            546
Sonder (NYC)    516
Daniel          473
Karen           439
Sarah           434
Maria           426
Anna            400
Name: id, dtype: int64

Monthly reviews count (first 12 months):
month
2012-07     2
2012-08     4
2012-09    10
2012-11     4
2012-12     6
2013-01     9
2013-03     4
2013-04     6
```

```
2013-05     8
2013-06     7
2013-07     5
2013-08     4
Freq: M, Name: id, dtype: int64
```

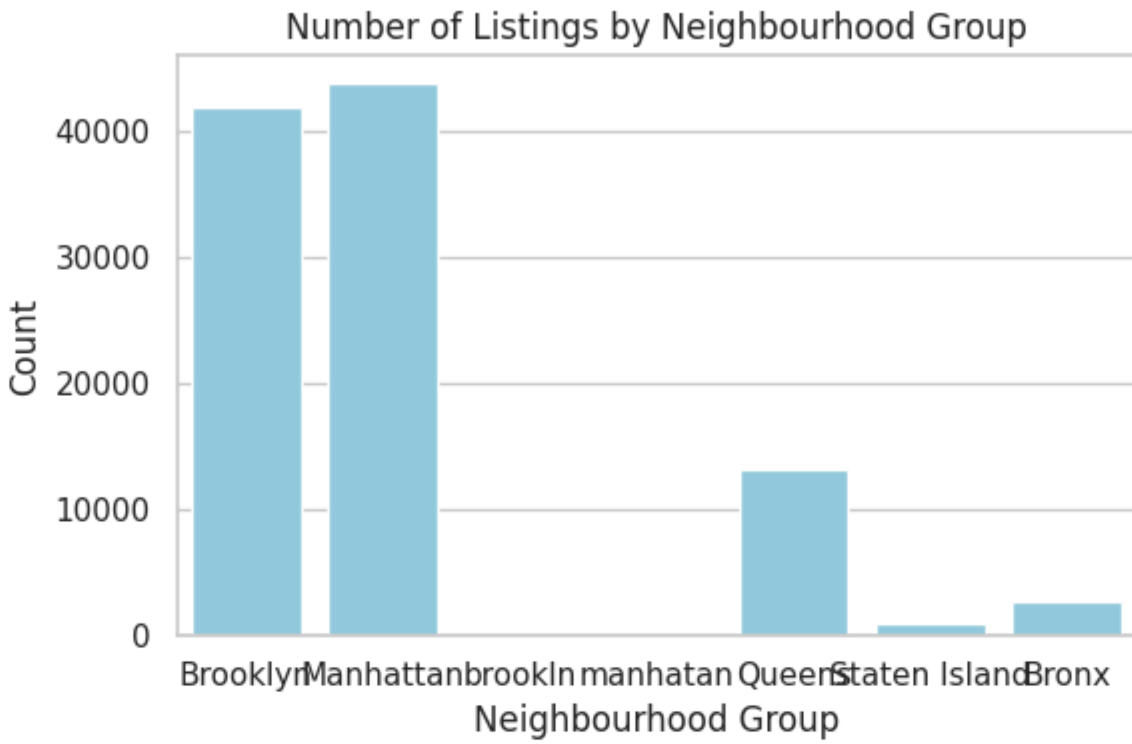
Data Visualization

In [ ]:

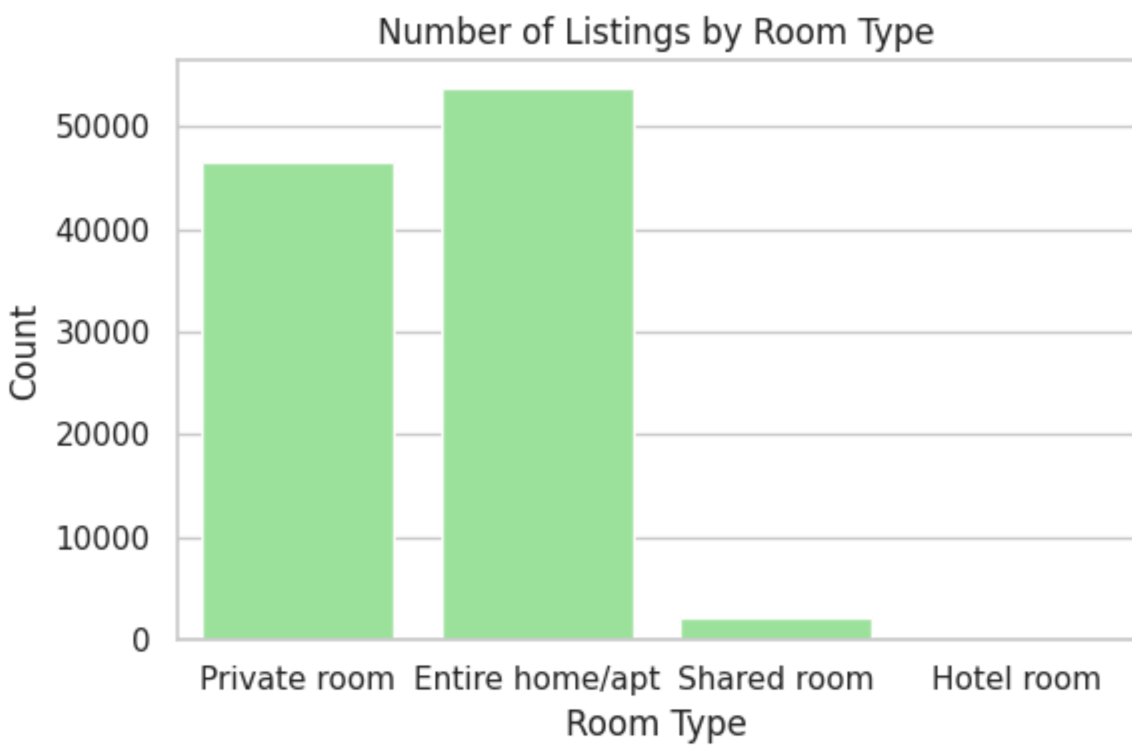
```
# Step 4 : VISUALIZATIONS

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

sns.set(style="whitegrid")

# Fix column names
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")

# Convert price and service_fee to numeric
df['price'] = df['price'].replace(r'[\$,]', '', regex=True).astype(float)
df['service_fee'] = df['service_fee'].replace(r'[\$,]', '', regex=True).asty

# Fill missing numeric values
for col in ['reviews_per_month','minimum_nights','number_of_reviews',
            'review_rate_number','calculated_host_listings_count','availabil
    if col in df.columns:
        df[col] = df[col].fillna(0)

# Convert last_review to datetime
if 'last_review' in df.columns:
    df['last_review'] = pd.to_datetime(df['last_review'], errors='coerce')

# Bar Chart: Listings by Neighbourhood Group
print("Bar Chart: Listings by Neighbourhood Group - shows the count of listi
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='neighbourhood_group', color='skyblue')
plt.title("Number of Listings by Neighbourhood Group")
plt.xlabel("Neighbourhood Group")
plt.ylabel("Count")
plt.tight_layout()
plt.show()

# Bar Chart: Listings by Room Type
print("\nBar Chart: Listings by Room Type - shows the count of listings per
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='room_type', color='lightgreen')
plt.title("Number of Listings by Room Type")
plt.xlabel("Room Type")
plt.ylabel("Count")
plt.tight_layout()
plt.show()



# Histogram: Price Distribution
```

```
print("\nHistogram: Price Distribution - shows how listing prices are distri
plt.figure(figsize=(8,4))
sns.histplot(df['price'], bins=50, color='skyblue')
plt.title("Price Distribution of Listings")
plt.xlabel("Price (USD)")
plt.ylabel("Number of Listings")
plt.xlim(0, 1000)
plt.tight_layout()
plt.show()

# Histogram: Number of Reviews
print("\nHistogram: Number of Reviews - shows distribution of review counts.
plt.figure(figsize=(8,4))
sns.histplot(df['number_of_reviews'], bins=50, color='lightgreen')
plt.title("Number of Reviews Distribution")
plt.xlabel("Number of Reviews")
plt.ylabel("Count")
plt.xlim(0, 200)
plt.tight_layout()
plt.show()

# Scatter Plot: Price vs Reviews per Month
print("\nScatter Plot: Price vs Reviews per Month - shows relationship betwe
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x='reviews_per_month', y='price', alpha=0.5)
plt.title("Price vs Reviews per Month")
plt.xlabel("Reviews per Month")
plt.ylabel("Price (USD)")
plt.ylim(0, 1000)
plt.tight_layout()
plt.show()

# Correlation Heatmap for Numerical Columns
numerical_cols = ['price','minimum_nights','number_of_reviews','reviews_per_
                  'review_rate_number','calculated_host_listings_count','ava
print("\nHeatmap: Correlation between numerical variables - shows relationsh
plt.figure(figsize=(8,6))
sns.heatmap(df[numerical_cols].corr(), annot=True, cmap='coolwarm', fmt=".2f
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()
```

```
Bar Chart: Listings by Neighbourhood Group - shows the count of listings per
neighbourhood group.
```

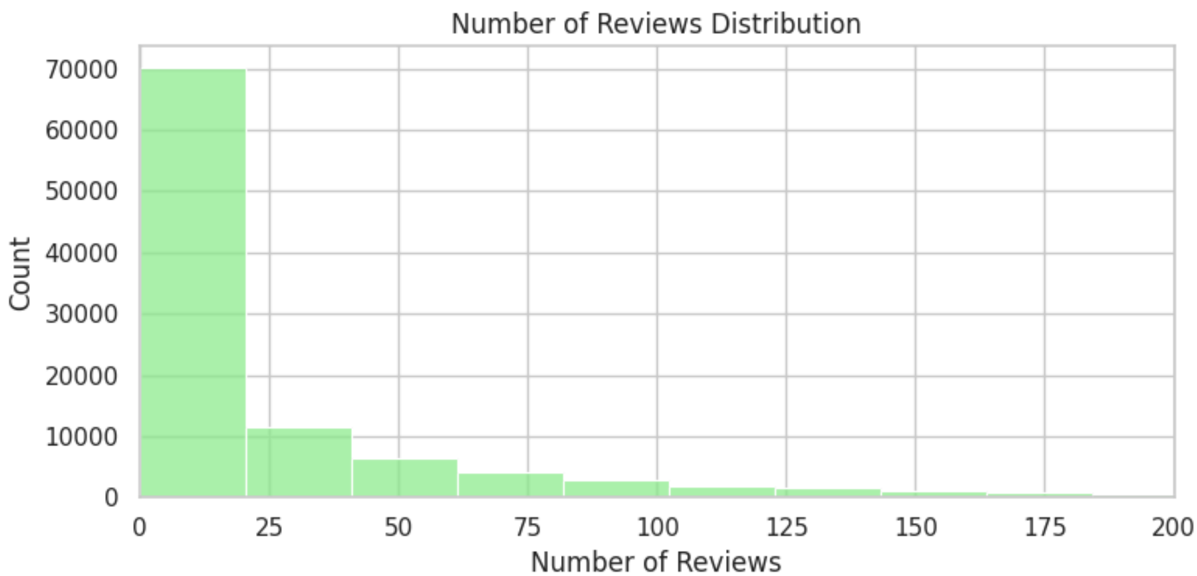


Bar Chart: Listings by Room Type - shows the count of listings per room type.



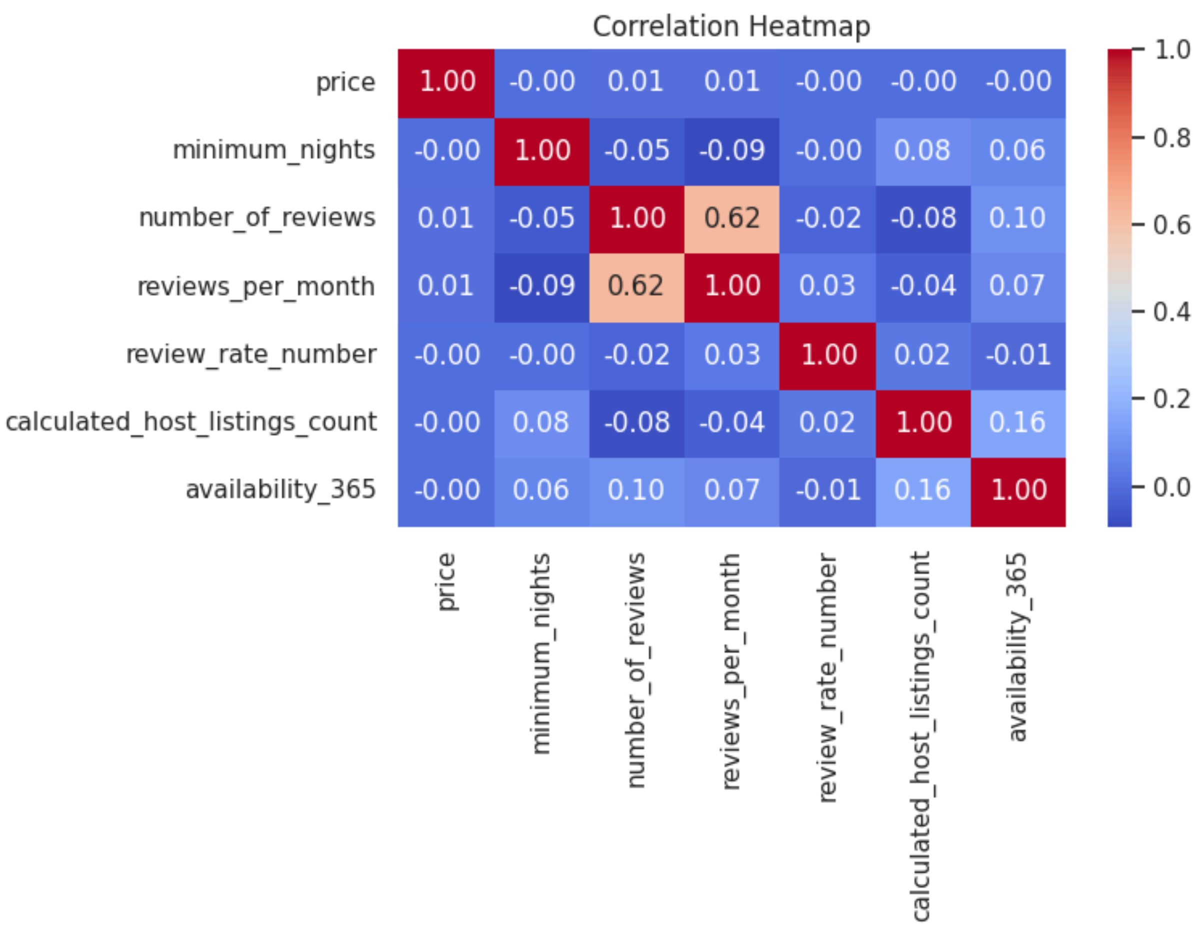
Histogram: Price Distribution - shows how listing prices are distributed.

Histogram: Number of Reviews - shows distribution of review counts.



Scatter Plot: Price vs Reviews per Month - shows relationship between price and review frequency.

Heatmap: Correlation between numerical variables - shows relationships between key numeric columns.



Dashboard Creation

In [ ]:

```
# Step 5: Dashboard Creation - Airbnb Listings

# Install plotly if needed
# !pip install plotly

import plotly.express as px
import plotly.graph_objects as go
```

In [17]:

```
# Clean column names
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

# Clean price column
df['price'] = df['price'].replace(r'[\$,]', '', regex=True).astype(float)

# Group by neighbourhood_group
group_summary = df.groupby('neighbourhood_group')[['price',
                                    'number_of_reviews']].mean().reset_index

# Interactive bar chart for average price
fig_price = px.bar(group_summary,
                   x='neighbourhood_group',
                   y='price',
                   title='Average Price by Neighbourhood Group',
                   labels={'price':'Average Price',
                           'neighbourhood_group':'Neighbourhood Group'},
                   color='price',
                   color_continuous_scale='Blues')
fig_price.show()

# Interactive bar chart for average number of reviews
fig_reviews = px.bar(group_summary,
                     x='neighbourhood_group',
                     y='number_of_reviews',
                     title='Average Number of Reviews by Neighbourhood Group
                     labels={'number_of_reviews':'Avg Reviews',
                             'neighbourhood_group':'Neighbourhood Group'},
                     color='number_of_reviews',
                     color_continuous_scale='Greens')
fig_reviews.show()

# Insight placeholder
print("Insight: Manhattan has the highest average price, while Brooklyn show
```

Insight: Manhattan has the highest average price, while Brooklyn shows highe
r average reviews per listing.

In [16]:
```
# Top 10 hosts by number of listings
top_hosts = df.groupby('host_name')['id'].count().sort_values(ascending=Fals

# Interactive bar chart
fig_top_hosts = px.bar(top_hosts,
                       x='id', y='host_name',
                       title='Top 10 Hosts by Number of Listings',
                       orientation='h',
                       labels={'id':'Number of Listings', 'host_name':'Host'
                       color='id',
                       color_continuous_scale='Oranges')
fig_top_hosts.show()

# Insight placeholder
print("Insight: Michael, David, and "+
"John have the most listings on Airbnb in this dataset,"+
      "dominating the market share among hosts.")
```

Insight: Michael, David, and John have the most listings on Airbnb in this d
ataset,dominating the market share among hosts.

In [14]:

```
# Convert last_review to datetime
df['last_review'] = pd.to_datetime(df['last_review'], errors='coerce')

# Extract month period
df['month'] = df['last_review'].dt.to_period('M')
monthly_reviews = df.groupby('month')['id'].count().reset_index()
monthly_reviews['month'] = monthly_reviews['month'].dt.to_timestamp()

# Interactive line chart
fig_monthly = go.Figure()
fig_monthly.add_trace(go.Scatter(x=monthly_reviews['month'], y=monthly_revie
                                 mode='lines+markers', name='Number of Revie

fig_monthly.update_layout(title='Monthly Reviews Trends',
                          xaxis_title='Month',
                          yaxis_title='Number of Reviews',
                          template='plotly_white')
fig_monthly.show()

# Insight placeholder
```

```
print("Insight: Review activity increases"
+ "during peak tourist months, with noticeable spikes in summer and holiday
```

```
Insight: Review activity increasesduring peak tourist months, with noticeabl
e spikes in summer and holiday periods.
```

In [13]:

```
# Select numerical columns for correlation
num_cols = ['price', 'minimum_nights', 'number_of_reviews',
            'reviews_per_month', 'review_rate_number',
            'calculated_host_listings_count', 'availability_365']

# Compute correlation
corr = df[num_cols].corr()

# Plot correlation heatmap
fig_heatmap = px.imshow(corr,
                        text_auto=True,
                        color_continuous_scale='RdBu_r',
                        title='Correlation Heatmap')
fig_heatmap.show()

# Insight placeholder
print("Insight: Price does not strongly correlate with most other features.
"Number of reviews correlates moderately with reviews_per_month, "+
"while availability slightly correlates with host listings count.")
```

```
Insight: Price does not strongly correlate with most other features. Number
of reviews correlates moderately with reviews_per_month, while availability
slightly correlates with host listings count.
```

Conclusions

Key Insights:

- Manhattan listings have the highest average price; Brooklyn shows higher review activity.
- Top hosts (e.g., Michael, David, John) dominate the number of listings.
- Reviews peak during summer and holiday months.
- Numerical features show limited correlation; reviews per month moderately correlate with number of reviews.

Recommendations:

- Price strategically in high-demand areas.
- New hosts can learn from top hosts’ strategies.
- Prepare for peak booking seasons.

- Monitor reviews and availability trends to adjust offerings.