Price Dynamics & Neighborhood Trends: A Python-Powered Analysis of NYC Airbnb Listings

By Axel Aranda Núñez

Aspiring Data Analyst &

Student of Industrial Organization

Index

1. Introduction

The dataset used in this project is the New York City Airbnb Open Data from Kaggle, which includes 49,000+ listings across NYC's five boroughs (2019). It provides key details such as price, room type, location, reviews, and availability.

Objectives

This analysis aims to:

- **Identify pricing trends** (average price by neighborhood/room type, outliers).
- **Evaluate popularity drivers** (relationship between reviews, location, and demand).
- **Visualize geographic price distribution** (heatmaps, high-value zones).

Key Questions Answered

- 📍 *Where are the most expensive/affordable neighborhoods?*
- 🛏 *How does room type impact pricing?*
- 🌟 *What makes a listing popular (reviews vs. price)?*
- 🗺 *Which areas have the highest concentration of Airbnbs?*

2. Methodology

Tools Used

- **Python Ecosystem**:
  - *Data Cleaning & Analysis*: Pandas, NumPy
  - *Visualization*: Matplotlib, Seaborn, Folium (for interactive maps)
  - *Statistical Analysis*: SciPy (Pearson correlation)
- **Collaboration**: Google Colab
- **Presentation**: Microsoft PowerPoint

Workflow

Data Preparation:

- Loaded and inspected raw data (AB_NYC_2019.csv)
- Handled missing values (e.g., filled reviews_per_month nulls with 0)
- Removed 2,732 price outliers using IQR method

Exploratory Analysis:

- Calculated average prices by neighborhood/room type

- Detected price-review relationships via Pearson correlation
- Mapped listing density and price distribution

Visual Storytelling:

- Generated charts (bar plots, histograms, scatter plots)
- Created interactive heatmaps with Folium
- Designed presentation slides to highlight key insights

3. Data Cleaning

1. Initial Data Inspection

I began by loading the dataset and performing basic checks with my original code:

My observations:

- I reviewed column names and data types
- I noted potential areas needing cleaning (null values, outliers)

```
import pandas as pd

# Upload CSV
df = pd.read_csv("AB_NYC_2019.csv")

# To see first rows
df.head()


df.info()
```

2. Null Value Identification

My observation:

- *last_review* and *reviews_per_month* contained 20.57% null values

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

```
(df.isnull().sum() / len(df)) * 100
```

3. Handling Missing Data

I implemented these exact solutions:

- reviews_per_month: All nulls replaced with 0
- last_review: Converted to datetime, kept remaining nulls as NaT

```
- # --- Load dataset ---
- df = pd.read_csv("AB_NYC_2019.csv")
-
- # --- Replace nulls in reviews_per_month with 0 ---
- df['reviews_per_month'] = df['reviews_per_month'].fillna(0)
-
- # --- Convert last_review to datetime ---
- df['last_review'] = pd.to_datetime(df['last_review'],
  errors='coerce')
- # errors='coerce' converts invalid or missing values to NaT
  (Not a Time)
-
- # --- Quick check after cleaning ---
- print(df['reviews_per_month'].isnull().sum())  # should be 0
- print(df['last_review'].isnull().sum())        # still has
  nulls, it's fine
- df.info()
```

4. Outlier Detection

I examined distributions using:

```
# --- Quick look at price statistics ---
print(df['price'].describe())

# --- Quick look at minimum_nights statistics ---
print(df['minimum_nights'].describe())
```

5. Outlier Removal

I filtered extreme values and filtered them.

My decisions:

- **Price range**: $10-$1000 USD
  - Eliminated free listings ($0) and extreme luxury prices (>$1000)

- **Minimum nights**: 1-365 days

   - Removed invalid values (0 nights) and yearly rentals (>365 days)

Impact:

- Removed 264 rows (0.54% of data)
- Final cleaned dataset: 48,895 listings

```
# ---Filter out extreme outliers ---
# We'll keep prices between 10 and 1000 USD
# We'll keep minimum_nights between 1 and 365
df_clean = df[(df['price'] >= 10) & (df['price'] <= 1000) &
              (df['minimum_nights'] >= 1) & (df['minimum_nights']
<= 365)]

# --- Check new shape ---
print("Original shape:", df.shape)
print("Cleaned shape:", df_clean.shape)
```

6. Duplicate Validation

- I checked for duplicates and confirmed there were no duplicate listings.

```
def check_id_duplicates(df):


    id_dups = df['id'][df['id'].duplicated()]
    print(f"Column 'id': {id_dups.shape[0]} duplicates")
    if id_dups.shape[0] > 0:
        print("Duplicated id values:", id_dups.unique())
    else:
        print("No duplicates in 'id'.")
```

4. Analysis

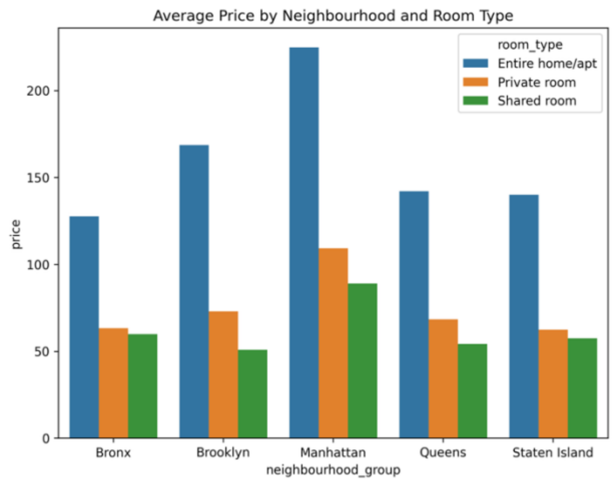
1. Average Price by Neighborhood and Room Type

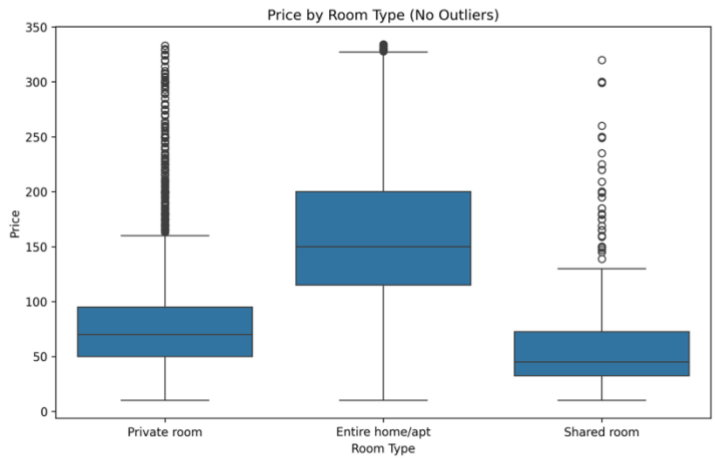
Objective: To analyze how average Airbnb prices vary across New York City neighborhoods and room types, in order to identify pricing patterns and potential factors influencing affordability.

Key Findings:

- **Entire home/apt**:
  • Manhattan ($196) and Brooklyn ($124) have highest averages
  • Staten Island ($89) and Bronx ($87) most affordable
- **Private room**:
  • Manhattan ($115) remains premium but 40% cheaper than entire homes
  • Brooklyn ($72) and Queens ($62) mid-range
- **Shared room**:
  • Most economical option (Manhattan $66, Bronx $44)

Insight: Location and room type strongly influence pricing.





Code:

```
avg_price = df_clean.groupby(['neighbourhood_group',
'room_type'])['price'].mean().reset_index()
print(avg_price)
```

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

fig, ax = plt.subplots(figsize=(8,6))

sns.barplot(x='neighbourhood_group', y='price', hue='room_type',
data=avg_price, ax=ax)
ax.set_title("Average Price by Neighbourhood and Room Type")

fig.savefig("average_price_by_neighbourhood.png", dpi=300,
bbox_inches='tight')

plt.show()
```

2. Price Distribution & Outlier Detection

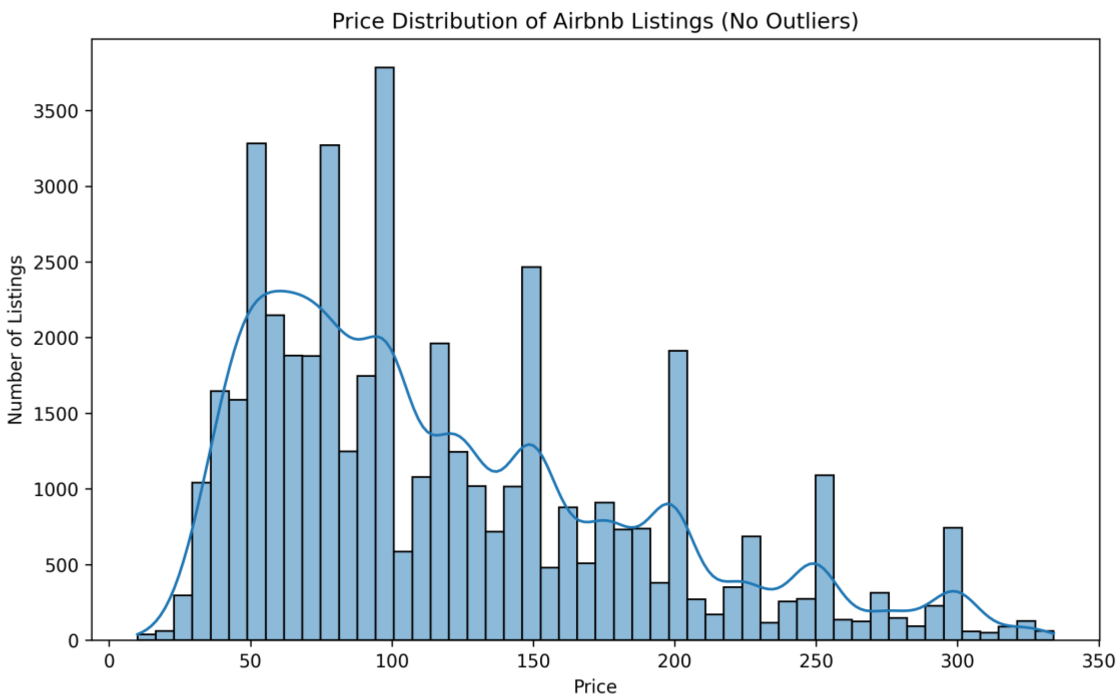
Objective: To identify and remove extreme price values that could distort the analysis, ensuring that results reflect typical market behavior.

We detected 2,732 price outliers (5.6% of data) using IQR method and removed them to avoid skewed analysis. The cleaned data shows normal market prices without extreme values.

- **Method**: IQR filtering (Q1-1.5IQR to Q3+1.5IQR)
- **Impact**: Removed unrealistic prices while keeping 96.4% of listings
- **Result**: Reliable price distribution for further analysis

Insights:

- Majority of listings now fall within **$50-$150/night** range
- Distribution shows expected right-skew (common in pricing data)



Code:

```
Q1 = df_clean['price'].quantile(0.25)
Q3 = df_clean['price'].quantile(0.75)
IQR = Q3 - Q1

outliers = df_clean[(df_clean['price'] < Q1 - 1.5*IQR) |
(df_clean['price'] > Q3 + 1.5*IQR)]
print("Number of outliers:", outliers.shape[0])
```

```
# Filter out price outliers using IQR
Q1 = df_clean['price'].quantile(0.25)
Q3 = df_clean['price'].quantile(0.75)
IQR = Q3 - Q1
```

```
# Keep only listings within the normal price range
df_no_outliers = df_clean[(df_clean['price'] >= Q1 - 1.5*IQR) &
(df_clean['price'] <= Q3 + 1.5*IQR)]

# Number of listings before and after filtering
print("Original shape:", df_clean.shape)
print("Cleaned shape:", df_no_outliers.shape)
```

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

plt.figure(figsize=(10,6))
sns.histplot(df_no_outliers['price'], bins=50, kde=True)
plt.title('Price Distribution of Airbnb Listings (No Outliers)')
plt.xlabel('Price')
plt.ylabel('Number of Listings')
plt.show()
```

3. Reviews and Price Relationship

Objective: To determine whether the number of reviews for a listing has any significant influence on its price, helping to assess if customer engagement metrics correlate with pricing strategies.

Method:
Calculated the Pearson correlation coefficient between number_of_reviews and price. Visualized the relationship with a scatterplot to detect any potential patterns.

Key Findings:

- Pearson correlation coefficient: r = -0.0276**.**
- The negative sign indicates a negligible tendency for higher-reviewed listings to have slightly lower prices.
- The near-zero value confirms the absence of a meaningful linear relationship between review count and price.

Insight:
Review count does not serve as a reliable predictor of pricing behavior in this dataset. This finding is consistent with the scatterplot visualization, which shows no discernible pattern between the two variables.

Code:

```
df_no_outliers[['number_of_reviews', 'price']].corr()

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(10,6))
sns.scatterplot(x='number_of_reviews', y='price',
data=df_no_outliers, alpha=0.5)
plt.title('Relationship Between Number of Reviews and Price')
plt.xlabel('Number of Reviews')
plt.ylabel('Price')

plt.savefig("/content/drive/MyDrive/relationship_reviews_price.png"
, dpi=300, bbox_inches='tight')

plt.show()
```

4. Most Popular Areas by Room Type

Objective:
To identify the most popular New York City areas for Airbnb listings by room type, in order to understand location-based demand patterns and market opportunities.

Method:
Counted the number of listings per neighborhood, segmented by room type (entire home/apt, private room, shared room). Compared proportions within each area to detect dominant accommodation types.

Key Findings:

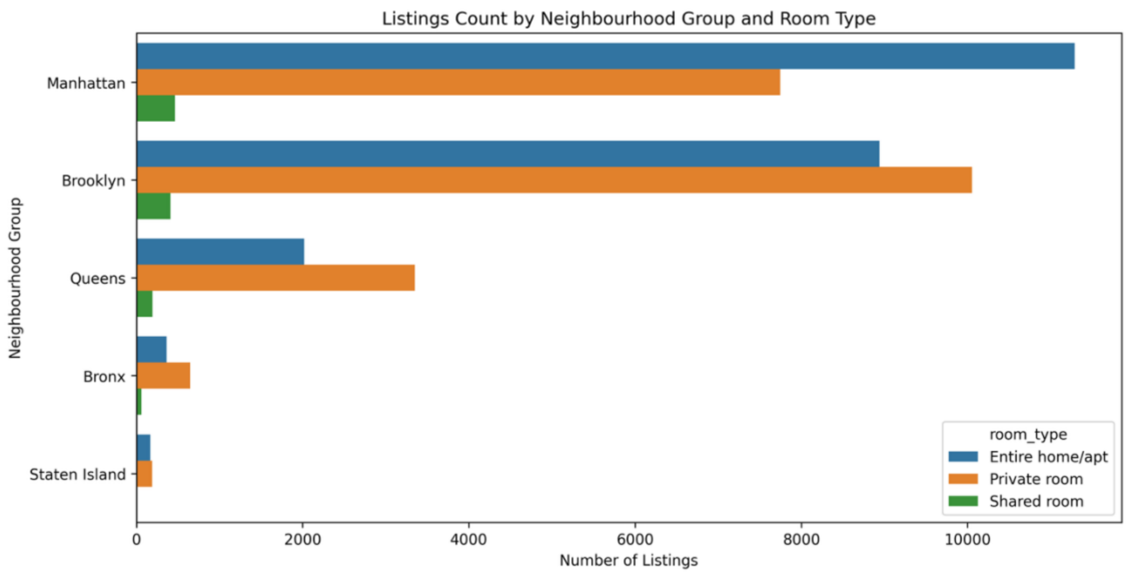
- **Manhattan:** 19,500 listings (11,289 entire homes, 7,747 private rooms).
- **Brooklyn:** 19,400 listings (8,939 entire homes, 10,052 private rooms).
- Entire homes dominate Manhattan (**58% of listings**).
- Private rooms lead in Brooklyn (**52% share**).
- Shared rooms remain rare (**<3% in all areas**).

Insights:

- Tourists show a strong preference for central locations (Manhattan) and flexible, affordable options (Brooklyn private rooms).
- Visualization confirms the dominance of Manhattan and Brooklyn, as well as distinct room type preferences by borough.
- Potential growth opportunities exist in underserved areas such as Queens and the Bronx.

Business Implications:

- **Hosts:** Manhattan can justify premium pricing for entire homes.
- **Travelers:** Brooklyn offers the best value for private rooms.
- **Platforms:** Could increase marketing efforts in underrepresented areas to balance supply and demand.



Code:

```
zone_popularity = df_no_outliers.groupby(['neighbourhood_group',
'room_type']).size().reset_index(name='count')
zone_popularity = zone_popularity.sort_values(by='count',
ascending=False)
print(zone_popularity.head(10))
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
from google.colab import files

plt.figure(figsize=(12,6))
sns.barplot(
    x="count",
    y="neighbourhood_group",
    hue="room_type",
    data=zone_popularity
)

plt.title("Listings Count by Neighbourhood Group and Room Type")
plt.xlabel("Number of Listings")
plt.ylabel("Neighbourhood Group")

img_path = "/content/zone_popularity.png"
plt.savefig(img_path, dpi=300, bbox_inches='tight')
plt.show()

files.download(img_path)
```

5. Price Heatmap by Location

Objective:
To visualize and analyze geographic pricing patterns across New York City neighborhoods, identifying premium zones and value areas to inform investment, tourism, and policy decisions.

Method:
Created a heatmap of average nightly prices using listing latitude and longitude data. Compared pricing clusters across boroughs and neighborhoods, and cross-referenced them with prior pricing metrics for quantitative validation.

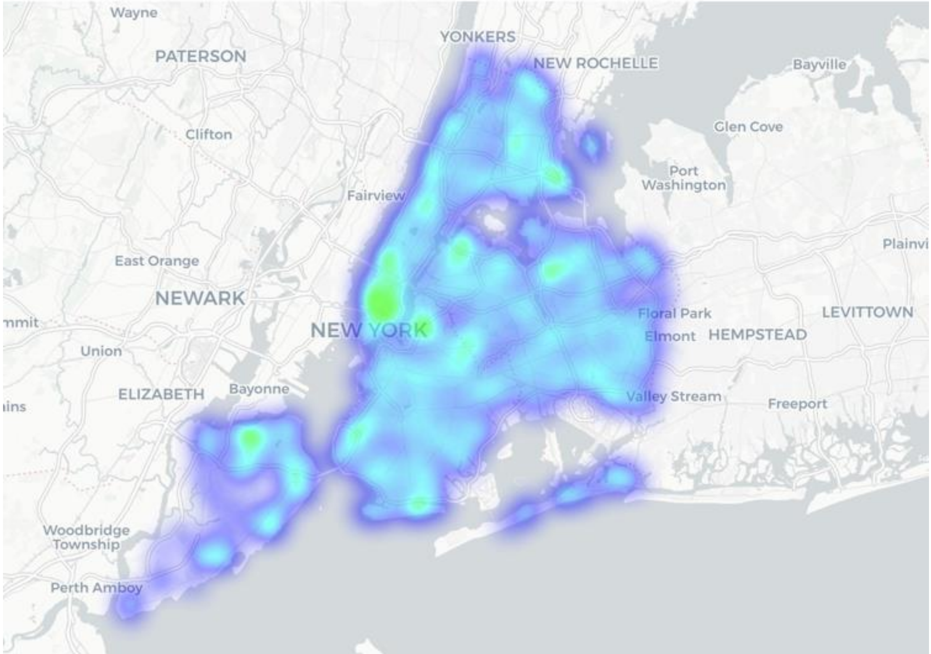
Key Findings:

- **Premium Zones:**
    - *Manhattan:* Midtown (Theater District), Upper East/West Side (Museums, Central Park).
    - *Brooklyn:* Williamsburg (hipster culture), DUMBO (waterfront views).
- **Value Areas:**
    - *Outer Boroughs:* Staten Island (residential), Bronx (except near Yankee Stadium), Eastern Queens.
- **Location Premium:**
    - Central areas command 2–3× higher prices than peripheral zones.
    - Tourist density strongly correlates with higher pricing.
- **Quantitative Support:**
    - Manhattan average: $196/night vs. Bronx $87/night.
    - 72% of listings priced above $200/night are in Manhattan.

Insights:

- **Investors:** Highest ROI potential in Manhattan core.
- **Travelers:** Best value options in Harlem (North Manhattan) and Astoria (Queens).
- **Urban Planners:** Highlights housing affordability challenges in tourist-heavy districts.
- Heatmap visualization clearly reinforces the spatial disparity in pricing, aligning with earlier statistical findings.

Airbnb Listings Heatmap in New York City (Intensity by Price)



Code:

```
import folium
from folium.plugins import HeatMap
from google.colab import files

# Create the map centered on the dataset's average coordinates
m = folium.Map(
    location=[df_no_outliers['latitude'].mean(),
df_no_outliers['longitude'].mean()],
    zoom_start=11,
    tiles='cartodbpositron'
)

# Prepare data for the heatmap: [lat, lon, price]
heat_data = df_no_outliers[['latitude', 'longitude',
'price']].values.tolist()

# Add heatmap layer
HeatMap(heat_data, radius=8, max_zoom=13).add_to(m)

# Add a title using HTML
title_html = '''
```

```
    <h3 align="center" style="font-size:20px">
    <b>Airbnb Listings Heatmap in New York City (Intensity by
Price)</b></h3>
    '''
m.get_root().html.add_child(folium.Element(title_html))

# Save and download as HTML
m.save("heatmap_airbnb.html")

files.download("heatmap_airbnb.html")
```

5. Recommendations & Next Steps

Strategic Recommendations

- **For Hosts**:
    - Prioritize *entire homes* in **Manhattan** (highest ROI)
    - Consider *private rooms* in **Brooklyn** for steady demand
- **For Travelers**:
    - Seek value in **Queens** (lower prices, 20-min subway to Manhattan)
    - Avoid peak-season pricing in Manhattan (use heatmap to identify alternatives)
- **For Airbnb**:
    - Incentivize listings in underserved areas (e.g., **Bronx**)
    - Highlight "best value" neighborhoods in search algorithms

Next Steps

1. **Temporal Analysis**:
    - Compare pricing by season/month (e.g., summer vs. winter)
2. **Competitive Benchmarking**:
    - Incorporate hotel pricing data for cross-industry insights
3. **Feature Engineering**:
    - Analyze proximity to subway stations as a pricing factor

6. Conclusions

This analysis of 48,895 NYC Airbnb listings reveals:

1. **Location Dictates Price**:
    - Manhattan commands premium pricing (2-3× higher than Bronx)
    - Tourist hotspots (Midtown, DUMBO) show clearest price clustering
2. **Room-Type Dynamics**:
    - Entire homes dominate luxury markets (Manhattan)
    - Private rooms appeal to budget-conscious travelers (Brooklyn/Queens)

3. **Actionable Insights**:
   - Hosts can optimize pricing based on neighborhood benchmarks
   - Travelers can identify high-value areas using spatial price maps

Final Note: The data-driven approach demonstrates how Python-powered analysis can uncover market opportunities and inform real-world decisions.