

1. NYC Short -Term Rental Insights

By Alina Barsan

08/06/2023

2. Summary

The objective of this project is to provide insights and recommendations to Pillow Palooza into the short-term rental market in New York City, by analyzing NYC Airbnb listings 2019 dataset.

This dataset contains information such as listing id, neighborhood, burrows(referred to as neighbourhood_group) price, and information about ratings. I chose to focus on neighbourhood_group, price, number_of_ratings, and room_type.

3. Context

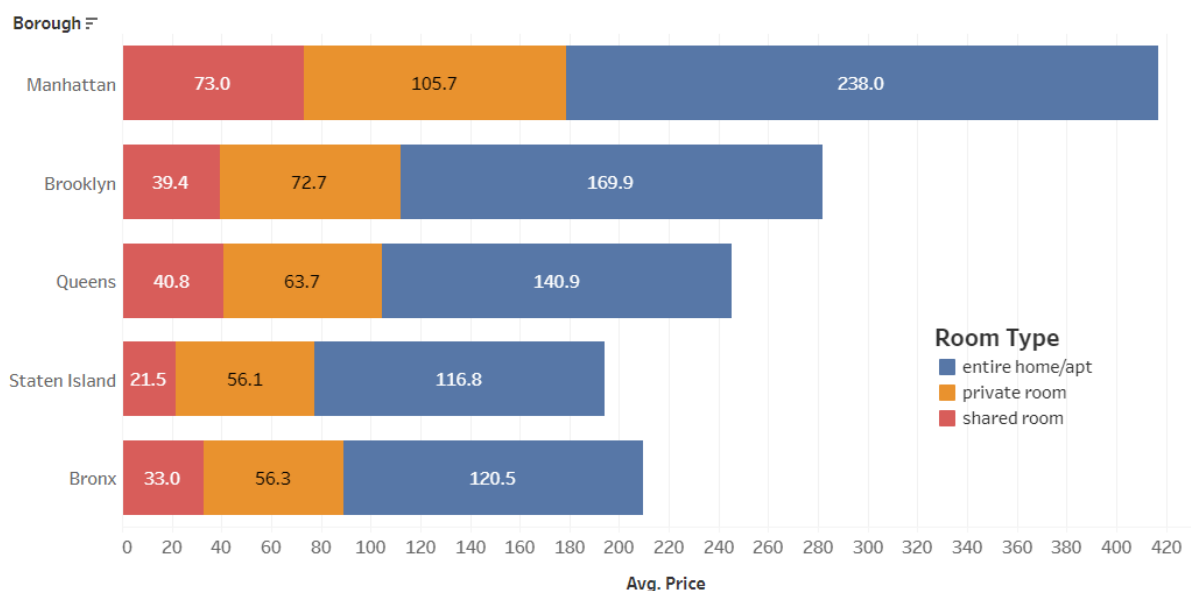
We conducted statistical descriptive analysis on the data to discover patterns and key findings in the short-term rental market in New York City.

We identified the most popular neighborhoods, the average price by neighborhood and property type, the most common rented property type and how it varies in the neighborhoods, and which neighborhood has the highest average price per month.

4. Key Findings

We concluded that the most popular neighborhoods with the highest concentration of short-term rentals in New York City are: Manhattan and Brooklyn, followed by Queens.

Average Price



We determined that the average rental prices are very different across the neighborhoods. The following neighborhoods have an average price less than 100USD:

Staten Island, Bronx and Queens, whereas Manhattan and Brooklyn have an average price higher than 100USD and a maximum price of over 5000 USD. The Bronx and Staten Island do not have many prices above 500 USD.

We discovered that the most popular type of accommodations among guests is renting the entire apartment/home, followed by private rooms and then shared rooms.

Queens has an average price of 92 USD, and the price is between a maximum of 2600 USD and minimum of 10 USD.

Queens - descriptive analytics

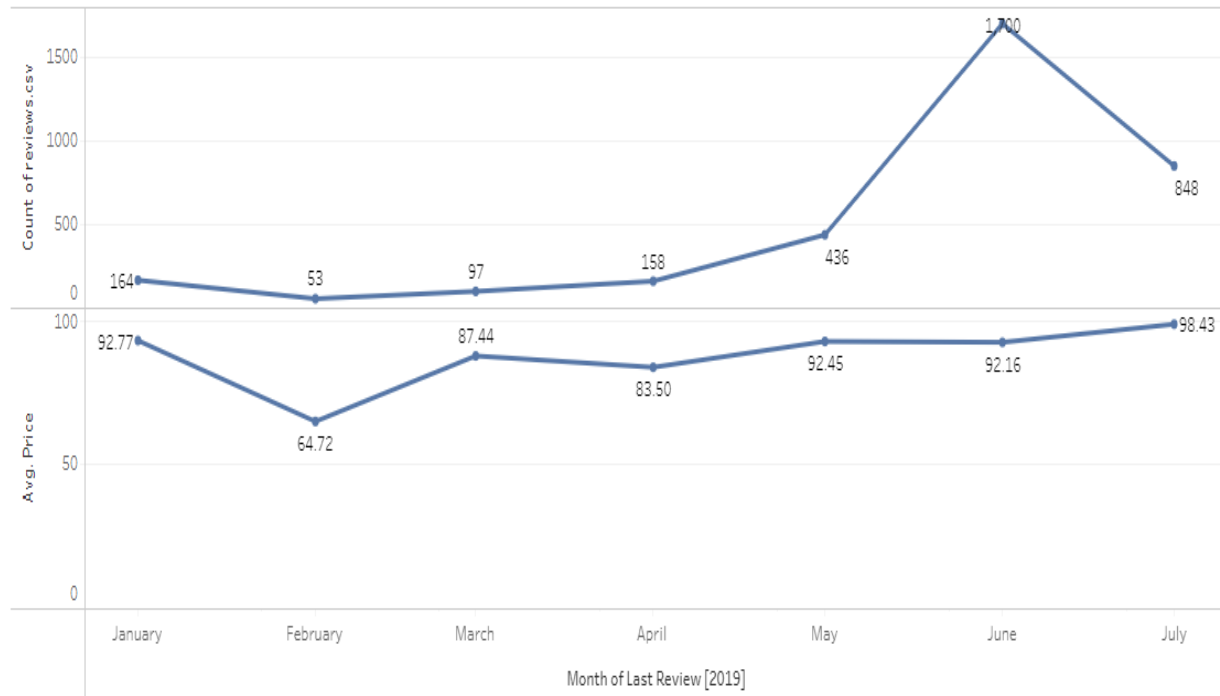
	Room Type		
	entire home/apt	private room	shared room
Avg. Price	141	64	41
Max. Price	2,600	900	120
Min. Price	10	22	15
Count of Calculated Host Listings Count	1,335	2,009	112
% of Number Of Reviews	38.63%	58.13%	3.24%

The counts of reviews seemed to have a considered rise in the month of June, and a lower count in the month of February.

As it can be observed, there is a sharp increase in reviews from May (~500) to June (~1700) with a sharp decrease by July (~800). This can be due to the summer season

and the influx of tourists staying in New York through the summer. Since the dataset ends in July, we cannot see how the trend is for the end of the year.

Reviews Queens



5. Recommendation

Through this exploratory data analysis and visualization, we gained several interesting insights into the Airbnb rental market in NYC. We proceeded with analyzing boroughs and neighborhood listing densities and what areas were more popular than another, their price variations, and room types preferred by guests.

We recommend focusing on the next popular neighborhood, which is Queens, and has the most promising opportunities for Pillow Palooza to focus on and expand its business. The data shows that there is demand, and focusing on Manhattan maybe would not be a great idea, because Airbnb already does very well in that area.

We could contact hosts, from listings in Queens, that have higher reviews and we could offer higher commission fees in exchange for them to promote Pillow Palooza.

Through this analysis, we have a better idea on the key factors that influence the demand of an airbnb listing property. Tourists/customers prefer an entire room which offers them more privacy when touring the city.

These can all be taken into consideration for Pillow Palooza's future business plans.

6. Appendix

Python - codes -

```
import numpy as np
import pandas as pd
import datetime as dt
#importing the data

prices = pd.read_csv("dataairbnb_price.csv")
xls = pd.ExcelFile("dataairbnb_room_type.xlsx")
room_types = xls.parse(0)
reviews = pd.read_csv("dataairbnb_last_review.tsv", sep="\t")

#cleaning price column

prices["price"] = prices["price"].str.replace('dollars', ' ')
prices["price"] = pd.to_numeric(prices["price"])

#Subset prices for listings costing $0, free_listings

free_listings = prices["price"] == 0

#Update prices by removing all free listings from prices

prices = prices.loc[~free_listings]
prices.head()

avg_price = round(prices["price"].mean(), 2)
print(avg_price)
prices.describe()
#Add a new column to the prices DataFrame, price_per_month

prices["price_per_month"] = prices["price"] * 365 / 12

# average_price_per_month
```

```
average_price_per_month = round(prices["price_per_month"].mean(), 2)
difference = round((average_price_per_month - 3100),2)
print(average_price_per_month)
```

```
room_types['room_type'].sample(10)
# Convert the room_type column to lowercase
```

```
room_types["room_type"] = room_types["room_type"].str.lower()
room_types['room_type'].sample(10)
# Update the room_type column to category data type
```

```
room_types["room_type"] = room_types["room_type"].astype("category")
assert room_types["room_type"].dtype == 'category'
# Create the variable room_frequencies
```

```
room_frequencies = room_types["room_type"].value_counts()
print(room_frequencies)
reviews.head(10)
# Change the data type of the last_review column to datetime
```

```
reviews["last_review"] = pd.to_datetime(reviews["last_review"])
reviews['last_review'].dt.year
```

```
# Create first_reviewed, the earliest review date
```

```
first_reviewed = reviews["last_review"].dt.date.min()
print(first_reviewed)
```

```
# Create last_reviewed, the most recent review date
```

```
last_reviewed = reviews["last_review"].dt.date.max()
print(last_reviewed)
```

```
# join dataframes
```

```
rooms_and_prices = prices.merge(room_types, how="outer", on="listing_id")
airbnb_merged = rooms_and_prices.merge(reviews, how="outer", on="listing_id")
```

```
# Drop missing values from airbnb_merged

airbnb_merged.dropna(inplace=True)
airbnb_merged.duplicated().sum()
print(airbnb_merged)

# Extract information from the nbhood_full column and store as a new column, borough

airbnb_merged["borough"] = airbnb_merged["nbhood_full"].str.partition(",")[0]

# Group by borough and calculate summary statistics

boroughs = airbnb_merged.groupby("borough")["price"].agg(["sum", "mean", "median",
"count"])
print(boroughs)

# Round boroughs to 2 decimal places, and sort by mean in descending order

boroughs = boroughs.round(2).sort_values("mean", ascending=False)
print(boroughs)

# Create labels for the price range, label_names

label_names = ["Budget", "Average", "Expensive", "Extravagant"]

# Create the label ranges, ranges

ranges = [0, 69, 175, 350, np.inf]
print(ranges)

# Insert new column, price_range, into DataFrame

airbnb_merged["price_range"] = pd.cut(airbnb_merged["price"], bins=ranges,
labels=label_names)
print(airbnb_merged["price_range"])

# Calculate occurrence frequencies for each label, prices_by_borough

prices_by_borough = airbnb_merged.groupby(["borough",
"price_range"])[["price_range"].count()]
```

```
print(prices_by_borough)
```

```
airbnb_analysis = {'avg_price': avg_price,  
                  'average_price_per_month': average_price_per_month,  
                  'difference': difference,  
                  'room_frequencies': room_frequencies,  
                  'first_reviewed': first_reviewed,  
                  'last_reviewed': last_reviewed,  
                  'prices_by_borough': prices_by_borough}  
print(airbnb_analysis)
```

SQL - codes -

Most common room type in NYC Airbnb listings

```
SELECT COUNT (room_type) AS count,  
       room_type  
FROM room_types  
GROUP BY room_type  
ORDER BY count DESC;
```

Average price of a listing by room type

```
SELECT AVG(price) as avg_price,  
       room_type  
FROM prices p  
JOIN room_types rt  
  ON p.listing_id = rt.listing_id  
GROUP BY room_type;
```

Which borough has the highest average price per month?

```
SELECT AVG(price_per_month) as avg_price_month,  
       borough  
FROM prices  
GROUP BY borough  
ORDER BY avg_price_month DESC;
```

How many listings of each room type are in each borough?

```
SELECT COUNT(p.listing_id) as nr_listing,  
       borough,  
       room_type  
FROM prices p
```

```
JOIN room_types rt
  ON p.listing_id = rt.listing_id
GROUP BY borough, room_type
ORDER BY borough;
```

How many listings in each room type category have a price of over \$500 per night?

```
SELECT COUNT(p.listing_id) as nr_listings,
       room_type
FROM prices p
JOIN room_types rt
  ON p.listing_id = rt.listing_id
WHERE p.price > 500
GROUP BY room_type;
```

What is the distribution of listing prices by neighborhood?

```
SELECT MAX(price) as max_price,
       MIN(price) as min_price,
       AVG(price) as avg_price,
       borough
FROM prices
GROUP BY borough;
```

What is the estimated amount of revenue generated by hosts in each borough?

```
with ct_1 as (
SELECT SUM(price) as revenue,
       borough,
       booked_days_365
FROM prices p
JOIN reviews r
  ON p.listing_id = r.listing_id
GROUP BY borough, booked_days_365),
```

```
ct_2 as(
SELECT (revenue * booked_days_365) as total_revenue,
       borough
FROM ct_1
GROUP BY borough, ct_1.revenue, ct_1.booked_days_365)
```

```
SELECT SUM(total_revenue) as total,
       borough
```



```
FROM ct_2  
GROUP BY borough;
```

What is the average number of reviews per month for listings in each neighborhood?

```
SELECT AVG(price_per_month) as avg_price_month,  
       neighbourhood,  
       room_type  
FROM prices p  
JOIN room_types rt  
  ON p.listing_id = rt.listing_id  
GROUP BY neighbourhood, room_type  
ORDER BY avg_price_month DESC;
```

How many listings have no reviews?

```
SELECT COUNT(rt.listing_id)  
FROM room_types rt  
left join reviews r  
  ON rt.listing_id = r.listing_id  
WHERE reviews_per_month is NULL;
```

How do the estimated book days correlate with the price of an Airbnb listing in New York City?

```
SELECT CORR(price, booked_days_365)  
FROM prices p  
JOIN reviews r  
  ON p.listing_id = r.listing_id
```

Tableau link

https://public.tableau.com/shared/FHK88Y9PY?:display_count=n&:origin=viz_share_link