Storytelling Case Study: Airbnb, NYC-PPT1

Submitted By:

Khushi Vora

Manish Kumar Pandit

Dhruv Gaur

Background

- Airbnb has seen a major decline in revenue during covid time.
- Now that the restrictions have started lifting and people have started to travel more, Airbnb wants to make sure that it is fully prepared for
- The different leaders at Airbnb want to understand some important insights based on various attributes in the dataset so as to increase the revenue.

Objectives

1. • Understanding key insights from the pre covid data
2. • Post covid business Analysis and Growth opportunities
3. • Identify customer preferences and patterns.

1.Data Overview and Fixing columns

```
#Importing Libraries
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Reading the data
nyc = pd.read_csv('AB_NYC_2019.csv')
nyc.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_revie
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	[illegible]
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

- Imported and Read the data set
- Checked the data types and found that last_review should be date type. Hence, converted the same to date.

```
nyc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 19 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  
 16  availability_365_categories     48895 non-null  object 
 17  minimum_night_categories        48895 non-null  object 
 18  number_of_reviews_categories    48895 non-null  object 
dtypes: float64(3), int64(7), object(9)
memory usage: 7.1+ MB
```

2) Categorization of columns

```python
def availability_365_cat(n):
    if n <= 1:
        return 'very Low'
    elif n <= 100:
        return 'Low'
    elif n <= 200 :
        return 'Medium'
    elif (n <= 300):
        return 'High'
    else:
        return 'very High'
```

```python
def minimum_night_cat(n):
    if n <= 1:
        return 'very Low'
    elif n <= 3:
        return 'Low'
    elif n <= 5 :
        return 'Medium'
    elif (n <= 7):
        return 'High'
    else:
        return 'very High'
```

```python
def number_of_reviews_cat(n):
    if n <= 1:
        return 'very Low'
    elif n <= 5:
        return 'Low'
    elif n <= 10 :
        return 'Medium'
    elif (n <= 30):
        return 'High'
    else:
        return 'very High'
```

We have done the categorization of few features so that we can better understand the relationships and better communicate our findings.

3) Data types

```
# Categorical nominal
categorical_columns = nyc.columns[[1,3,4,5,8,16,17,18]]
categorical_columns
```

```
Index(['name', 'host_name', 'neighbourhood_group', 'neighbourhood',
       'room_type', 'availability_365_categories', 'minimum_night_categories',
       'number_of_reviews_categories'],
      dtype='object')
```

4.2 Numerical

```
numerical_columns = nyc.columns[[0,2,9,10,11,13,14,15]]
numerical_columns
```

```
Index(['id', 'host_id', 'price', 'minimum_nights', 'number_of_reviews',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365'],
      dtype='object')
```

4.3 Coordinates and date

```
cdate = nyc.columns[[6,12]]
nyc[cdate]
```

	latitude	last_review
0	40.64749	2018-10-19
1	40.75362	2019-05-21
2	40.80902	NaT
3	40.68514	2019-05-07
4	40.79851	2018-11-19
...	...	...

- We have 3 data types: Categorical, Numerical and date type.

4) Missing Values

```
# Percentage of missing values
round((nyc.isnull().sum()/len(nyc))*100,2)
```

```
id                                 0.00
name                               0.03
host_id                            0.00
host_name                          0.04
neighbourhood_group                0.00
neighbourhood                      0.00
latitude                           0.00
longitude                          0.00
room_type                          0.00
price                              0.00
minimum_nights                     0.00
number_of_reviews                  0.00
last_review                       20.56
reviews_per_month                 20.56
calculated_host_listings_count     0.00
availability_365                   0.00
availability_365_categories        0.00
minimum_night_categories           0.00
number_of_reviews_categories       0.00
dtype: float64
```

- Two columns (last_review , reviews_per_month) has around 20.56% missing values. name and host_name has 0.3% and 0.4 % missing values

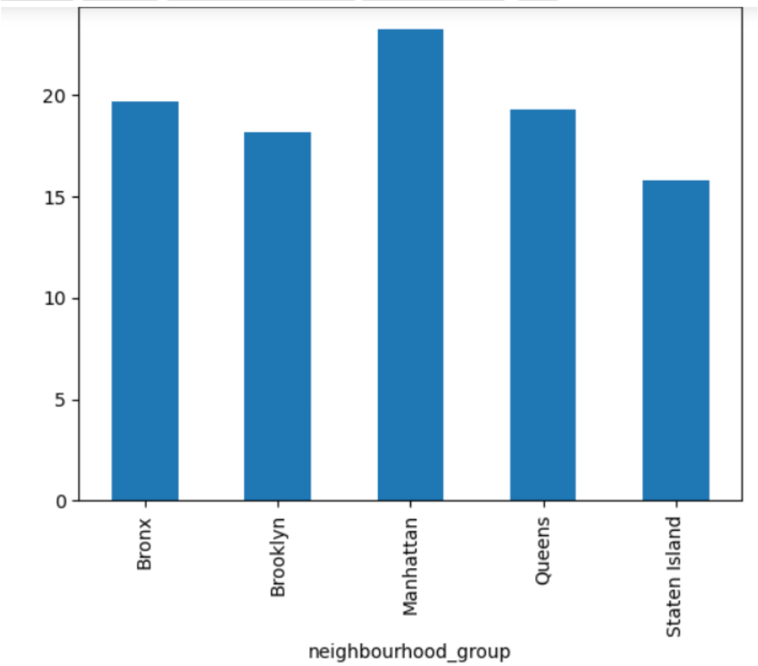
- We need to see if the values are, MCAR: It stands for Missing completely at random.

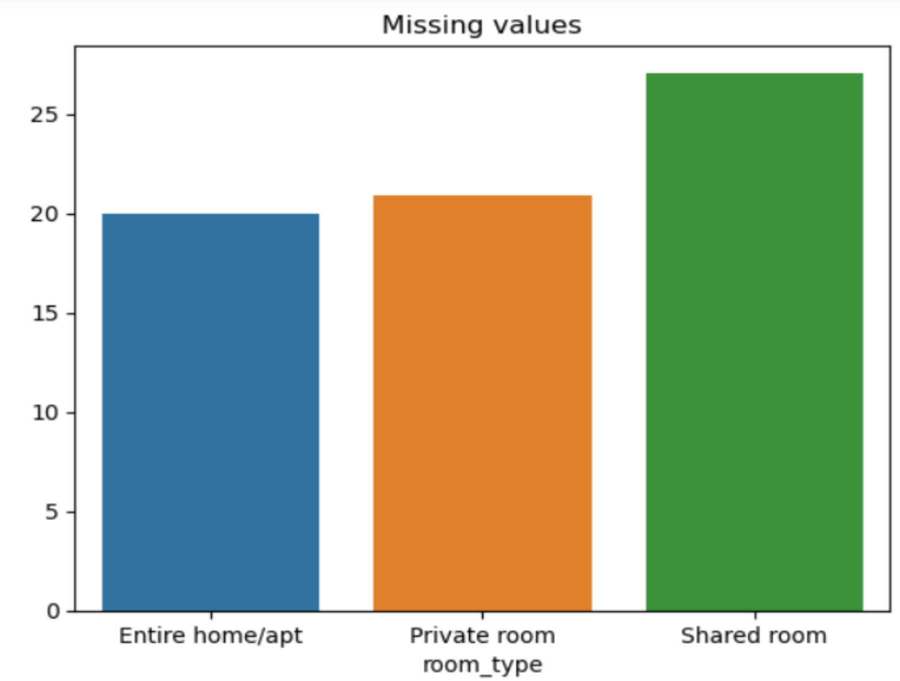
- If the analysis is primarily for storytelling and no predictive model is being created, imputing missing values may not be necessary. In such cases, the missing data itself can tell an important story, such as:

1)Why are higher-priced listings less reviews? 2)Why are shared rooms getting fewer reviews?

-Imputing values in this scenario might obscure these insights. Instead, you could focus on explaining the reasons behind the missing data and what it indicates about customer behavior. Highlighting the missing data can provide valuable context for decision-making rather than trying to fill it in.

Missing Value Analysis- “Last_Review”






```
Mean and Median of prices with last_review feature missing
Mean   =  192.9190210903303
Median =  120.0

Mean and Median of prices with last_review feature not missing
Mean   =  142.317946605566
Median =  101.0
```

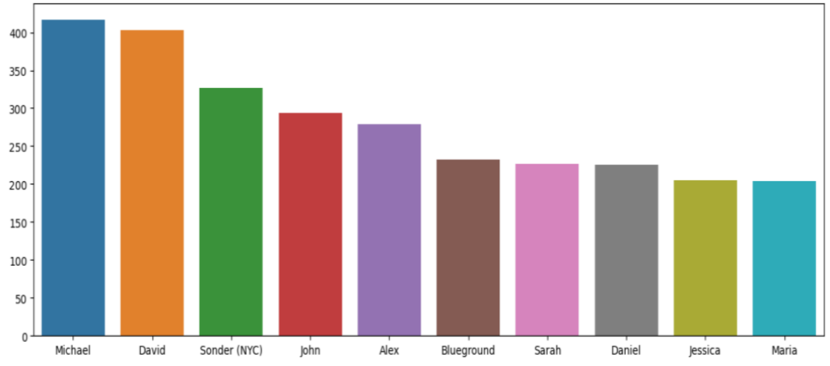
- Higher prices are linked to missing last_review values, indicating that high-priced listings are less likely to receive reviews.
- Shared rooms tend to have fewer reviews, which contributes to the missing last_review data for these room types.
- As prices increase, the likelihood of receiving reviews decreases, possibly due to fewer bookings or higher customer expectations.
- The missing values in last_review are not random but influenced by factors like price and room type.
- This suggests the missing data is not MAR (Missing at Random), where missingness depends on observable features.

```
: ((nyc1.groupby('neighbourhood_group').neighbourhood_group.count()/nyc.groupby('neighbourhood_group').neighbourhood_group.count())
: 19.240898461107257
```

- Each neighbourhood_group has about 19 % missing values in 'last_review' feature.
- Shared room' has the highest missing value percentage (27 %) for 'last_review' feature while to other room types has only about 20 %

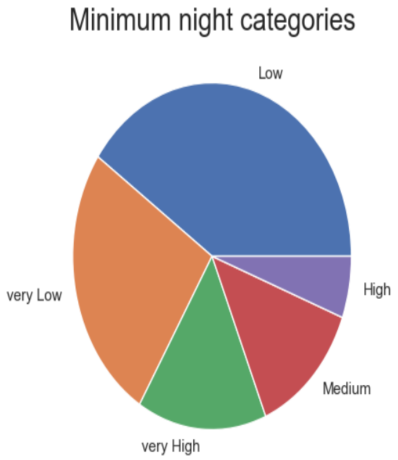
Univariate Analysis

```
# Top 10 host's
plt.figure(figsize=(15,5))
sns.barplot(x = nyc.host_name.value_counts().index[:10] , y = nyc.host_name.value_counts().values[:10])
plt.show()
```

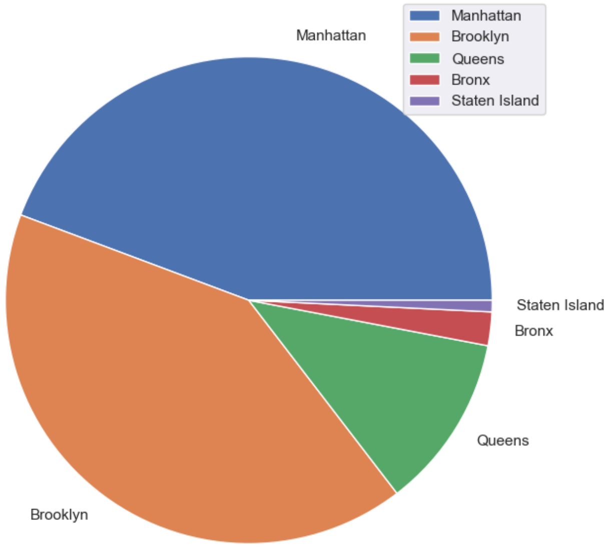


Hosts

```
plt.figure(figsize=(7,5))
plt.title('Minimum night categories', fontdict={'fontsize': 20})
plt.pie(x = nyc.minimum_night_categories.value_counts(),labels=nyc.minimum_night_categories.value_counts().index)
plt.show()
```



Minimum Night Categories

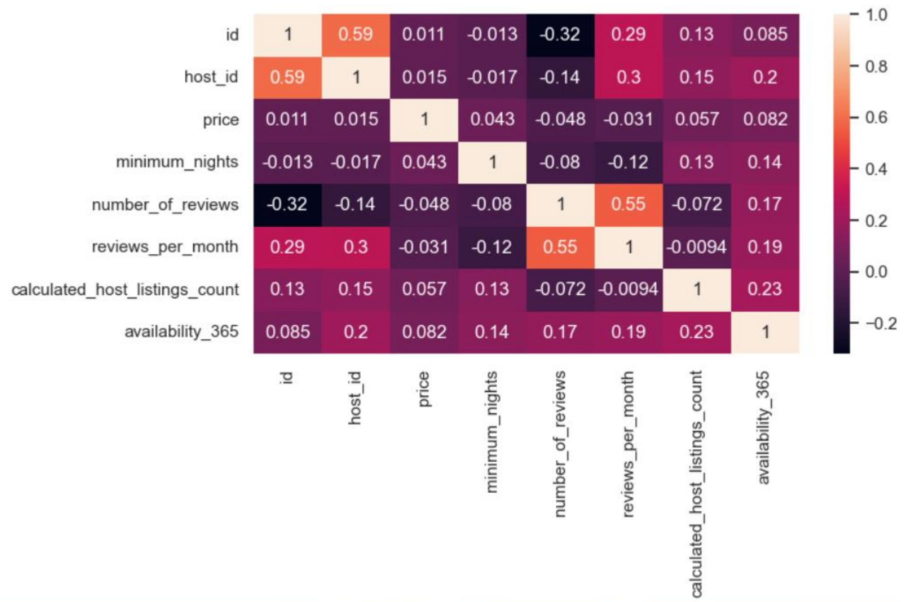


Neighbourhood Groups

- Michael is the top host
- Minimum Nights offered by hosts for Properties are either very low or low.
- Manhattan and Brooklyn has highest listing properties

Bivariate Analysis

```python
sns.heatmap(data = nyc[numerical_columns].corr(),annot=True)
plt.show()
```

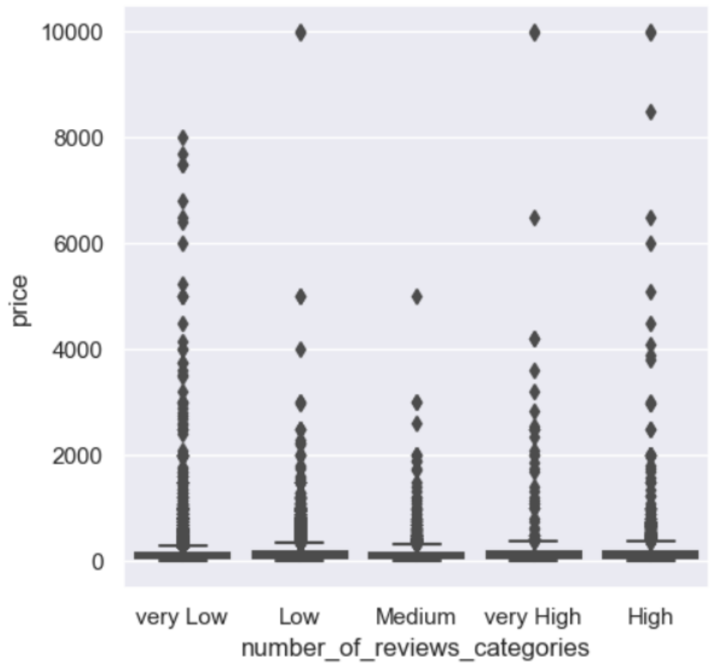



```python
corr_matrix
```

	id	host_id	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
id	1.000000	0.588290	0.010619	0.013224	0.319760	0.291828	0.133272	0.085468
host_id	0.588290	1.000000	0.015309	0.017364	0.140106	0.296417	0.154950	0.203492
price	0.010619	0.015309	1.000000	0.042799	0.047954	0.030608	0.057472	0.081829
minimum_nights	0.013224	0.017364	0.042799	1.000000	0.080116	0.121702	0.127960	0.144303
number_of_reviews	0.319760	0.140106	0.047954	0.080116	1.000000	0.549868	0.072376	0.172028
reviews_per_month	0.291828	0.296417	0.030608	0.121702	0.549868	1.000000	0.009421	0.185791
lated_host_listings_count	0.133272	0.154950	0.057472	0.127960	0.072376	0.009421	1.000000	0.225701
availability_365	0.085468	0.203492	0.081829	0.144303	0.172028	0.185791	0.225701	1.000000

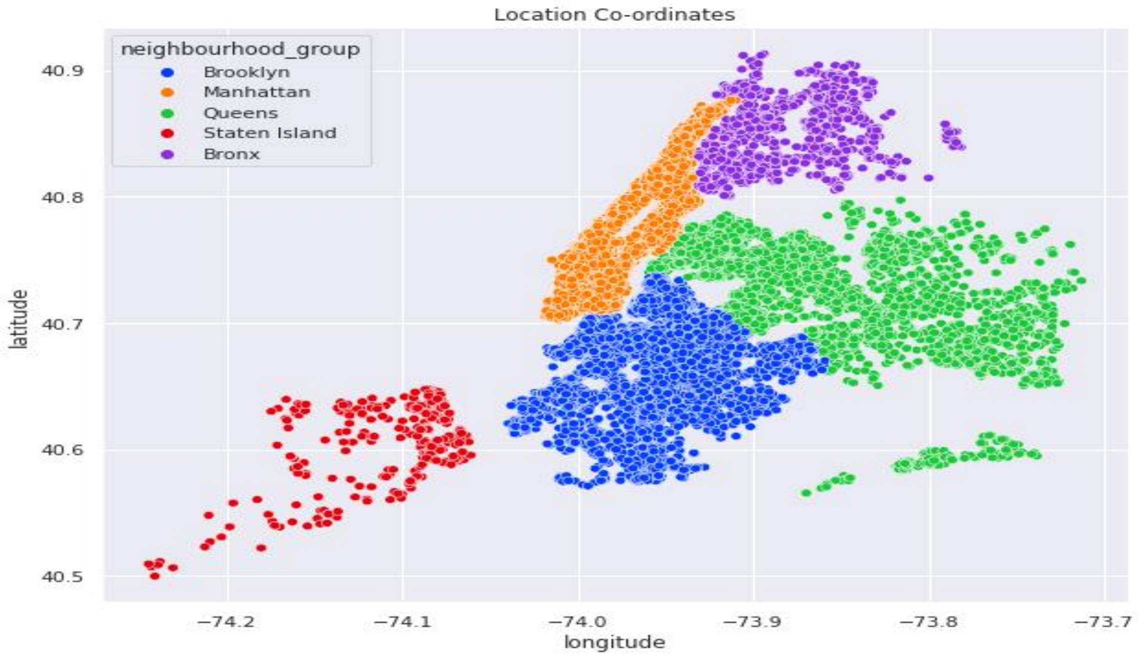
```python
plt.figure(figsize=(5,5))
sns.boxplot(x = nyc.number_of_reviews_categories , y = nyc.price)
```

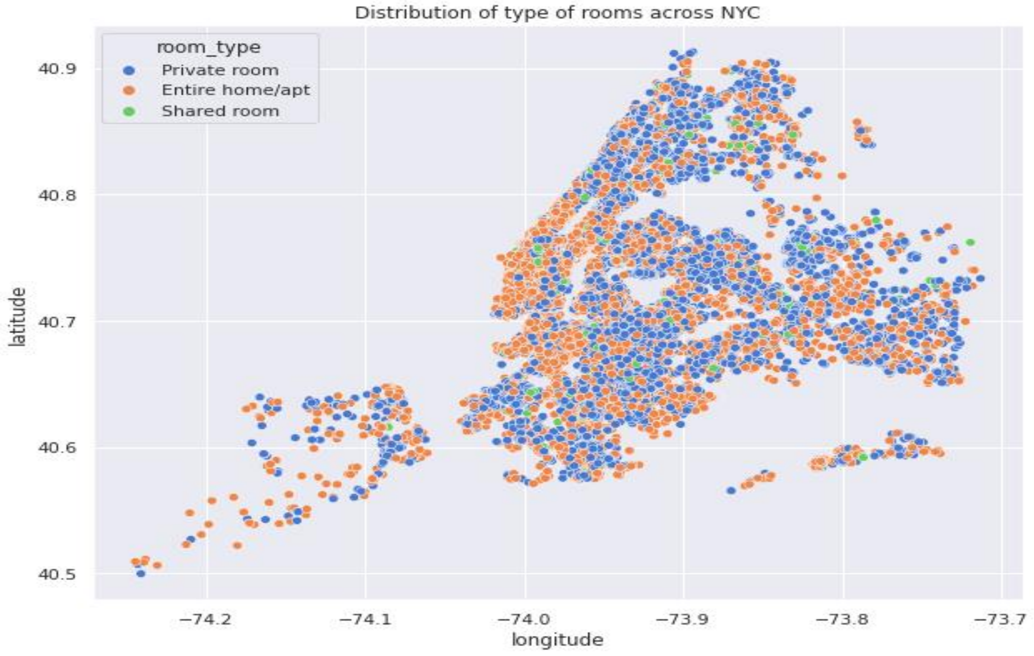
<Axes: xlabel='number_of_reviews_categories', ylabel='price'>



- The total price for 'Low' or 'very Low' number_of_reviews_categories are high.

Distribution of Neighbourhood Group and Types of Rooms in NYC





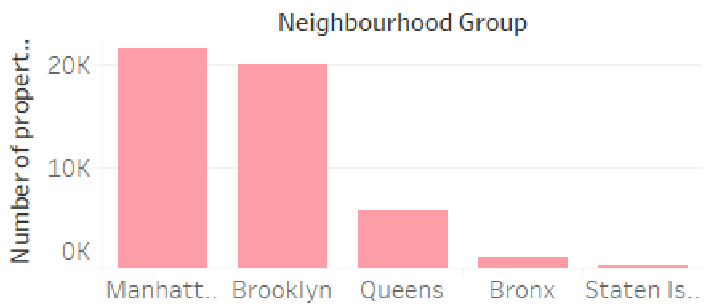
- From the scatterplots of latitude vs. longitude, we can infer that there are **very few** **shared rooms** throughout NYC compared to private rooms and entire homes/apartments.

- **95% of Airbnb listings** are either **private rooms** or **entire homes/apartments**, with only a small number of guests opting for shared rooms. Additionally, guests generally prefer these room types when booking on Airbnb, as our previous analysis indicated.

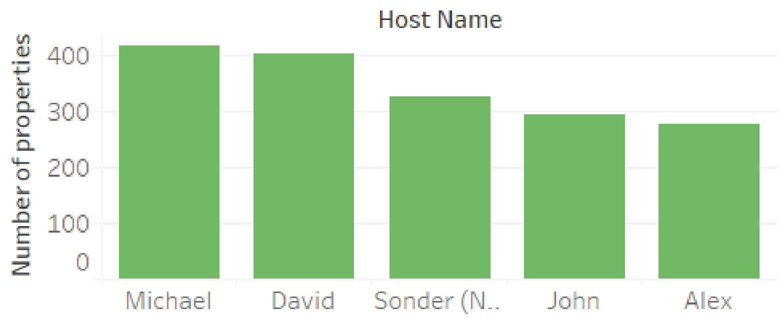
Tableau Visualiation-Dashboards

- 85% of listings are **Manhattan and Brooklyn** Neighbourhood Groups
- The low number of listings in **Staten island** but high prices indicates an untapped market.
- **Manhattan** and **Brooklyn** has the highest number of reviews for room types with **Entire home/apt** ranging to nearly 200000+, followed by **Private room** .
- Sonder has the highest number of properties in Manhattan
- Maximum Number of reviews is provided for either entire home/apt or Private rooms in Manhattan and Brooklyn

Max Properties listed Neighbourhood Group



Host having maximum properties

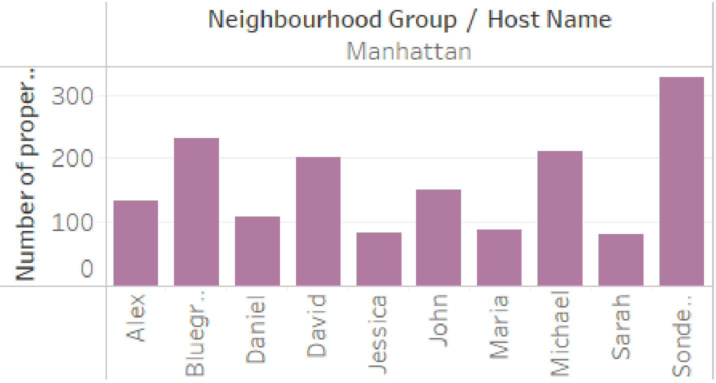


Most Preffered Room Type in Neighbourhood Groups

Room Type	Bronx	Brooklyn	Manhattan	Queens	Staten Island
Entire home/apt	11,627	2,67,128	2,35,147	60,644	5,857
Private room	16,312	2,13,653	2,09,150	93,561	5,670
Shared room	432	5,793	10,272	2,745	14

Number Of Reviews
14 2,67,128

Host having max properties in Manhattan



Average Price for Neighbourhood Groups

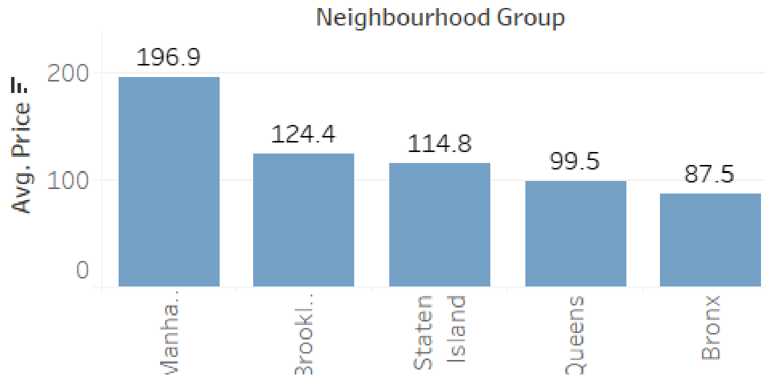
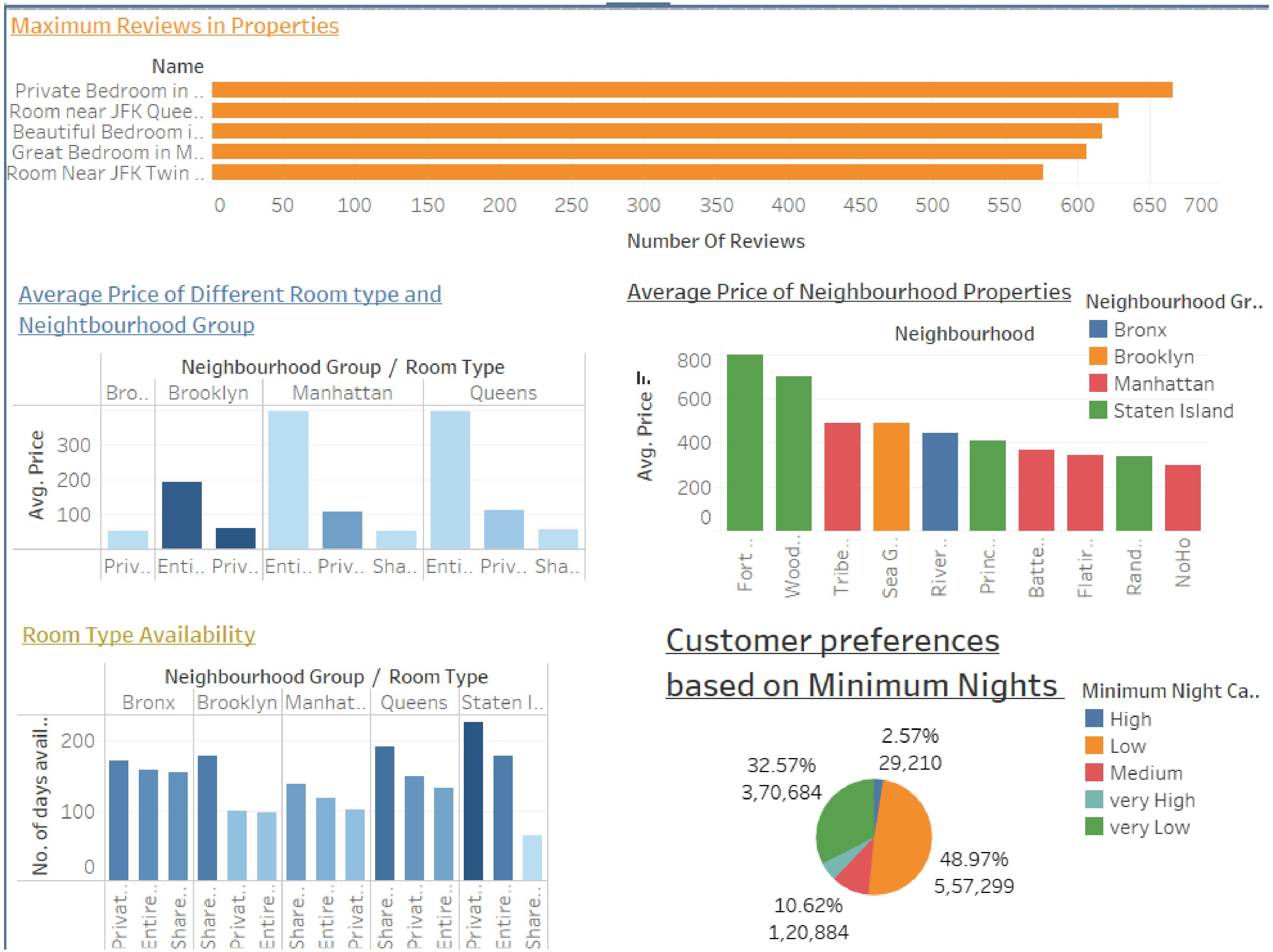


Tableau Visualisations

- **High Price listings** have lower number of reviews and minimum nights is low/very low provided by hosts.
- Availability is low in Brooklyn and Manhattan, making it customer preferrable
- **Private Rooms** in Staten Island are most available
- Airbnb could launch **targeted marketing campaigns** to promote the benefits of private rooms in Staten , showcasing the scenic beauty of the island.



Recommendations and Conclusion:

High Missing values in “last_Review “ column:

- Take customer feedback before the check out so that the facilities can be improved in case of customer dissatisfaction.

Optimize Pricing in High-Demand Areas:

•Implement dynamic pricing in Brooklyn and Manhattan.
•Use promotions during off-peak times and adjust rates based on demand

Host from Manhattan has the highest contribution towards adding the revenue:

Promote Staten Island’s Unique Selling Points:

Highlight Staten Island’s attractions, such as scenic views and cultural sites, in marketing materials.

APPENDIX-DATA DICTIONARY

Note: The price column contains the price/night.

Column	Description
id	listing ID
name	name of the listing
host_id	host ID
host_name	name of the host
neighbourhood_group	location
neighbourhood	area
latitude	latitude coordinates
longitude	longitude coordinates
room_type	listing space type
price	
minimum_nights	amount of nights minimum
number_of_reviews	number of reviews
last_review	latest review
reviews_per_month	number of reviews per month
calculated_host_listings_count	amount of listing per host
availability_365	number of days when listing is available for booking

Dataset Description

APPENDIX-DATA METHODOLOGY

- Understanding the business problem
- Reading the dataset in Python
- Categorisation of features for easy analysis
- Data Wrangling:
- Checking the Duplicates
- Verifying the data types: Numerical, categorical, date and time
- Missing values analysis
- Univariate analysis
- Bivariate analysis
- Using processed data to visualize further in Tableau.

- Please find attached the document for the methodology: