Storytelling Case Study: Airbnb, NYC

1. Analysis in Jupyter Notebook

- The necessary libraries were imported

```
# Import the necessary libraries
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

- Data was read from the CSV file

```
# Data conversion and Understanding
airbnb_data = pd.read_csv("AB_NYC_2019.csv")
airbnb_data.head(5)
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_
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	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

- The dataset has 48895 rows and 16 columns

```
# Check the rows and columns of the dataset
airbnb_data.shape
```

```
(48895, 16)
```

- All the datatypes of the columns were checked

```
# Check the Columns and datatypes
airbnb_data.info()
```

```
Data columns (total 16 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
```

- The dataset was described to find the min, max and percentiles of each numeric attribute

```
# Describe the dataset
airbnb_data.describe()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_lis
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	4
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	

- It was found that columns like name, host_name, last_review and reviews_per_month had null values. The latter 2 had more than 10000 null values.

```python
# Calculating the missing values in the dataset
airbnb_data.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
```

- The unique values for Room_type and Neighbourhood_groups were found

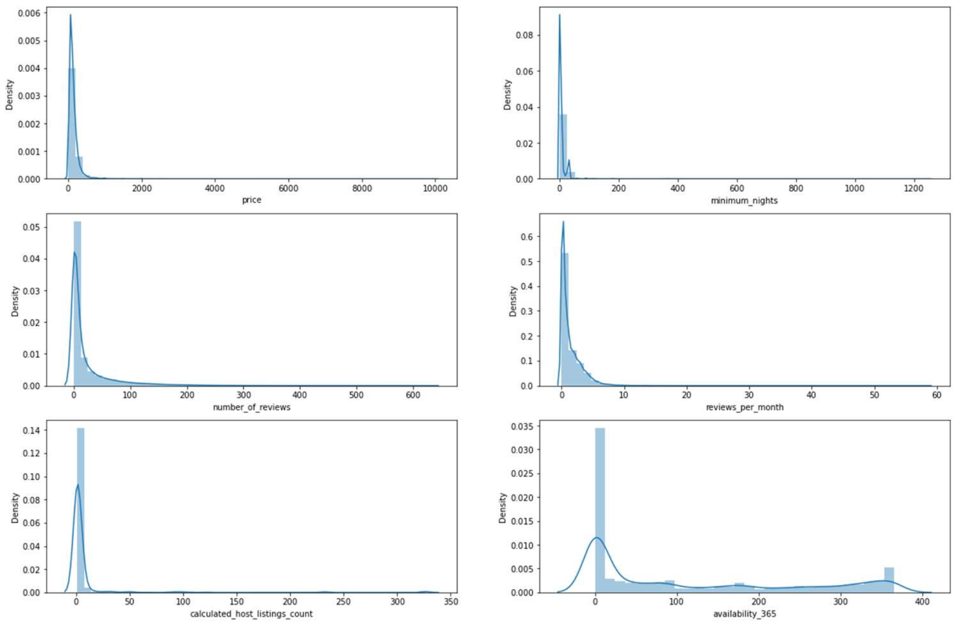
```python
# Now to check the unique values of Room type
airbnb.room_type.unique()
```

```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

```python
# Now to check the unique values of Neighbourhood Group
airbnb.neighbourhood_group.unique()
```

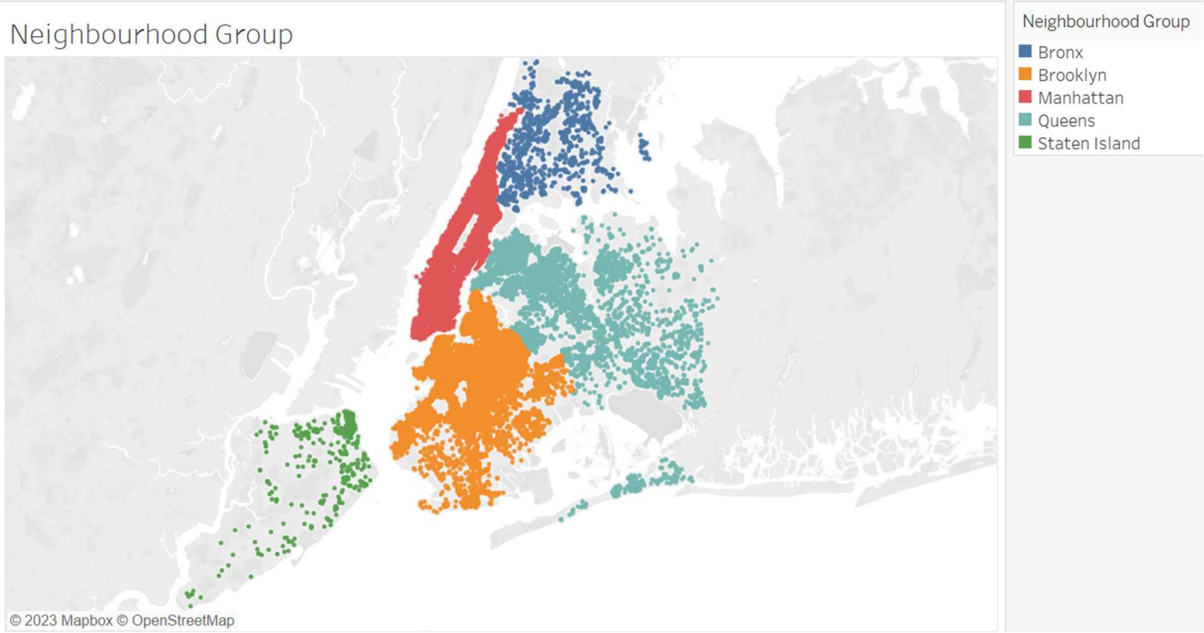
```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

- The density of all numeric variables was visualized

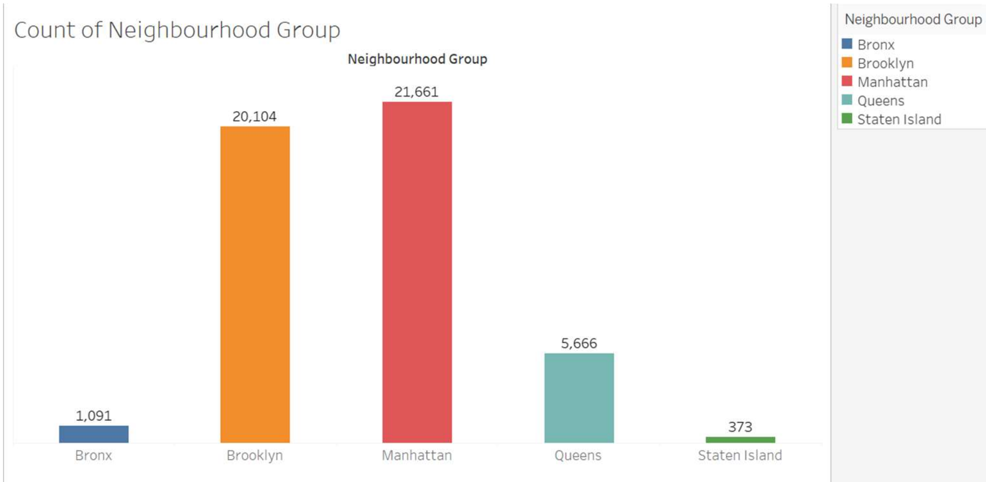


2. Analysis in Tableau

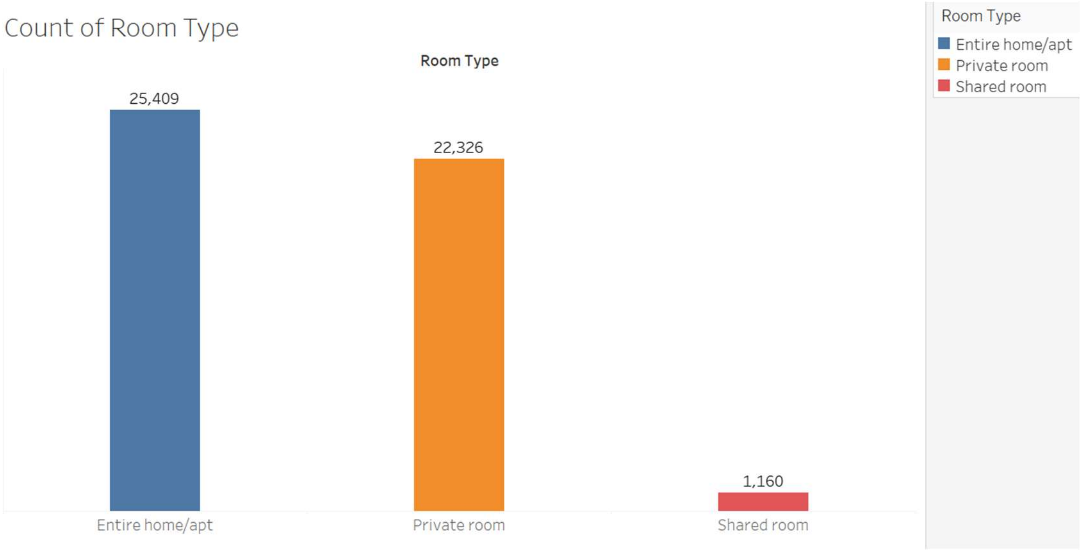
- The bookings were spread across the neighbourhood groups. Manhattan, Brooklyn and Queens had most customer bookings



- The count of bookings made in each location was visualized

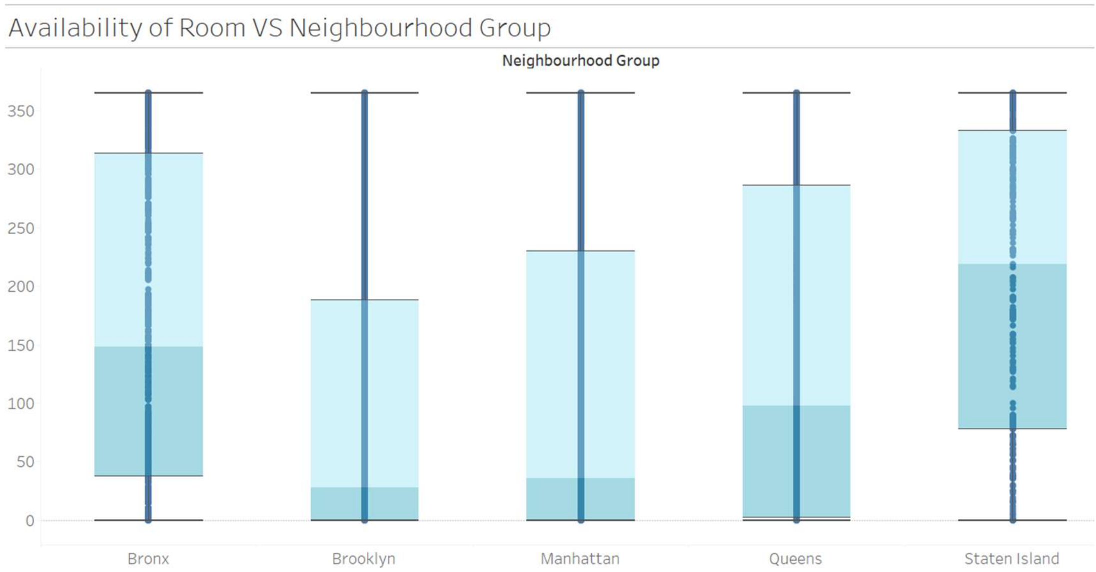


- Most customers preferred an entire home or private rooms. This could mean that most of them travelled as a group like family, couple or friends. Shared rooms were not preferred much.



- When looking at room availability with respect to the room type, shared rooms were almost always available compared to home/apartments and private rooms. This may be due to lack of shared rooms being booked at all.

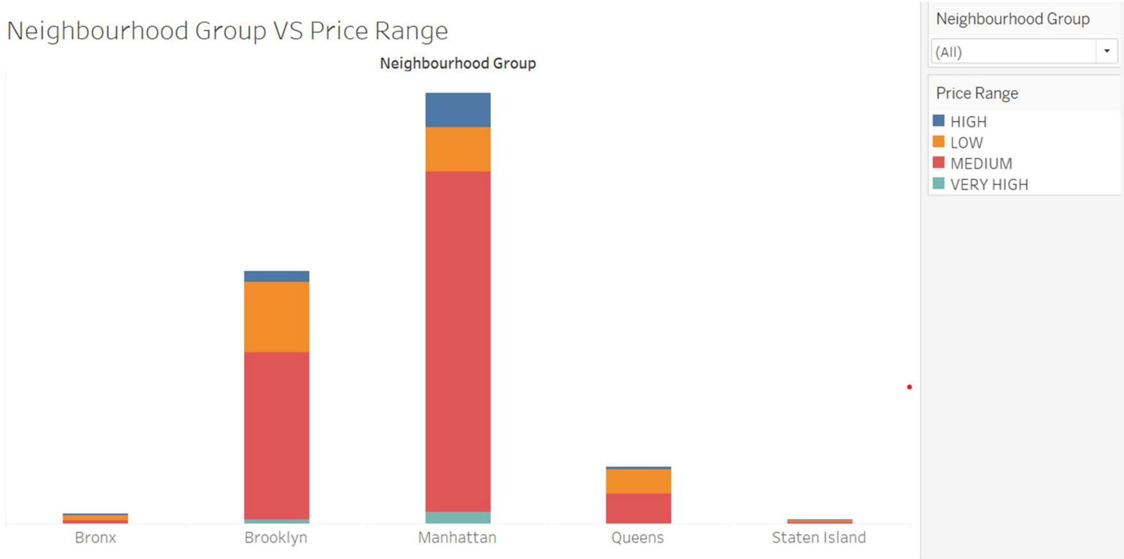
- With respect to the Neighbourhood groups, room availability was lesser in areas like Manhattan and Brooklyn due to high bookings in these areas. Bronx and Staten Island mostly have rooms available.



- A new field called price range was created to know how the bookings were distributed based on the pricing. The most preferred were medium priced places, followed by low priced. Bookings for high priced rooms were less and were mostly seen in Manhattan and Brooklyn.

Price Range

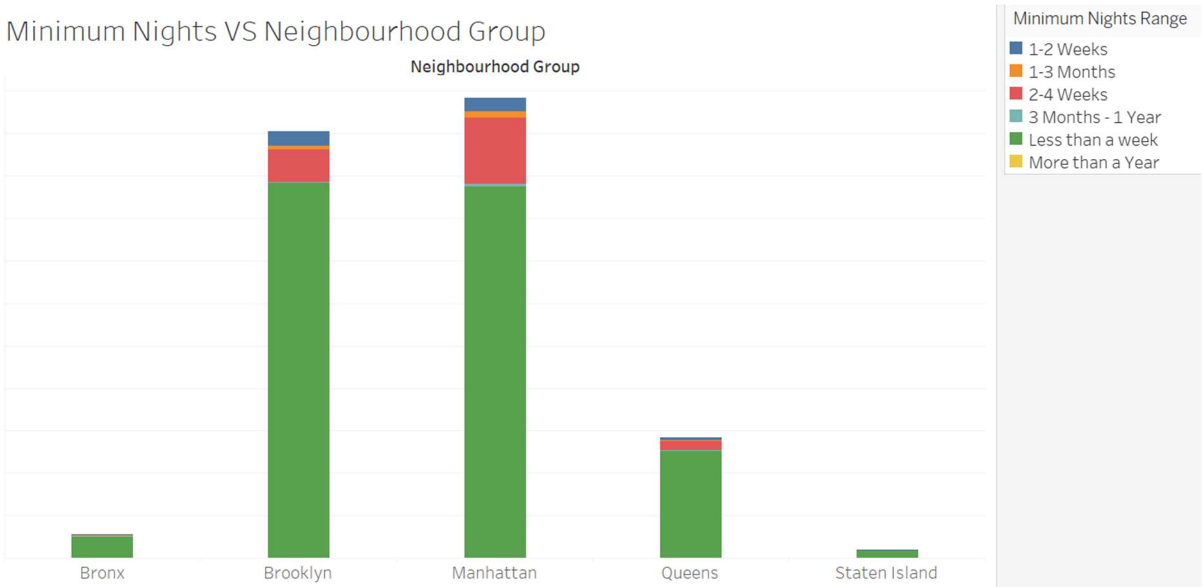
```
IF [Price] < 100 THEN "LOW"
ELSEIF [Price] >= 100 AND [Price] < 1000 THEN "MEDIUM"
ELSEIF [Price] >= 1000 AND [Price] < 5000 THEN "HIGH"
ELSE "VERY HIGH"
END
```



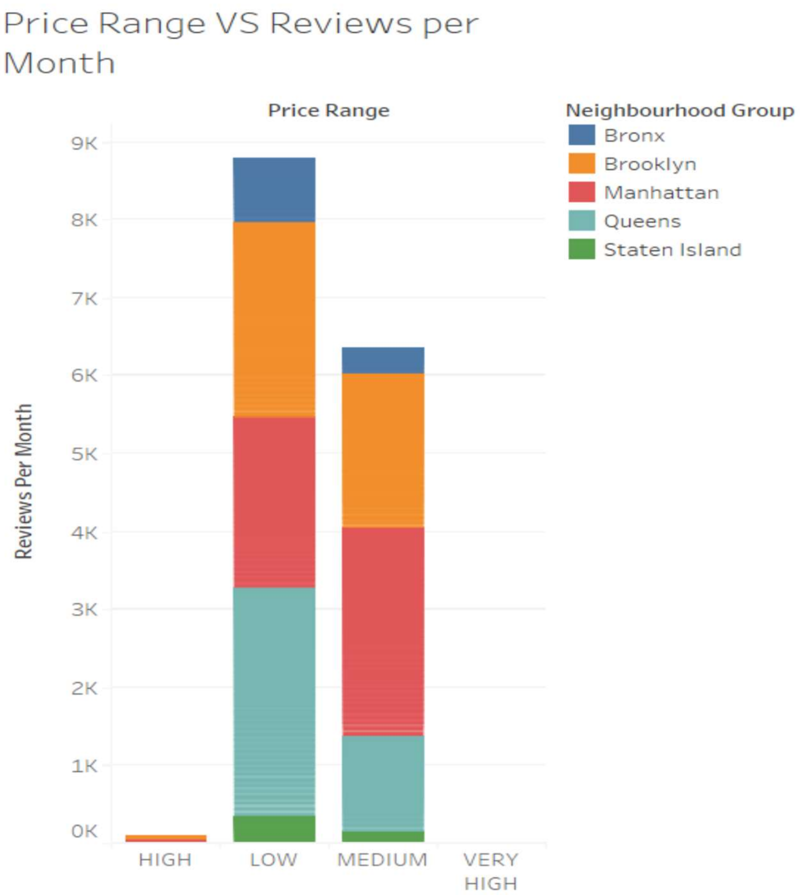
- A new field called Minimum night range was created to know how many nights are the customers spending in their respectively booked places. Most bookings were for less than a week across all the neighbourhood groups. Brooklyn, Manhattan and Queens have had bookings for more than a 2 weeks duration.

ıimum Nights Range

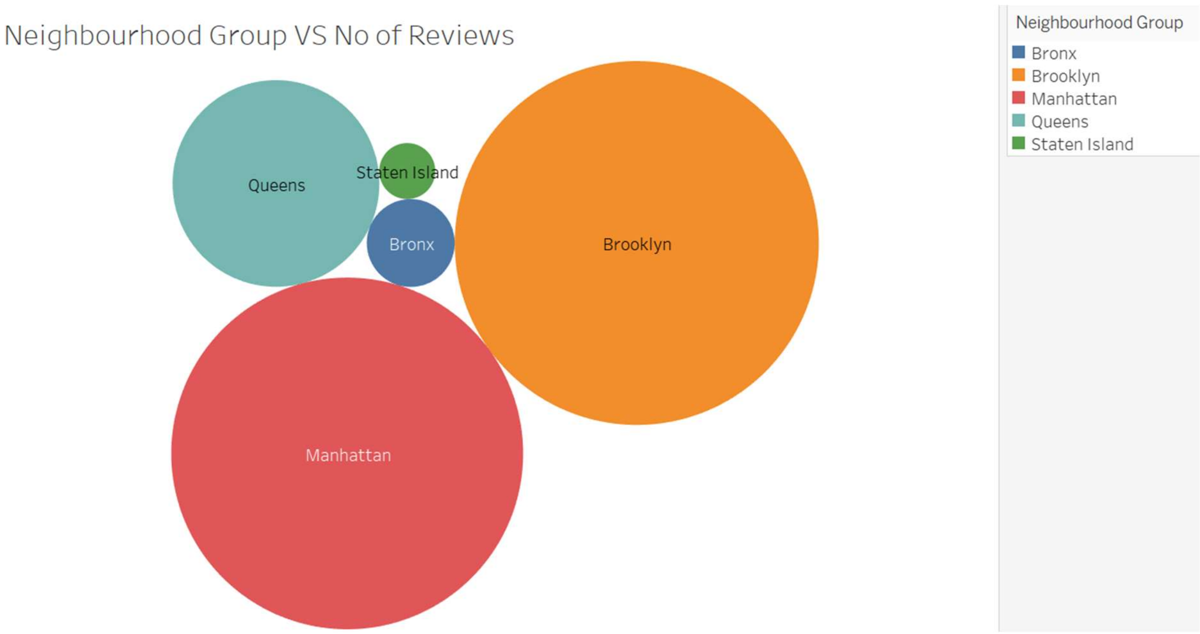
```
IF [Minimum Nights] < 8 THEN "Less than a week"
ELSEIF [Minimum Nights] >=8 AND [Minimum Nights] < 15 THEN "1-2 Weeks"
ELSEIF [Minimum Nights] >=15 AND [Minimum Nights] < 31 THEN "2-4 Weeks"
ELSEIF [Minimum Nights] >=31 AND [Minimum Nights] < 91 THEN "1-3 Months"
ELSEIF [Minimum Nights] >=91 AND [Minimum Nights] < 365 THEN "3 Months - 1 Year"
ELSE "More than a Year"
END
```



- Comparing price range and number of reviews, Low price range has received more reviews.



- The greatest number of reviews were received for Manhattan and Brooklyn but this data cannot be fully relied upon as 'last_review' column has more than 10000 null values.



Conclusion

- The data is not in a good state as a few columns have a large number of null values. Columns like Room types, Neighbourhood groups etc are high imbalanced.
- Most customers prefer locations around Manhattan and Brooklyn which are medium priced. So, room availability in such places should be higher.
- The price rates should be maintained as customers do not prefer high priced locations.
- Customers should be encouraged to provide more reviews to know what can be bettered.
- Customers usually stay for less than a week duration. So, after they check out, it should be immediately made available. Good services should be provided to customers, especially who are staying for long time duration.
- Services and rooms in Manhattan and Brooklyn should be the finest as they are the most popular locations.