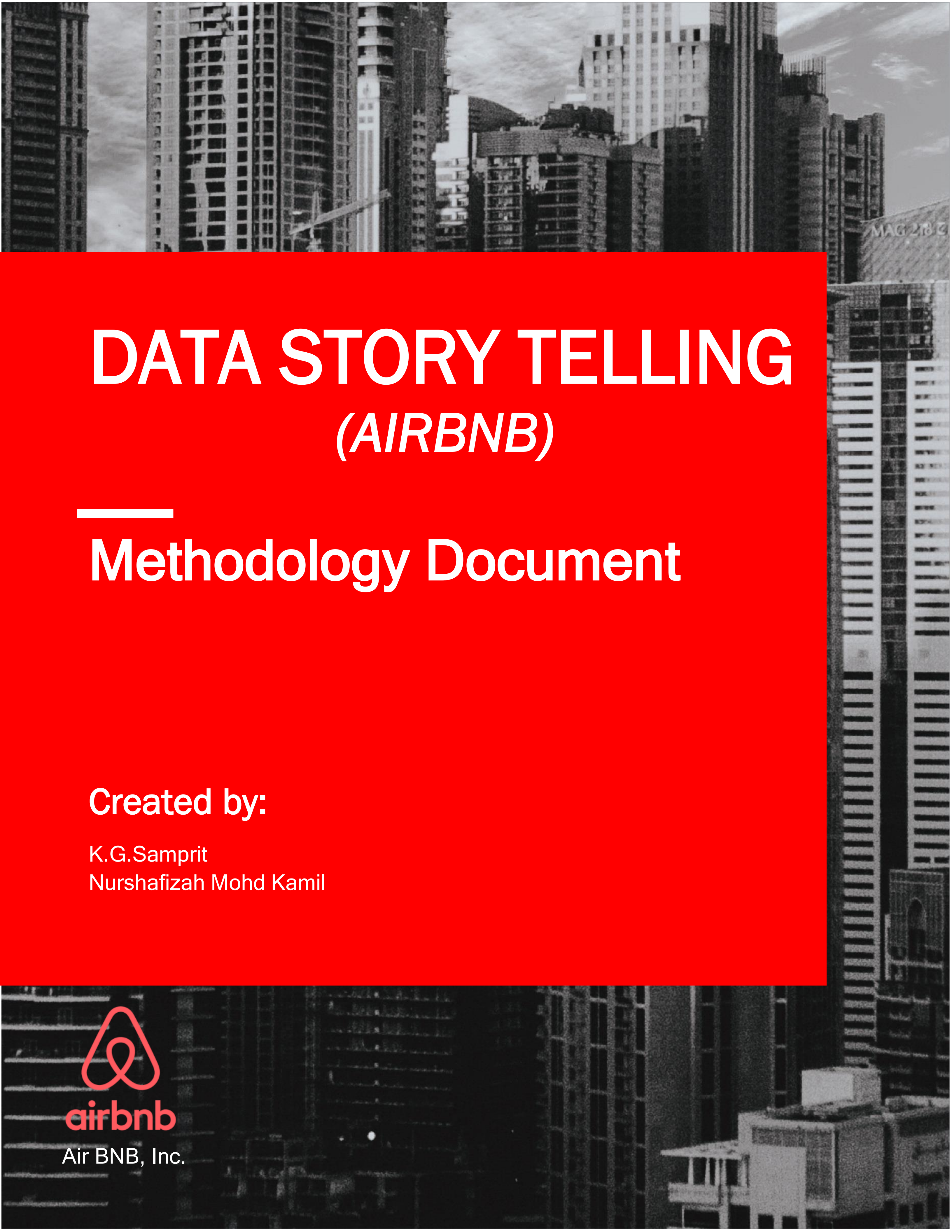


DATA STORY TELLING
(AIRBNB)

Methodology Document

Created by:

K.G.Samprit
Nurshafizah Mohd Kamil

airbnb
Air BNB, Inc.

DATA STORY TELLING: AIRBNB (METHODOLOGY DOCUMENTS)

PROBLEM STATEMENTS:

For the past few months, Airbnb has seen a major decline in revenue. 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 this change.

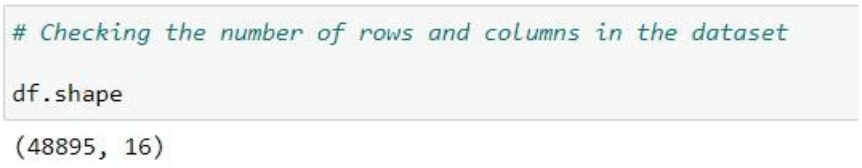
OBJECTIVE:

The different leaders at Airbnb want to understand some important insights based on various attributes in the dataset so as to increase the revenue.

DATA ANALYSIS/ DATA WRANGLING:

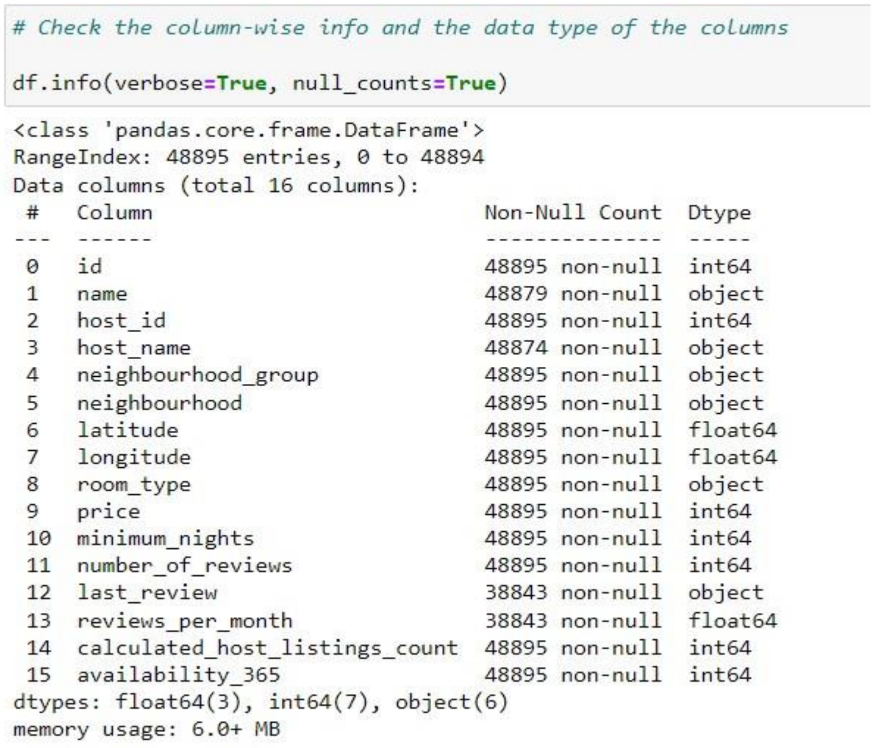
A. Analysing the attributes

The process of analysing the data had been carried out in Jupyter Notebook.



```
# Checking the number of rows and columns in the dataset

df.shape
```

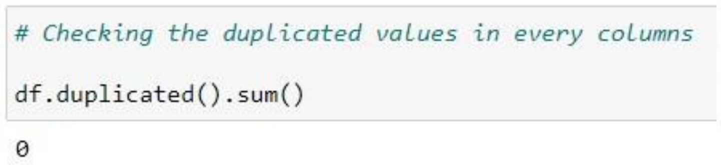
```
(48895, 16)
```



```
# Check the column-wise info and the data type of the columns

df.info(verbose=True, null_counts=True)
```

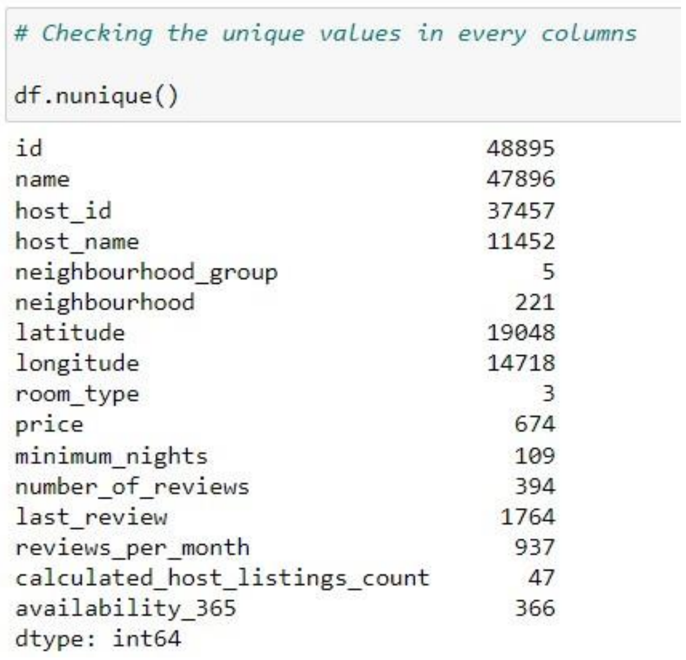
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   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  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

The dataset consists of 48895 rows and 16 columns which contains various attribute with regards with the Airbnb listings such as host id, name, price, room type and more.
There are no duplicated values in the columns.



```
# Checking the duplicated values in every columns

df.duplicated().sum()
```

```
0
```

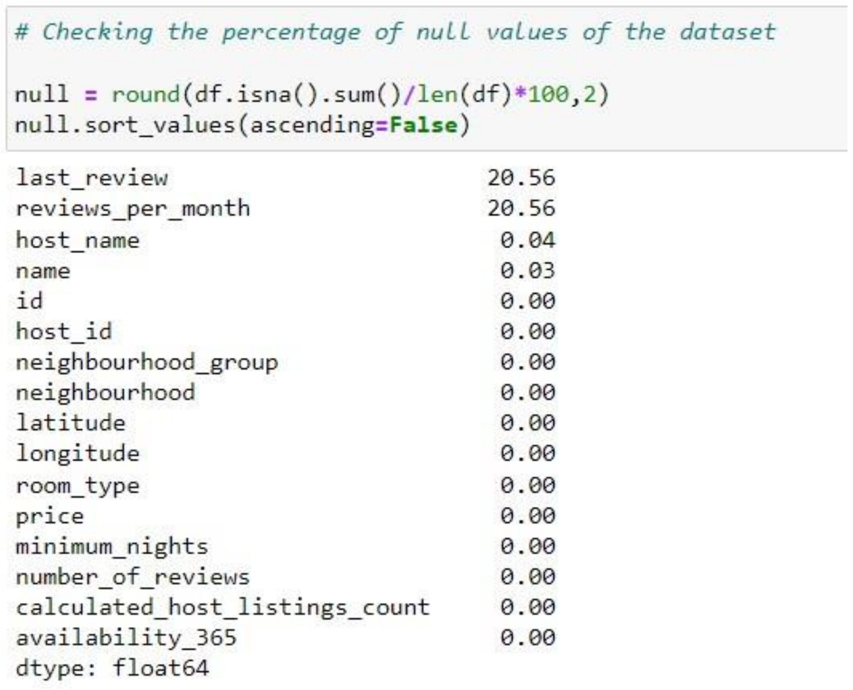


```
# Checking the unique values in every columns

df.nunique()
```

```
id                                48895
name                              47896
host_id                           37457
host_name                         11452
neighbourhood_group                   5
neighbourhood                       221
latitude                          19048
longitude                         14718
room_type                             3
price                               674
minimum_nights                      109
number_of_reviews                   394
last_review                        1764
reviews_per_month                   937
calculated_host_listings_count       47
availability_365                    366
dtype: int64
```

From the above, we can see on how many unique values available in the dataset. For example, there are 3 room type, 5 neighbourhood group in the datasets, out of the 5 neighbourhood group, there are total of 221 neighbourhood in the datasets. Also, there are total of 674 unique pricing in the dataset.

B. Handling the Missing Values



```
# Checking the percentage of null values of the dataset

null = round(df.isna().sum()/len(df)*100,2)
null.sort_values(ascending=False)
```

```
last_review                       20.56
reviews_per_month                 20.56
host_name                          0.04
name                               0.03
id                                 0.00
host_id                            0.00
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
calculated_host_listings_count     0.00
availability_365                   0.00
dtype: float64
```

- Last review column and the reviews per month columns consists of 20.56% of null values for both of the columns.
- 0.04% of null values in host_name and 0.03% null values exists in name column.

Steps:

i. There are totals of 4 columns with null values, which are last review, review per month, name and host name columns.

ii. Based from the above data, we can see that there 20.56% of null values in last review column and reviews per month column, 0.04% of null values in host_name and 0.03% null values exists in name column.

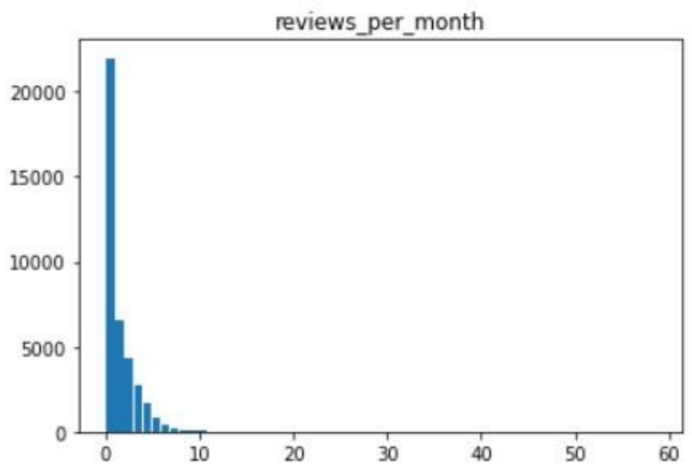
iii. To proceed further with the analysis, we have decided to drop id, name which is unnecessary and redundant. The last review column which consists of higher null values has been dropped as well.

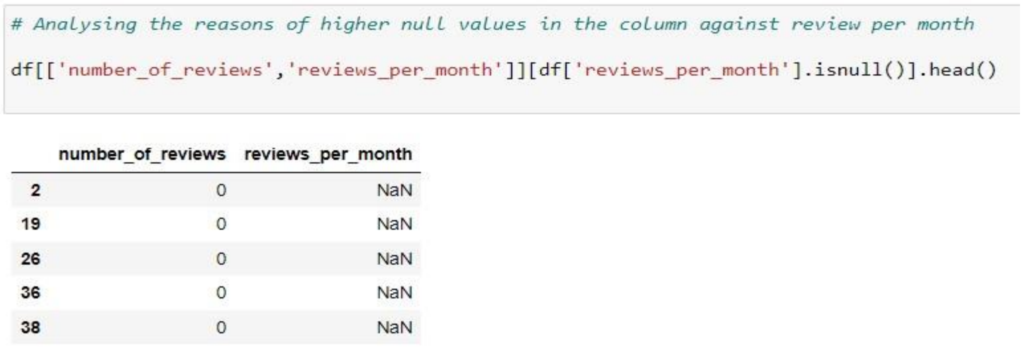
iv. The analysis for the Reviews per month column as per below, before we will decide on the suitable method to treat the missing values in the columns:

   a. The mean value for this column is 1.373, with median is 0.72 and max value is at 58.50. The distribution of this column is rightly skewed as per below histogram.

```
df.hist(column = 'reviews_per_month', bins = 60, grid = False, rwidth = .9)
```

```
array([[<AxesSubplot:title={'center':'reviews_per_month'}>]], dtype=object)
```

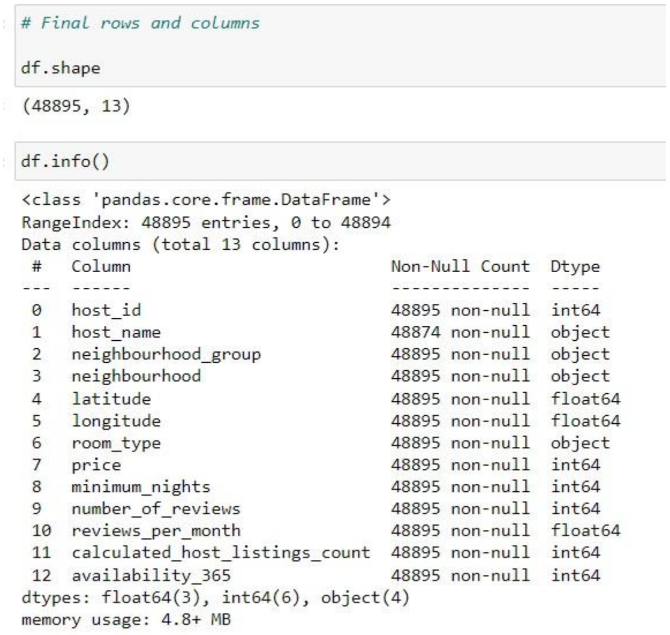




```
# Analysing the reasons of higher null values in the column against review per month

df[['number_of_reviews','reviews_per_month']][df['reviews_per_month'].isnull()].head()
```

	number_of_reviews	reviews_per_month
2	0	NaN
19	0	NaN
26	0	NaN
36	0	NaN
38	0	NaN

i. As we can see, the values in the reviews_per_month is related with the number_of_reviews column, where there is no review being added which is equivalent to null values in the reviews per month.
ii. Thus, it is logically concluded to replace those null values in the column with 0 instead, which we will proceed in the next step.



```
# Final rows and columns

df.shape
```

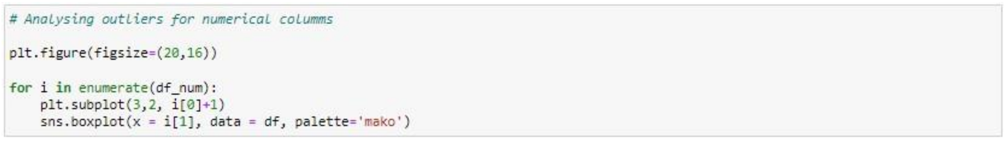
```
(48895, 13)
```

```
df.info()
```

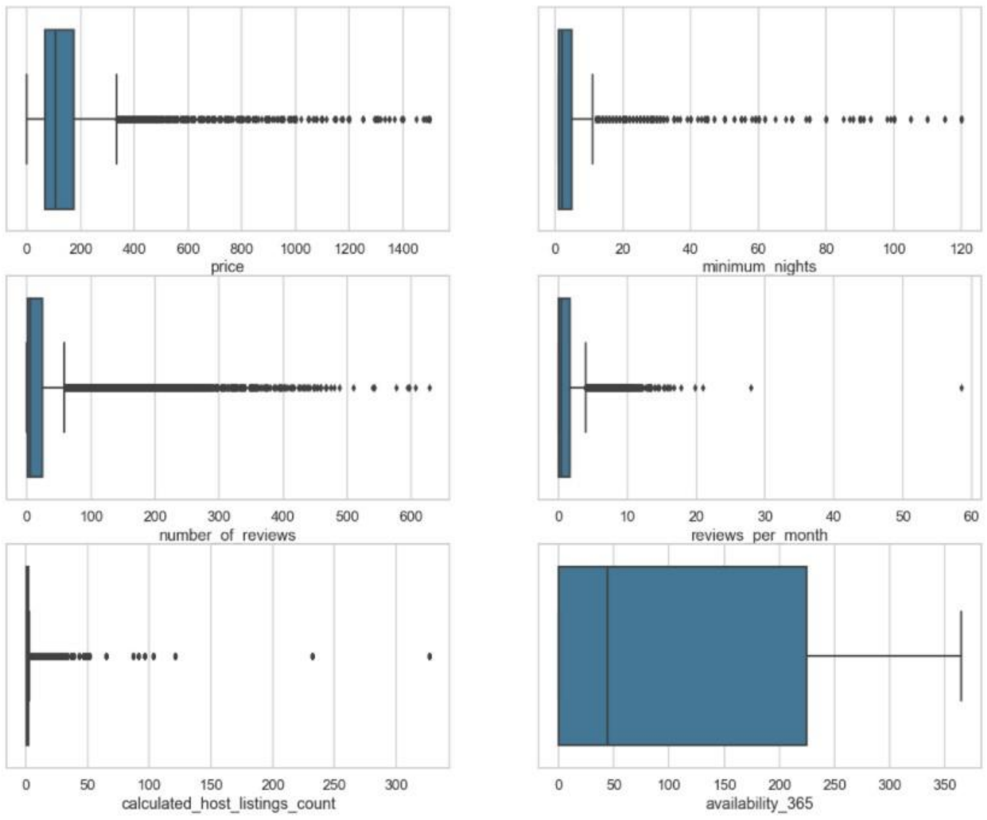
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 13 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   host_id                         48895 non-null  int64  
 1   host_name                       48874 non-null  object 
 2   neighbourhood_group             48895 non-null  object 
 3   neighbourhood                   48895 non-null  object 
 4   latitude                        48895 non-null  float64
 5   longitude                       48895 non-null  float64
 6   room_type                       48895 non-null  object 
 7   price                           48895 non-null  int64  
 8   minimum_nights                  48895 non-null  int64  
 9   number_of_reviews               48895 non-null  int64  
 10  reviews_per_month               48895 non-null  float64
 11  calculated_host_listings_count  48895 non-null  int64  
 12  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(6), object(4)
memory usage: 4.8+ MB
```

Above is the final shape of the datasets, upon completion of the null values treatments. Based on 48895 rows and 13 columns, we will further analyse the attributes in the datasets.

C. Handling The Outliers: Numerical Variables



```
# Analysing outliers for numerical columns

plt.figure(figsize=(20,16))

for i in enumerate(df_num):
    plt.subplot(3,2, i[0]+1)
    sns.boxplot(x = i[1], data = df, palette='mako')
```



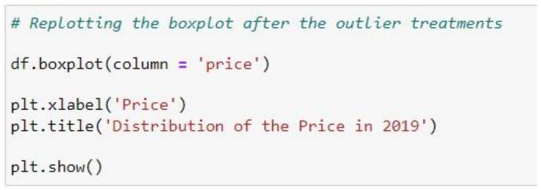
From the boxplot, we can summarize that:

- There is an obvious outlier for the price column, 10000 usd is too costly and it is unaffordable for most number of people.
- Same goes with the column minimum_nights where 1200 days which is equivalent to 3.28 years.

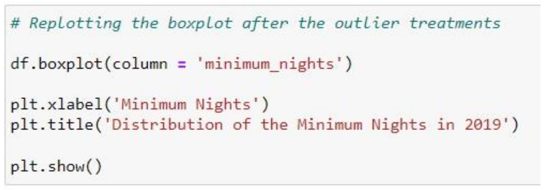
Steps:

i. For the outliers, we will mainly focus on the price and minimum nights column, and the outliers presents in the column due to the reason that this will be the important columns of our analysis further.

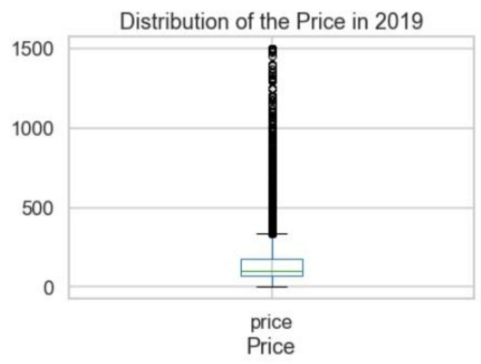
ii. The treatment of the outliers will be based on the capping of the outliers, we will bin the outliers in difference quantile as per the normal distribution rules. The quantile such as 0, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99 and 0.997 quantiles. We have dropped the values which is above the 0.997 quantile as well and it has been capped at 0.997.

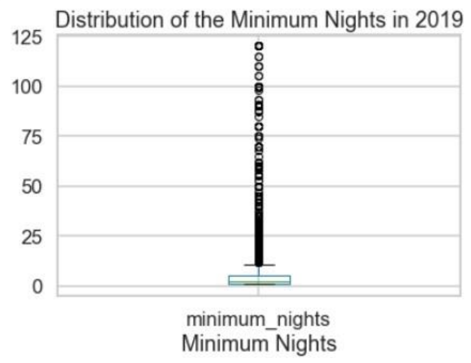


```
# Replotting the boxplot after the outlier treatments

df.boxplot(column = 'price')

plt.xlabel('Price')
plt.title('Distribution of the Price in 2019')

plt.show()
```

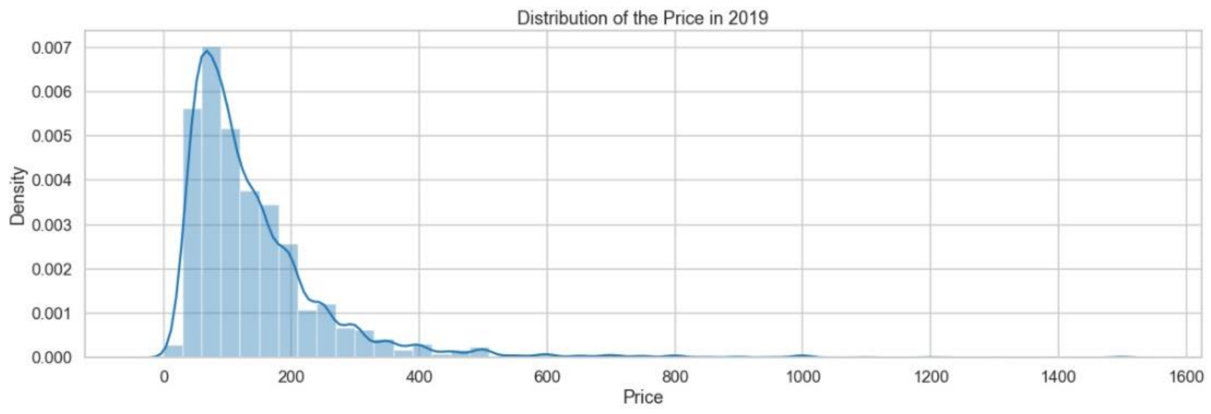


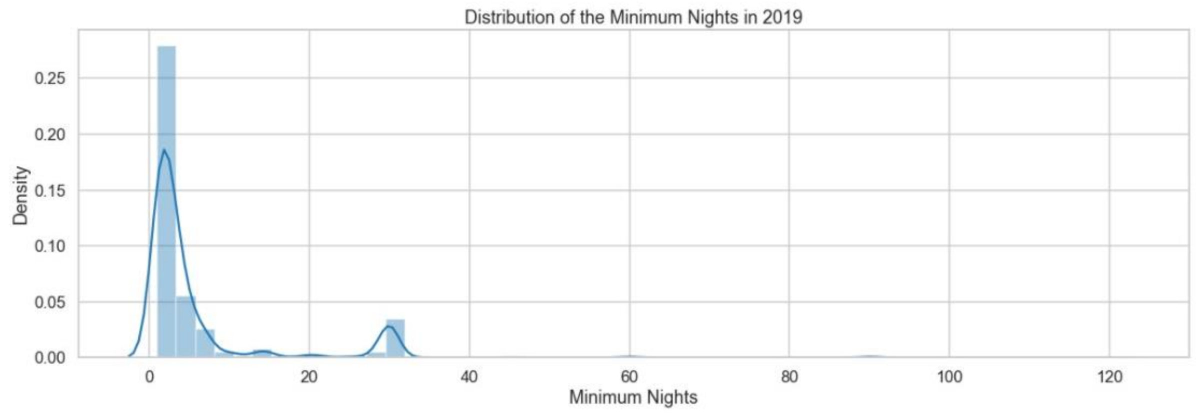
```
# Replotting the boxplot after the outlier treatments

df.boxplot(column = 'minimum_nights')

plt.xlabel('Minimum Nights')
plt.title('Distribution of the Minimum Nights in 2019')

plt.show()
```





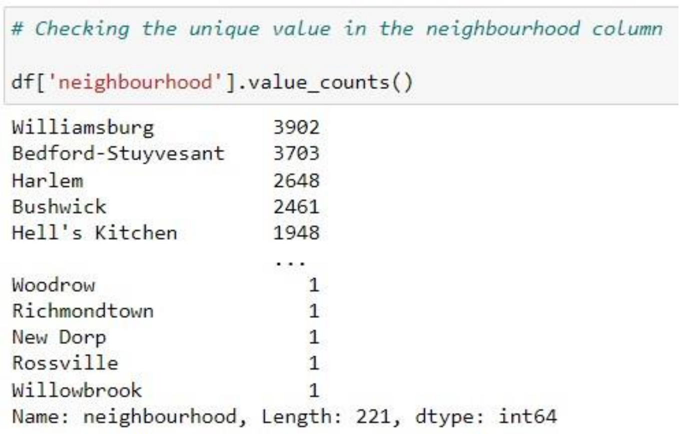
iii. It has been applied to both of the columns; the distributions are as follows:





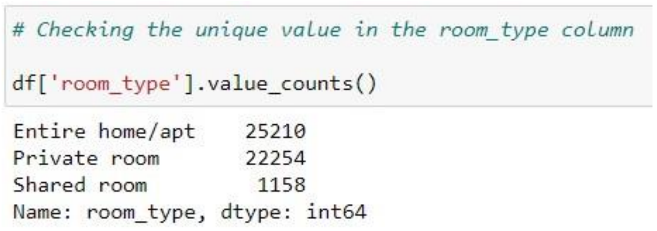
D. Categorical Variables

i. For the categorical variables, we have taken the value counts presents in each of the variable.



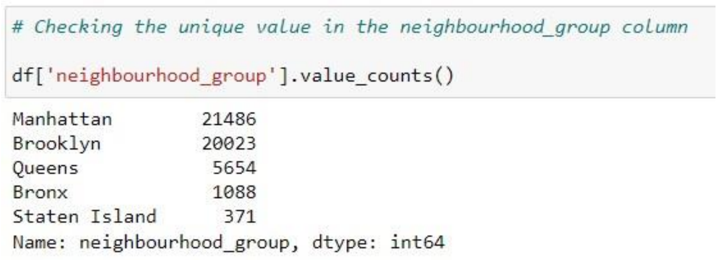
```
# Checking the unique value in the neighbourhood column

df['neighbourhood'].value_counts()
```

```
Williamsburg          3902
Bedford-Stuyvesant    3703
Harlem                2648
Bushwick              2461
Hell's Kitchen        1948
                      ...
Woodrow                  1
Richmondtown             1
New Dorp                 1
Rossville                1
Willowbrook              1
Name: neighbourhood, Length: 221, dtype: int64
```



```
# Checking the unique value in the room_type column

df['room_type'].value_counts()
```

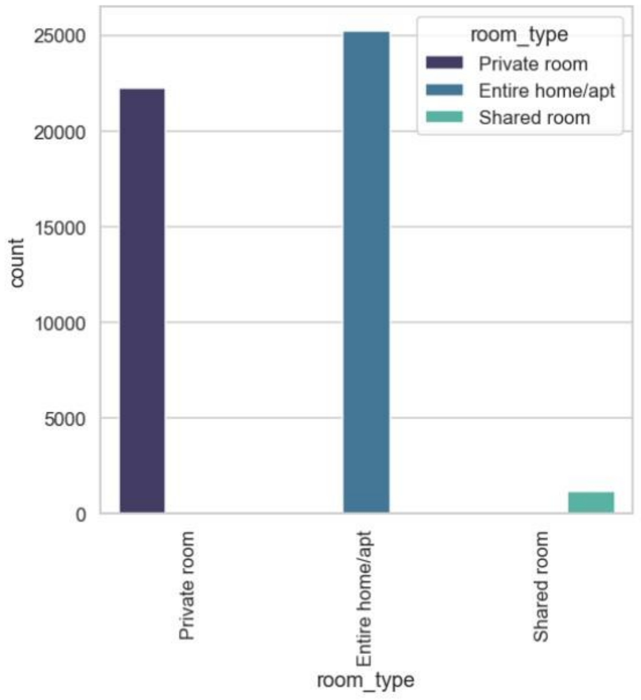
```
Entire home/apt    25210
Private room       22254
Shared room         1158
Name: room_type, dtype: int64
```

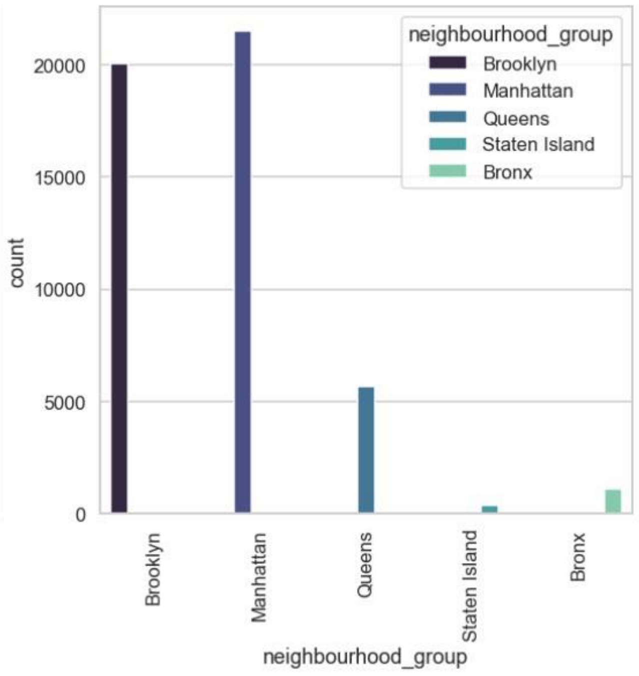


```
# Checking the unique value in the neighbourhood_group column

df['neighbourhood_group'].value_counts()
```

```
Manhattan        21486
Brooklyn         20023
Queens            5654
Bronx             1088
Staten Island      371
Name: neighbourhood_group, dtype: int64
```

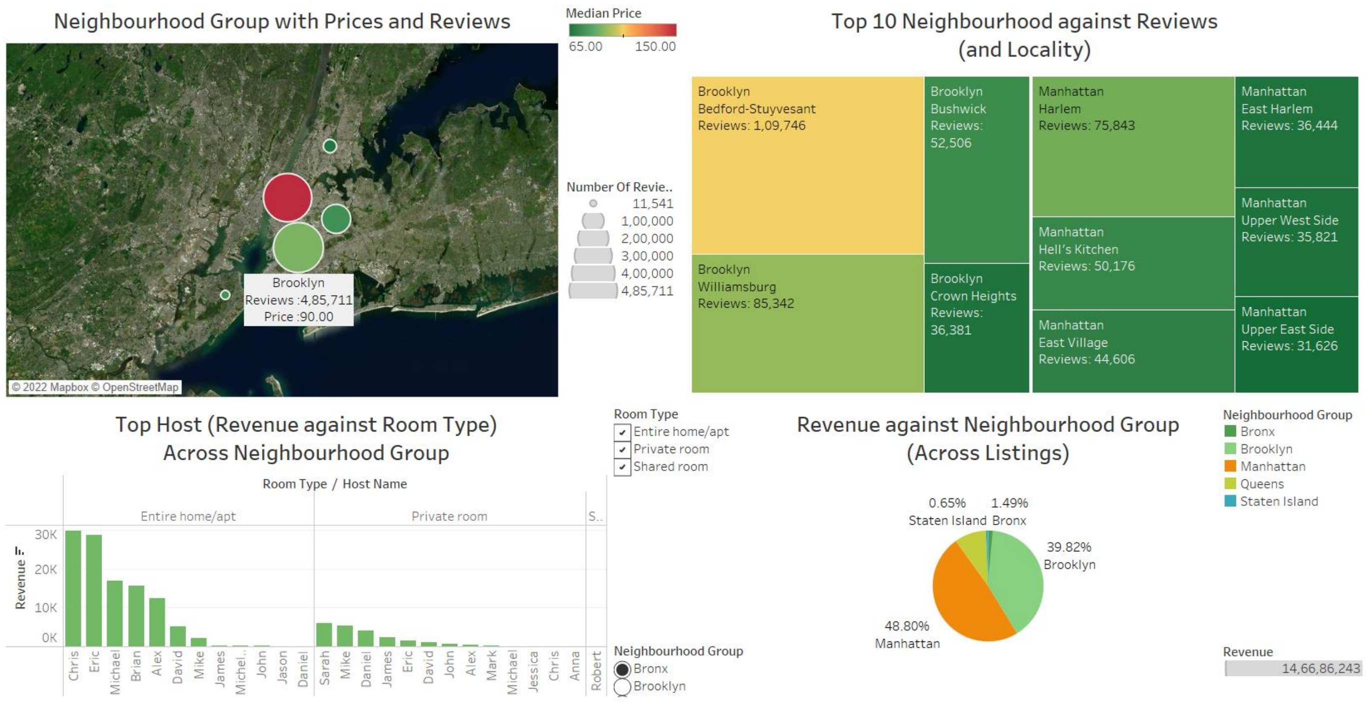
ii. We have plotted bar chart for the categorical variables, leave alone the neighbourhood column, as there are too many of unique values available in the columns.





DATA VISUALIZATIONS USING TABLEAU

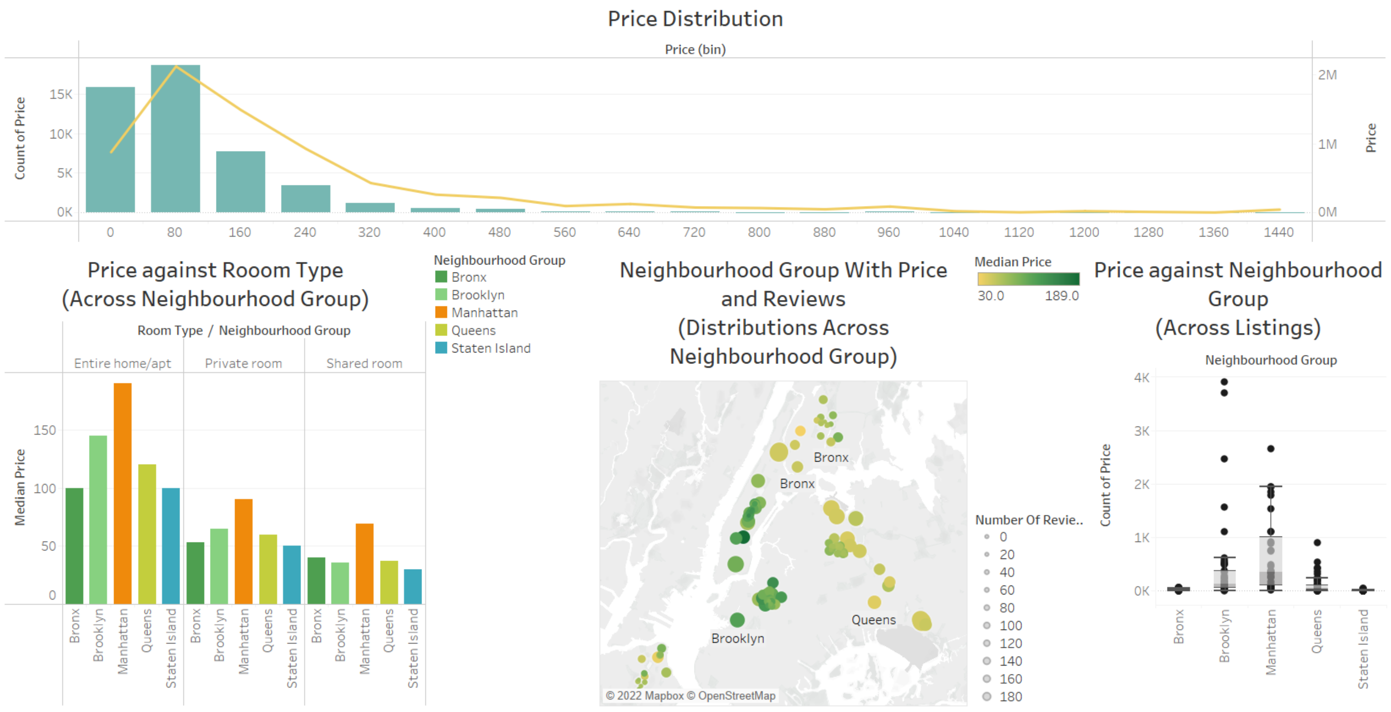
1. Overview Dashboard



Insights:

- Manhattan is the neighbourhood with the highest reviews 4,53,267 with the median price of 150.00, while Staten Island is the least with only 11,541 reviews, however it is much cheaper than Manhattan with only 75usd median price.
- Bedford – Stuyvesant is the neighbourhood with the highest reviews with 1,09,746 reviews, followed closely by Williamsburg with 85,342 reviews and Harlem with 75,843 reviews.
- Entire home/ apt is the most popular bookings with the greater revenue than the rest of the room type. Chris and Eric possess the highest revenue for the entire home/ apt while Sarah and Mike leads in the Private room category.
- Manhattan possess 48.80% of the total revenue in the listings, Brooklyn with 39.82%, Queens with 9.24%, Bronx 1.49% and least is Staten Island with 0.65%.

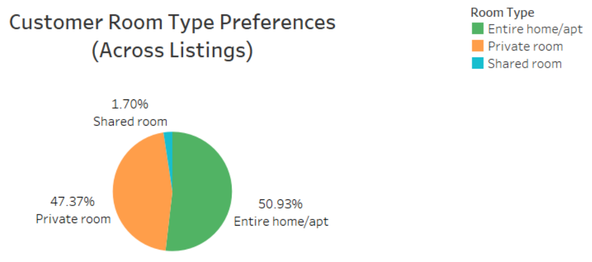
2. Pricing Perspectives

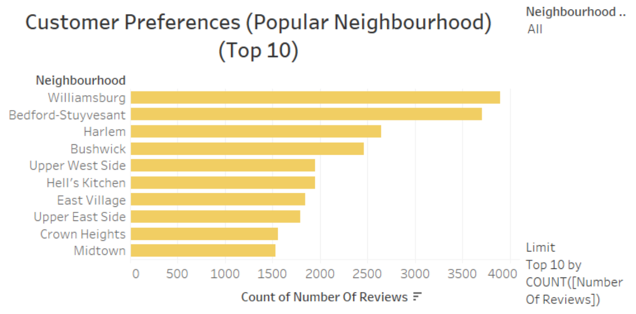


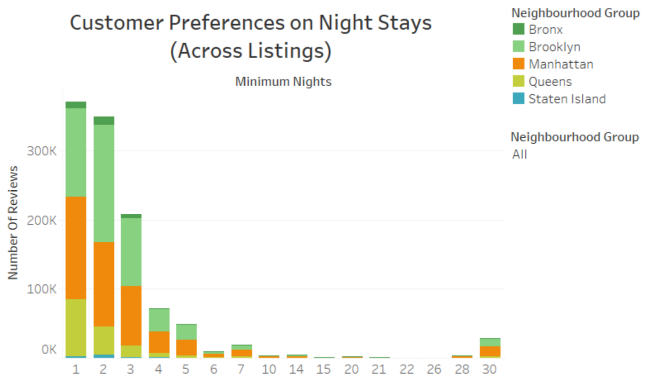
Insights:

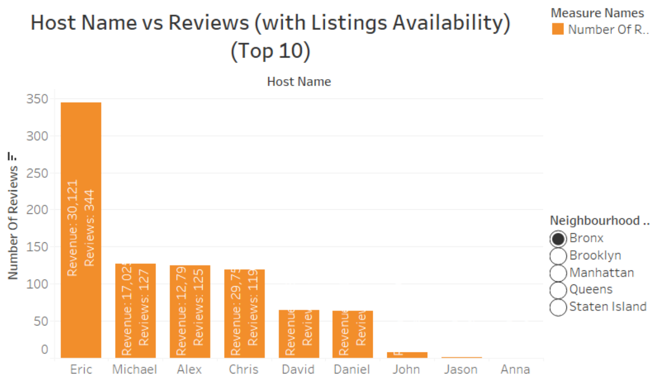
- The price distributions for the listings are rightly skewed and scattered within the range is 0-80usd, and reducing beyond 81usd and more.
- Manhattan had the highest price in the all of the categories, Entire home/ apt, Private room as well as the Shared Room. And all of the categories, States Island is having the least price. Form the map, we can clarify the distribution of the pricing across neighbour group where the higher prices are focused in the area of Manhattan and Brooklyn.
- The Boxplot plots the range of the pricing, as well as the minimum and the maximum value of the price in each of the Neighbourhood Group.
- As we can see, Manhattan having a huge Interquartile range, with greater differences in 0.25 and 0.75 quantile. However, from the boxplot, the distribution of price in Brooklyn seems to have a higher maximum price as compared with Manhattan.

3. Customer Preferences (Reviews Perspectives)









Insights:

- Customer preferred to book the Entire home/Apt with 50.93% percent of total, Private Room with 47.37% and the least in the Shared room with only 1.70%.
- The customer preferences in the Night stays mostly scattered in the range of the first 2 days and reduced significantly upon the 3rd days and so on. From the graph, we can see that the bookings done in Brooklyn and Bronx are highest within the first 2 days as well.
- The next graph is the Top 10 popular Host across listings, Eric leads the chart with 344 reviews with total revenue of 30,121usd, followed by Michael with 127 reviews and Alex with 125 reviews.

RECOMMENDATIONS

- ✓ From all of the above analysis, it will be advisable if Airbnb can provide more listings from Manhattan. As we know, it is the most densely populated borough in NYC, on top of its iconic state's buildings. Not forgotten Brooklyn as it is the second most populated borough in NYCBesides Manhattan and Brooklyn, Queen is also one of the potential boroughs which AirBnb can focus on, as there are 2 major airports in Queens, John F. Kennedy International Airport and LaGuardia Airport. The price range in this locality is acceptable and there are mostly high reviews in Queens as well. During the holiday or festive seasons when the number of inbound tourists are high, it is very important to put an extra efforts and marketing at Queens since that it is the first place of arrival for the tourists.
- ✓ The price range for each room type shall be revised to improved occupancy, for example within the range of 100-190usd for Entire home/apt, 50-90usd for Private room and 40-70usd for Shared Room across Neighbourhood Group.
- ✓ Airbnb shall focus more on providing listings for the Entire Home/apt and Private Room and least on Shared Room.
- ✓ Last but not least, Airbnb shall encourage the properties to encourage their customers to provide reviews upon check out completions. From the reviews we can further come out with analysis on ways to improving.