Storytelling Case Study: Airbnb, NYC

Problem background

Suppose that you are working as a data analyst at Airbnb. 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.

End Objective

To prepare for the next best steps that Airbnb needs to take as a business, you have been asked to analyse a dataset consisting of various Airbnb listings in New York. Based on this analysis, you need to give two presentations to the following groups.

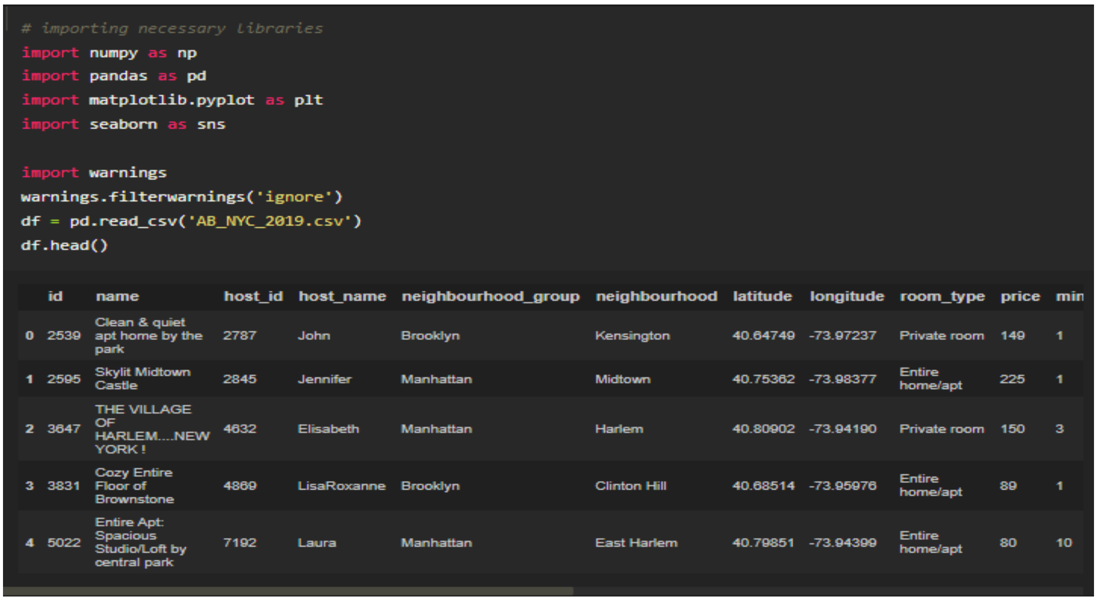
1. **Presentation - I**

- **Data Analysis Managers:** These people manage the data analysts directly for processes and their technical expertise is basic.
- **Lead Data Analyst:** The lead data analyst looks after the entire team of data and business analysts and is technically sound.

2. **Presentation - II**

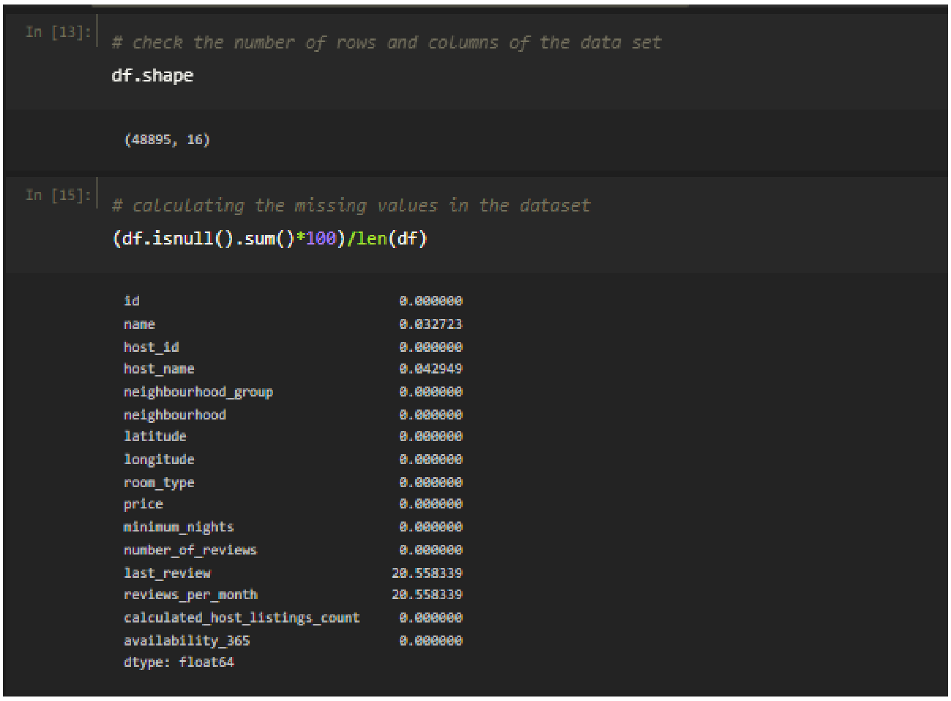
- **Head of Acquisitions and Operations, NYC:** This head looks after all the property and host acquisitions and operations. Acquisition of the best properties, price negotiation, and negotiating the services the properties offer falls under the purview of this role.
- **Head of User Experience, NYC:** The head of user experience looks after the customer preferences and also handles the properties listed on the website and the Airbnb app. Basically, the head of user experience tries to optimise the order of property listing in certain neighbourhoods and cities in order to get every property the optimal amount of traction.

Data Import and Preprocessing using Python:



```
# importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('AB_NYC_2019.csv')
df.head()
```

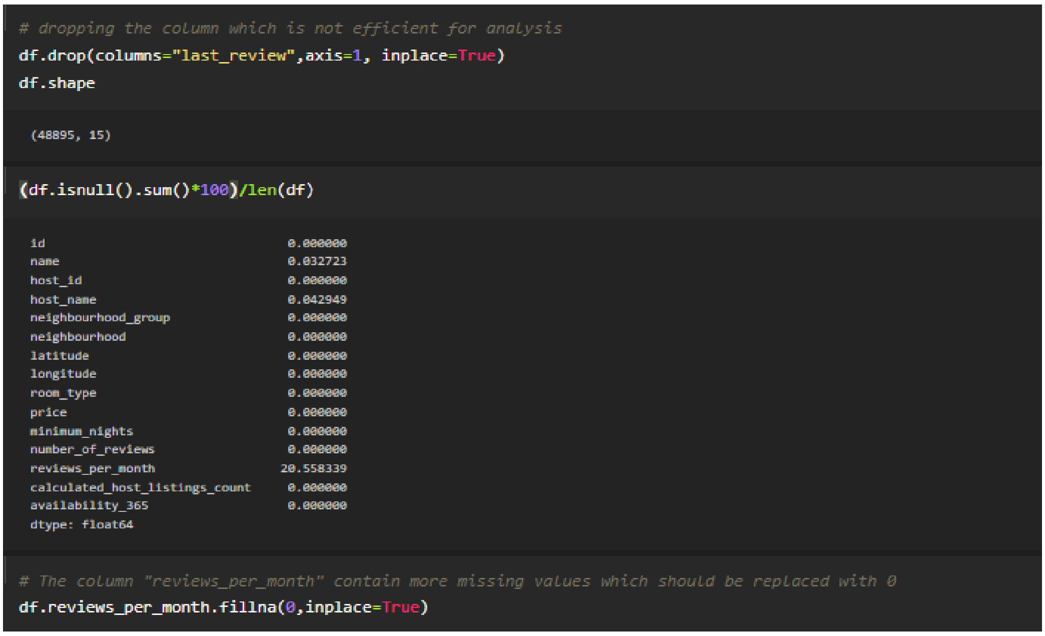
	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	min
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	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



```
In [13]: # check the number of rows and columns of the data set
         df.shape

          (48895, 16)

In [15]: # calculating the missing values in the dataset
         (df.isnull().sum()*100)/len(df)

          id                                 0.000000
          name                               0.032723
          host_id                            0.000000
          host_name                          0.042949
          neighbourhood_group                0.000000
          neighbourhood                      0.000000
          latitude                           0.000000
          longitude                          0.000000
          room_type                          0.000000
          price                              0.000000
          minimum_nights                     0.000000
          number_of_reviews                  0.000000
          last_review                       20.558339
          reviews_per_month                 20.558339
          calculated_host_listings_count     0.000000
          availability_365                   0.000000
          dtype: float64
```

- The dataset "AB_NYC_2019.csv" contains 48895 rows and 16 columns

- The columns last_review and reviews_per_month contain large number of missing values.



```
# dropping the column which is not efficient for analysis
df.drop(columns="last_review",axis=1, inplace=True)
df.shape
```

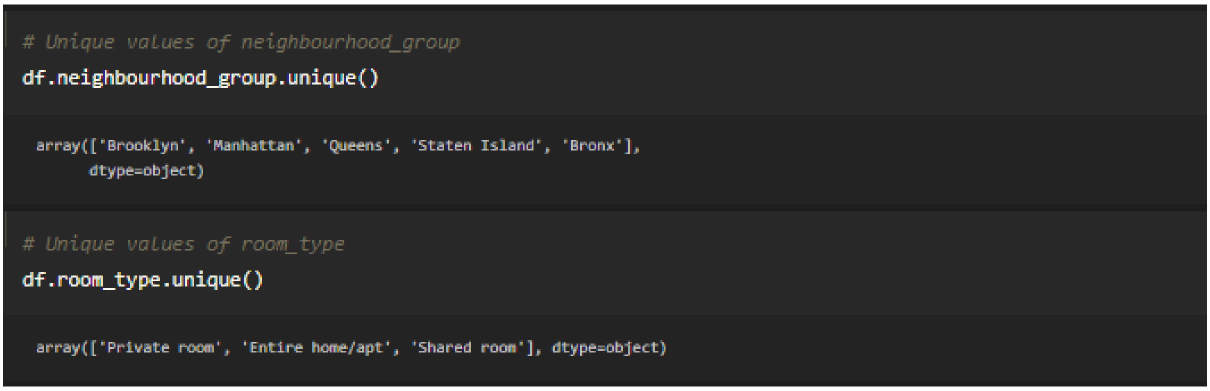
```
(48895, 15)
```

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

```
id                                 0.000000
name                               0.032723
host_id                            0.000000
host_name                          0.042949
neighbourhood_group                0.000000
neighbourhood                      0.000000
latitude                           0.000000
longitude                          0.000000
room_type                          0.000000
price                              0.000000
minimum_nights                     0.000000
number_of_reviews                  0.000000
reviews_per_month                 20.558339
calculated_host_listings_count     0.000000
availability_365                   0.000000
dtype: float64
```

```
# The column "reviews_per_month" contain more missing values which should be replaced with 0
df.reviews_per_month.fillna(0,inplace=True)
```

- Dropped the column “last_review” is not efficient for analysis.
- Handled the missing values in “reviews_per_month” by replacing the missing values with 0



```
# Unique values of neighbourhood_group
df.neighbourhood_group.unique()
```

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

```
# Unique values of room_type
df.room_type.unique()
```

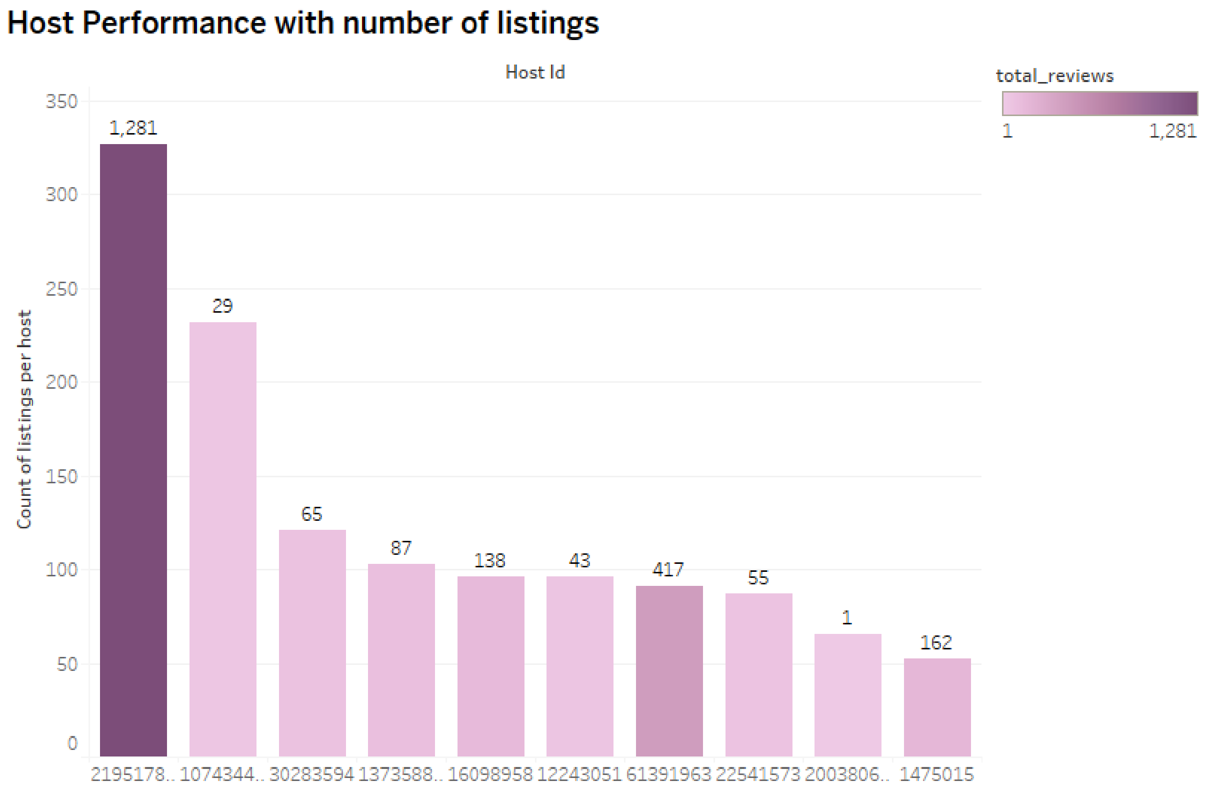
```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

PPT – I :

Data analysis and visualization using Tableu:

1. Host Listings vs Reviews :

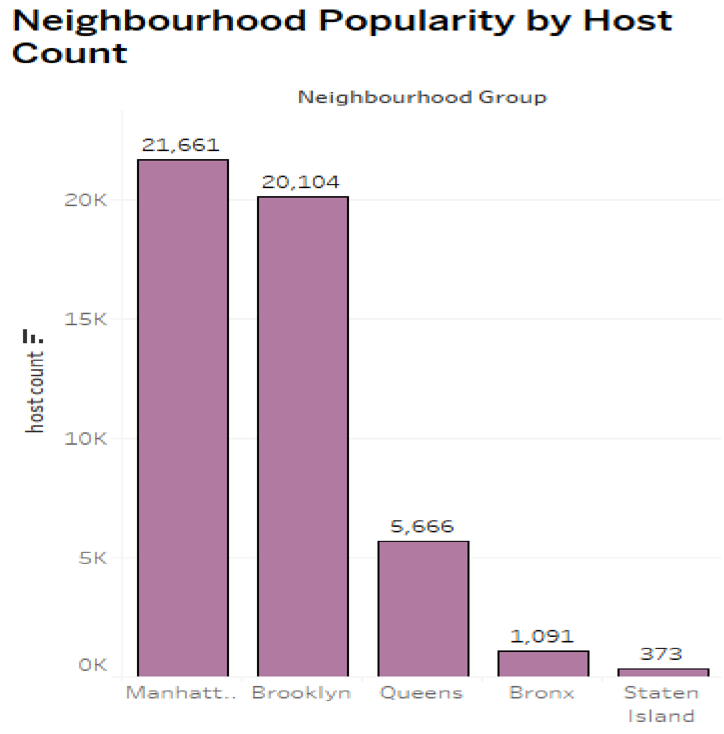
- We created the bar chart comparing the number of listings and reviews per host.
- We took Calculated host listings in rows, top 10 Host ids in columns and Total reviews in colour marks and label.


Host Performance with number of listings

Average of Calculated Host Listings Count for each Host Id. Colour shows total_reviews. The marks are labelled by sum of Number Of Reviews. The view is filtered on Host Id, which keeps 10 of 37,457 members.

2. Neighbourhood popularity by host count :

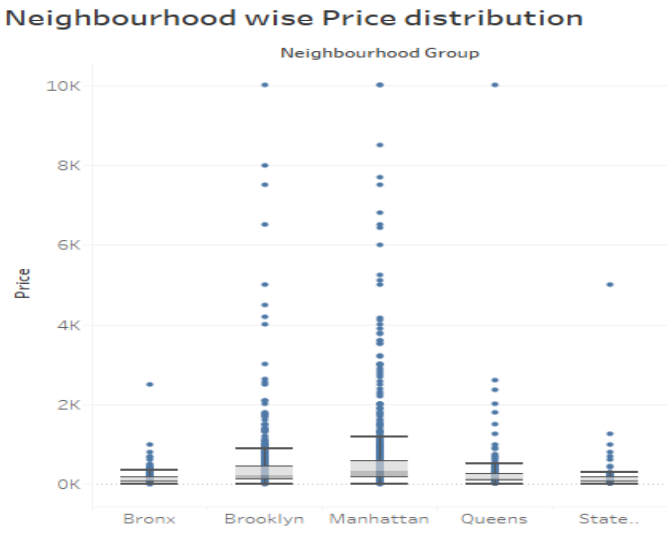
- We created the bar chart showing the neighbourhood popularity by host count.
- We took neighbourhood group in rows and Host count in columns.



Host count for each Neighbourhood Group. The marks are labelled by host count.

3. Neighbourhood wise price distribution :

- We created the boxplot the price distribution across neighbourhood groups.
- We took in Price in rows and neighbourhood group in columns.

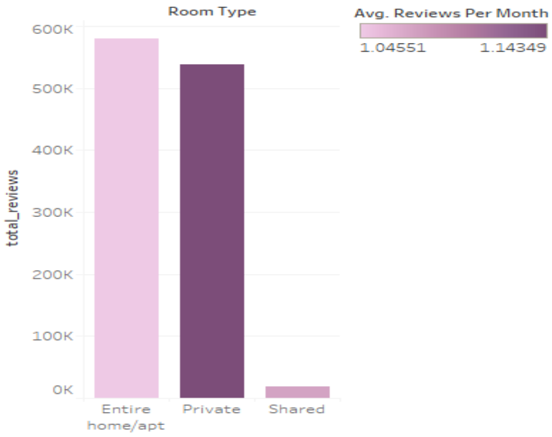


Price for each Neighbourhood Group.

4. Room Type Performance :

- We created the bar chart displaying the number of reviews for different room types.
- We took number of reviews, room type in column and Average reviews per month in colour marks.

Room type performance

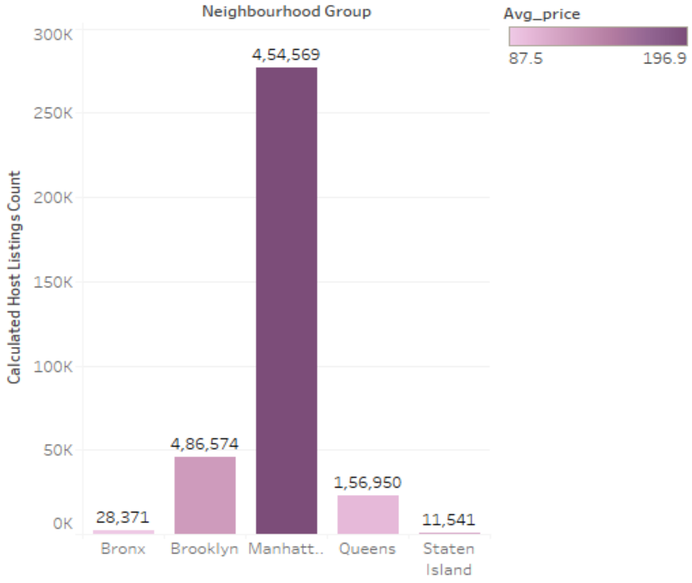


Total_reviews for each Room Type. Colour shows average of Reviews Per Month.

5. Listings per Neighbourhood :

- We created the bar chart comparing the number of listing and average price per neighbourhood group.
- We took Number of listings, Neighbourhood groups in column and Average price in colour marks.

Neighbourhoods to Target for Host Acquisition



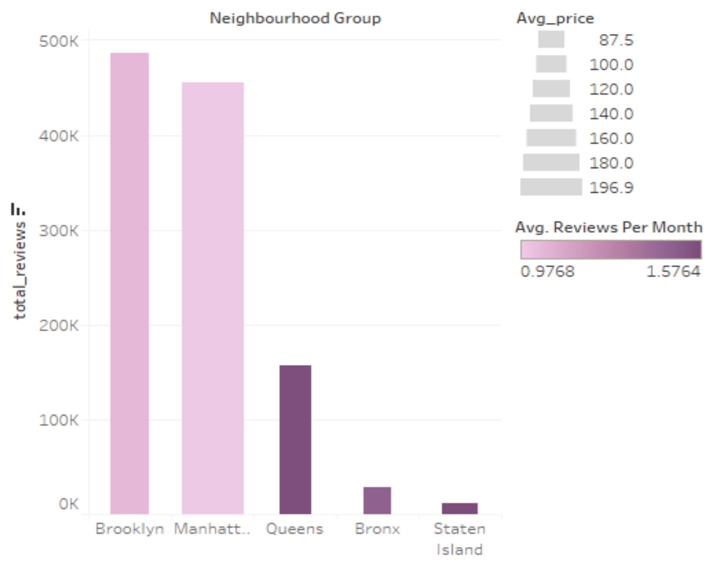
Sum of Calculated Host Listings Count for each Neighbourhood Group. Colour shows Avg_price. The marks are labelled by total_reviews.

6. Most popular localities and properties :

- We created the bar chart comparing the total reviews and Average price per Neighbourhood group.

- We took Total reviews in rows, neighbourhood group in columns, Average reviews per month in colour marks showing different levels of guest engagement in each neighbourhood group and Average price in size mark which reflects the average price of listings in each neighbourhood group.

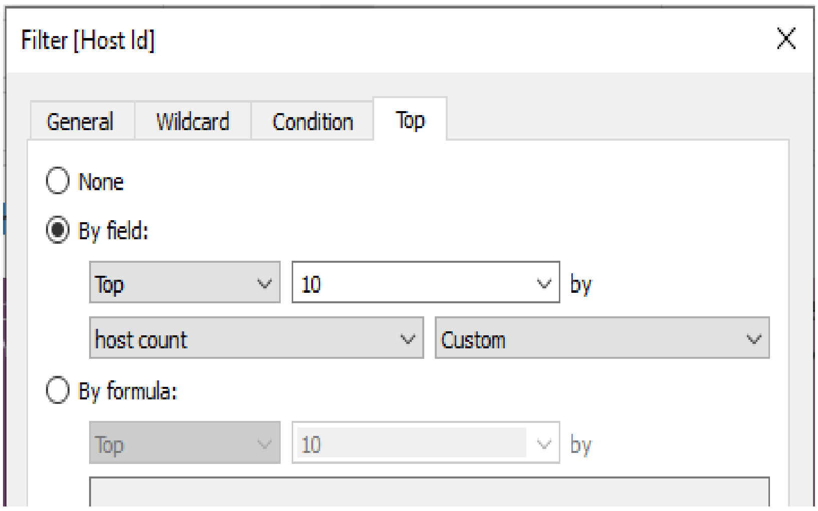
Most Popular Localities & Properties



Total_reviews for each Neighbourhood Group. Colour shows average of Reviews Per Month. Size shows Avg_price.

7. Top 10 hosts :

- We identified the top 10 Host Ids, Host Name with count of Host Ids using the treemap.



Filter [Host Id]

General | Wildcard | Condition | Top

○ None

● By field:

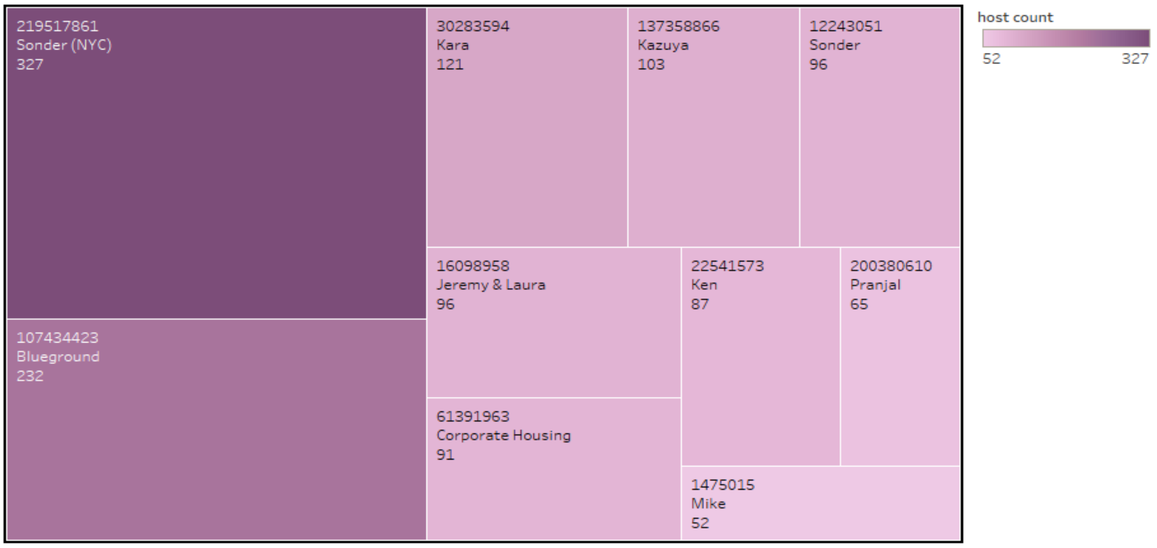
Top | 10 | by

host count | Custom

○ By formula:

Top | 10 | by

Top 10 hosts

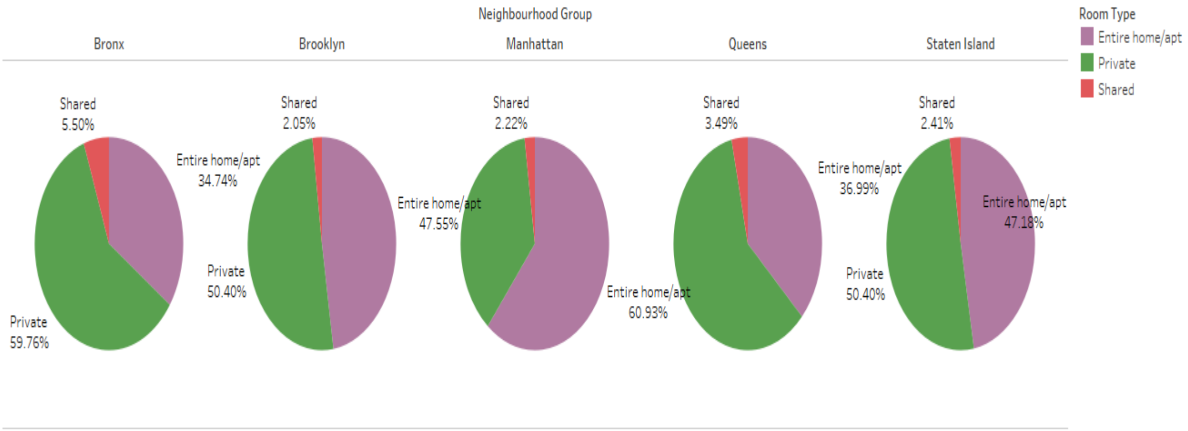


Host Id, Host Name and host count. Colour shows host count. Size shows host count. The marks are labelled by Host Id, Host Name and host count. The view is filtered on Host Id, which keeps 10 of 37,457 members.

8. Room type preferences wrt Neighbourhood groups :

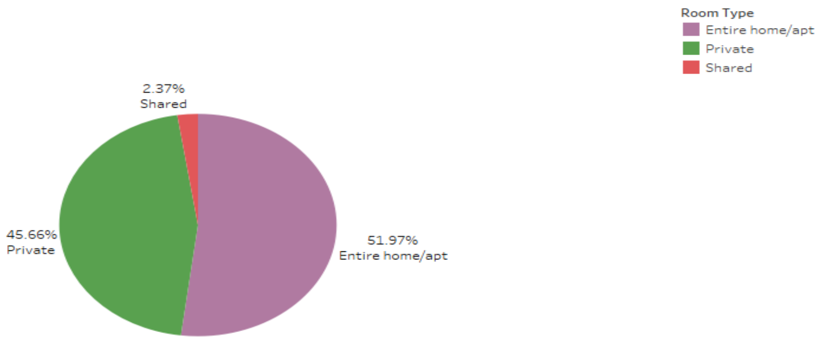
- We created a pie chart for understanding the percentage of room type preferred wrt neighbourhood group
- We added Room Type to the colours Marks card to highlight the different Room type in different colours and count of Host Id to the size

Room type wrt Neighbourhood group



Room Type and % of Total listing id count broken down by Neighbourhood Group. Colour shows details about Room Type. Size shows % of Total listing id count. The marks are labelled by Room Type and % of Total listing id count.

Overall listings wrt room type

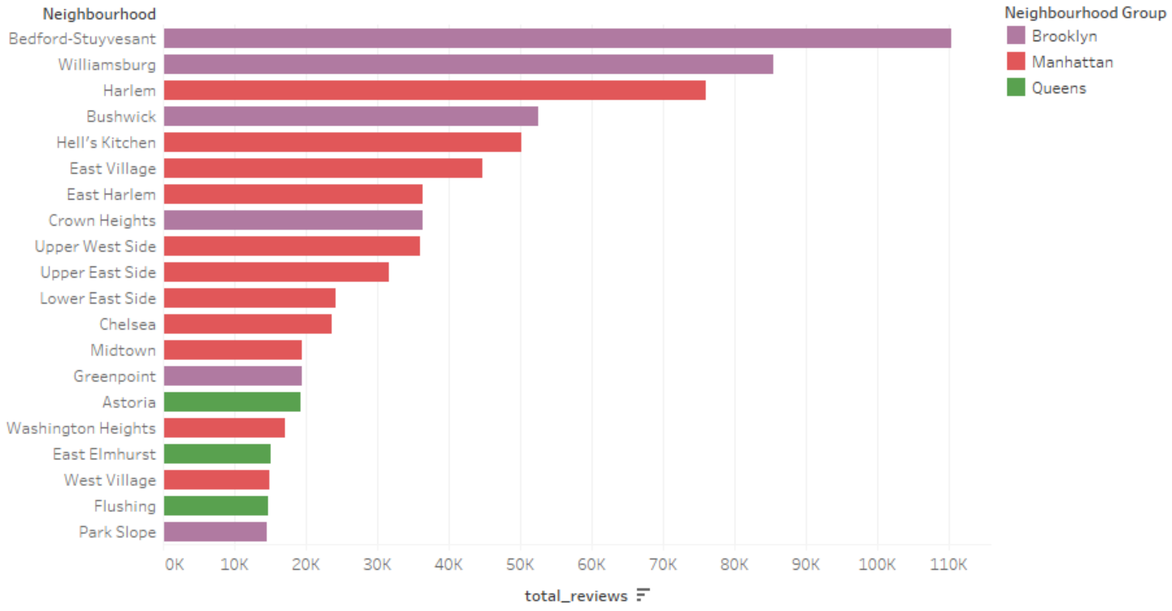


% of Total listing id count and Room Type. Colour shows details about Room Type. Size shows listing id count. The marks are labelled by % of Total listing id count and Room Type.

9. Popular Neighbourhoods :

- We created the bar plot showing the popular neighbourhoods of NYC
- We took neighbourhood in rows and total reviews in column and took neighbourhood groups in colour mark. We used filter to show Top 20 neighbours as per the total reviews.

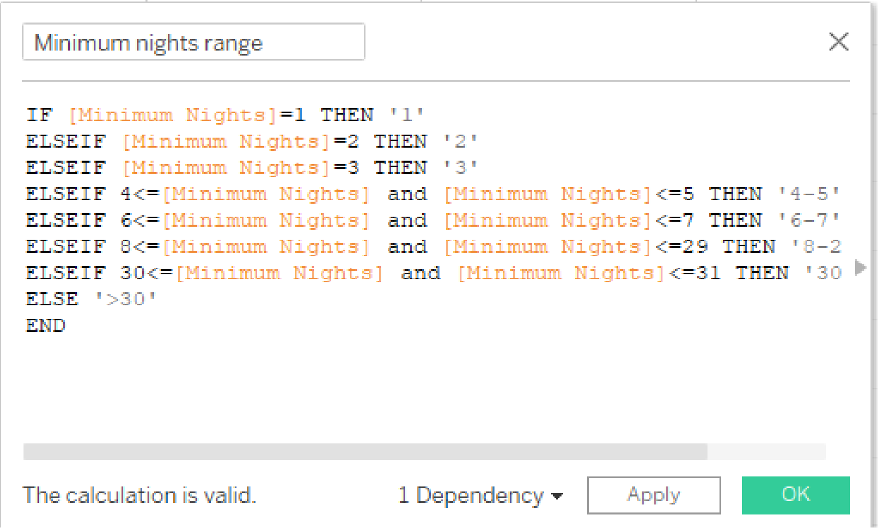
Popular Neighborhoods



Total_reviews for each Neighbourhood. Colour shows details about Neighbourhood Group. The view is filtered on Neighbourhood, which keeps 20 of 221 members.

10. Customer bookings wrt to minimum nights :

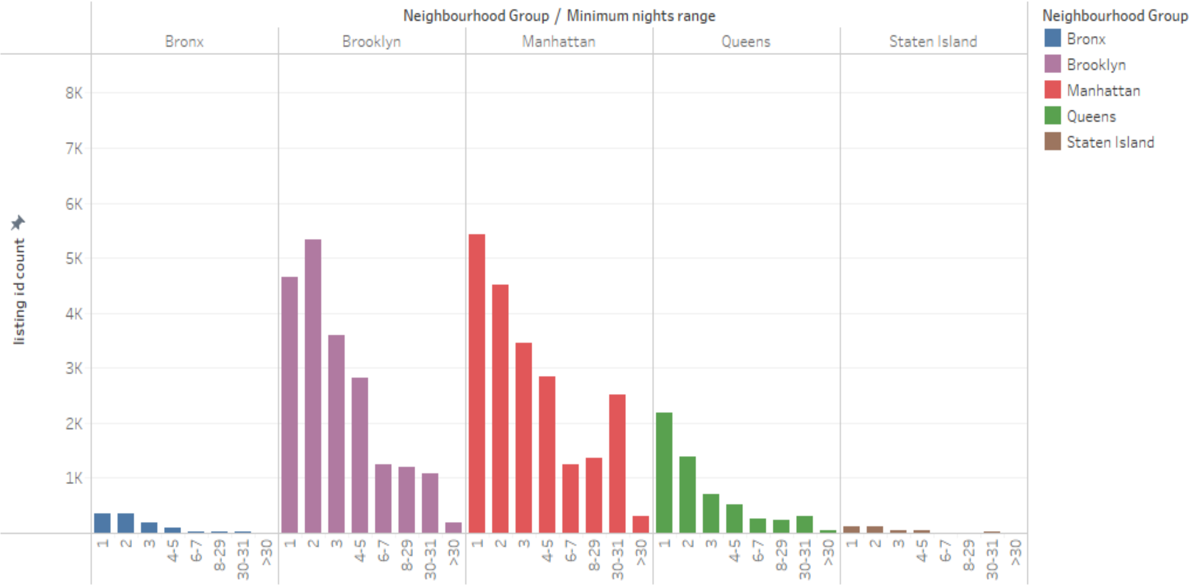
- We created the bin for Minimum nights as shown below.
- The bins were used to display the distribution of minimum nights based on the number of listing ids booked for each neighbourhood group.



```
IF [Minimum Nights]=1 THEN '1'
ELSEIF [Minimum Nights]=2 THEN '2'
ELSEIF [Minimum Nights]=3 THEN '3'
ELSEIF 4<=[Minimum Nights] and [Minimum Nights]<=5 THEN '4-5'
ELSEIF 6<=[Minimum Nights] and [Minimum Nights]<=7 THEN '6-7'
ELSEIF 8<=[Minimum Nights] and [Minimum Nights]<=29 THEN '8-2
ELSEIF 30<=[Minimum Nights] and [Minimum Nights]<=31 THEN '30
ELSE '>30'
END
```

The calculation is valid.

Customer bookings with respect to minimum nights



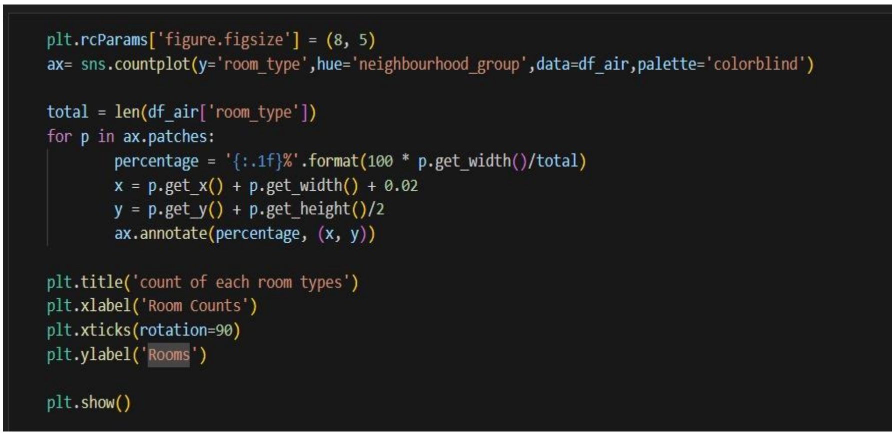
Listing id count for each Minimum nights range broken down by Neighbourhood Group. Colour shows details about Neighbourhood Group.

PPT – II :

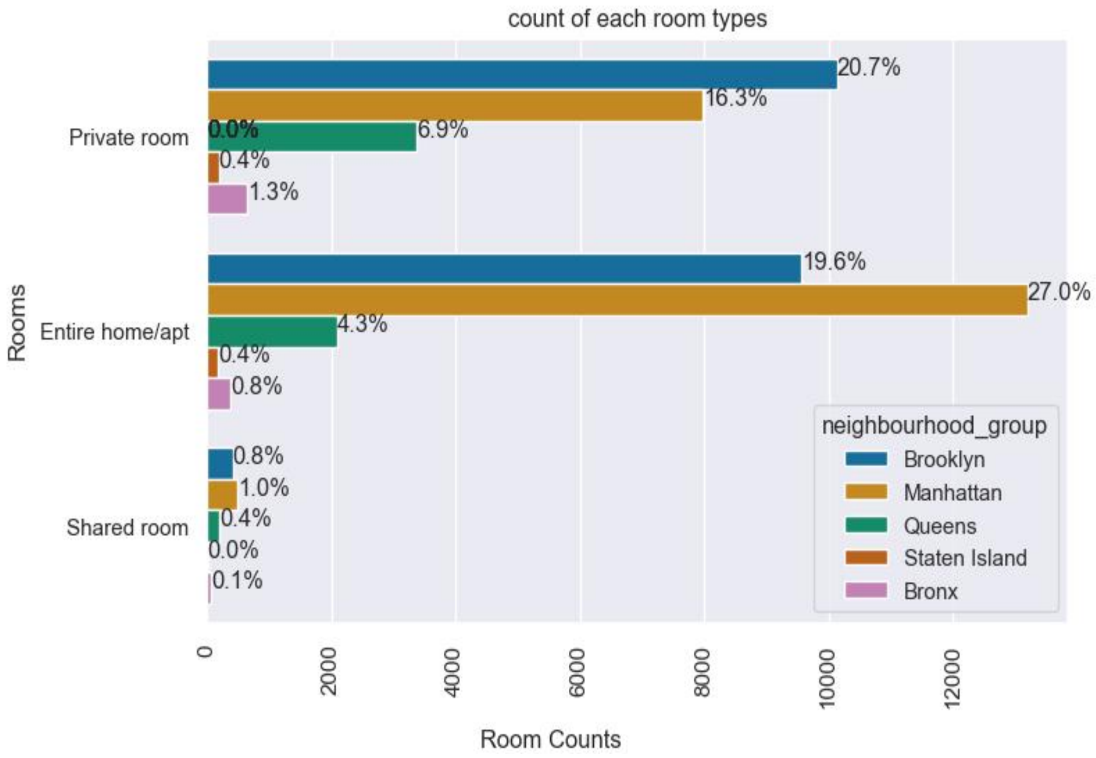
Data analysis and visualization using Python :

1. Key findings from Room types :

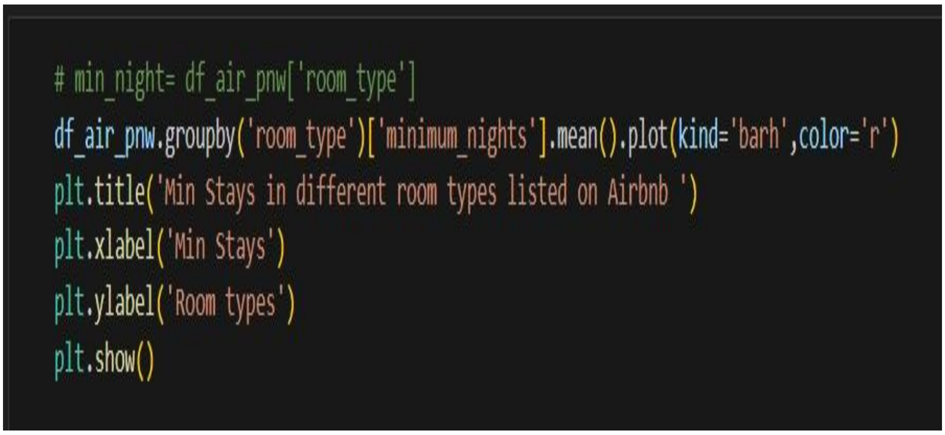
- We plotted a horizontal stacked bar chart to visualize the distribution of room types across different neighbourhood groups, showing the percentage of each room type.



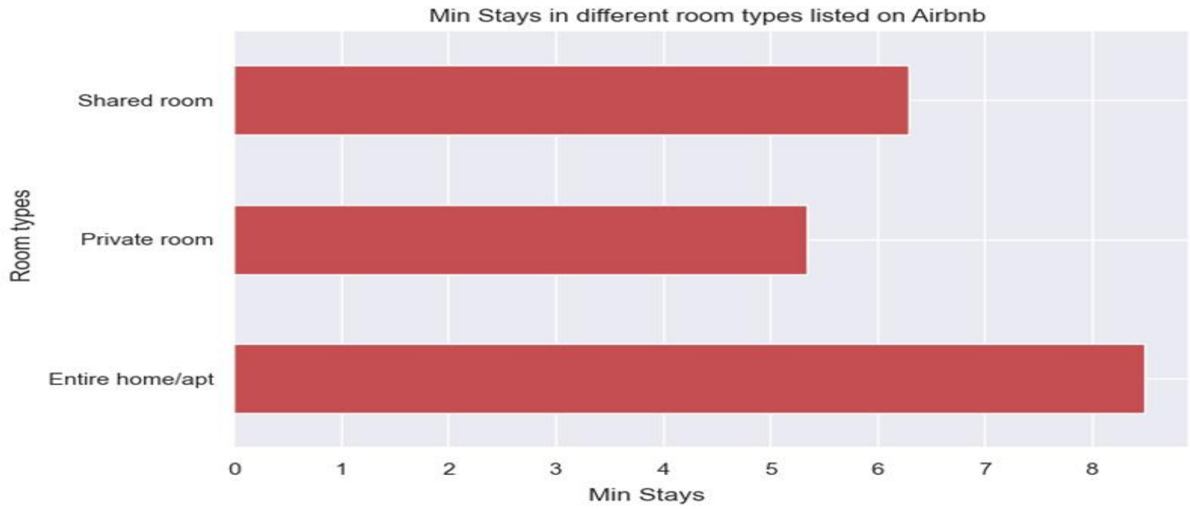
```
plt.rcParams['figure.figsize'] = (8, 5)
ax= sns.countplot(y='room_type',hue='neighbourhood_group',data=df_air,palette='colorblind')

total = len(df_air['room_type'])
for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_width()/total)
        x = p.get_x() + p.get_width() + 0.02
        y = p.get_y() + p.get_height()/2
        ax.annotate(percentage, (x, y))

plt.title('count of each room types')
plt.xlabel('Room Counts')
plt.xticks(rotation=90)
plt.ylabel('Rooms')

plt.show()
```



- We plotted the horizontal bar chart which represents the varying levels of minimum stay requirements across different room types, indicating how hosts set different booking policies based on room category.

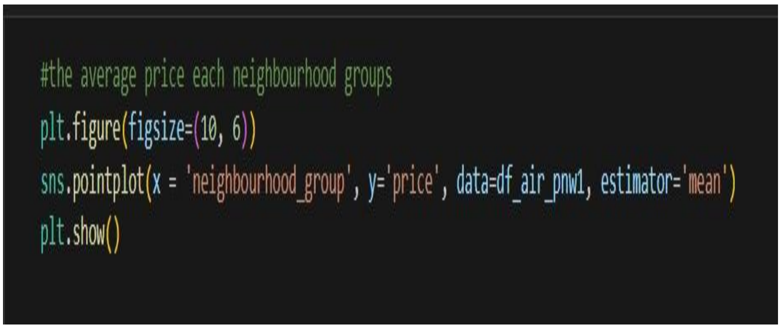


```
# min_night= df_air_pnw['room_type']
df_air_pnw.groupby('room_type')['minimum_nights'].mean().plot(kind='barh',color='r')
plt.title('Min Stays in different room types listed on Airbnb ')
plt.xlabel('Min Stays')
plt.ylabel('Room types')
plt.show()
```

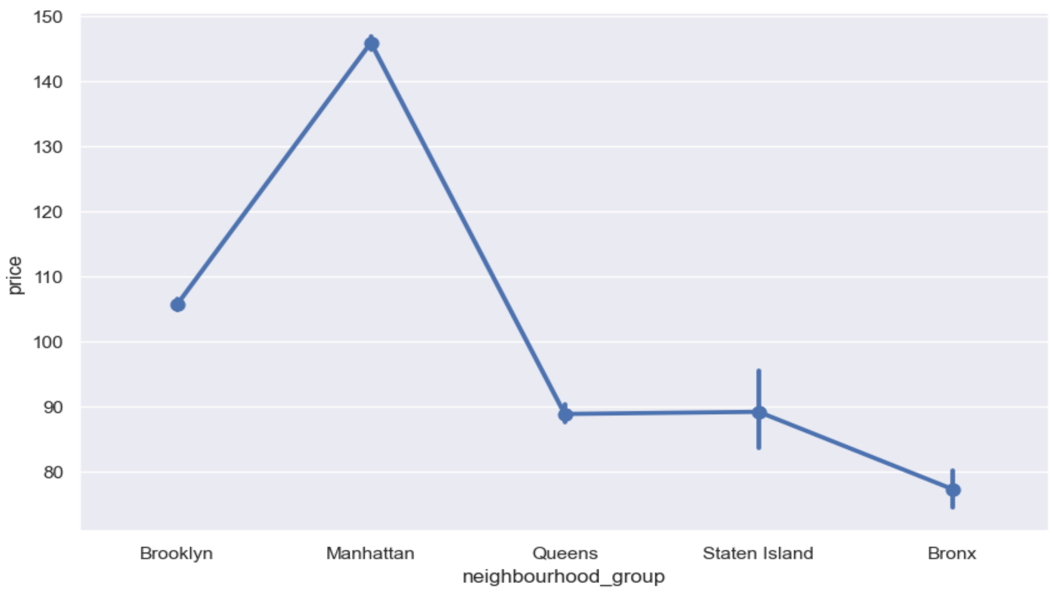


2. Key findings from Nighbourhoods :

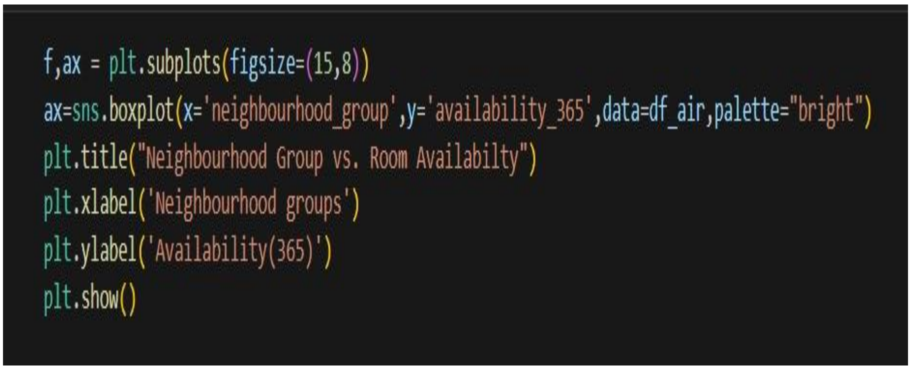
- We plotted line Chart (Neighborhood Group vs. Price) which shows price trends across different neighborhoods, highlighting how prices fluctuate within the five boroughs of NYC.



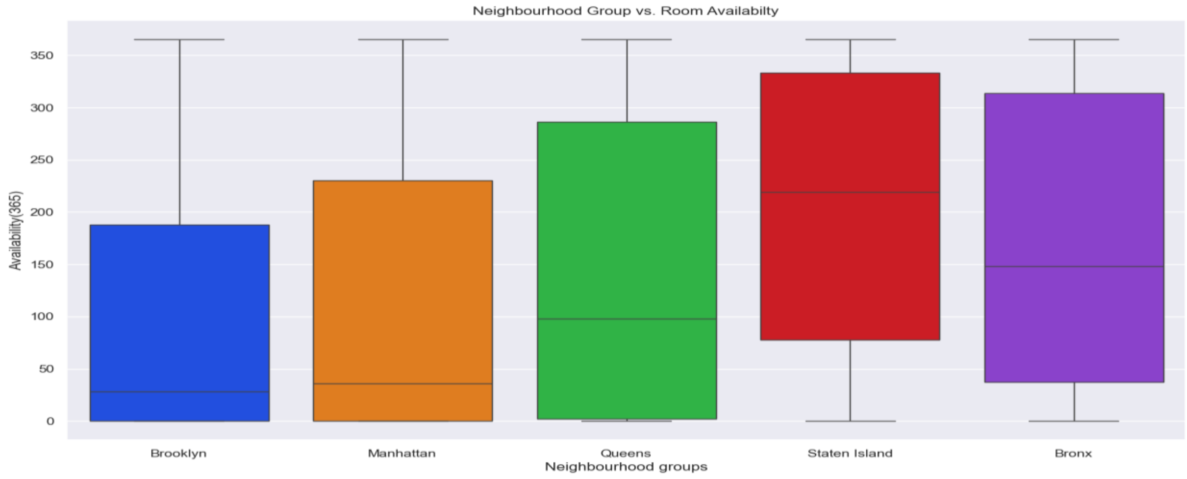
```
#the average price each neighbourhood groups
plt.figure(figsize=(10, 6))
sns.pointplot(x = 'neighbourhood_group', y='price', data=df_air_pnw1, estimator='mean')
plt.show()
```



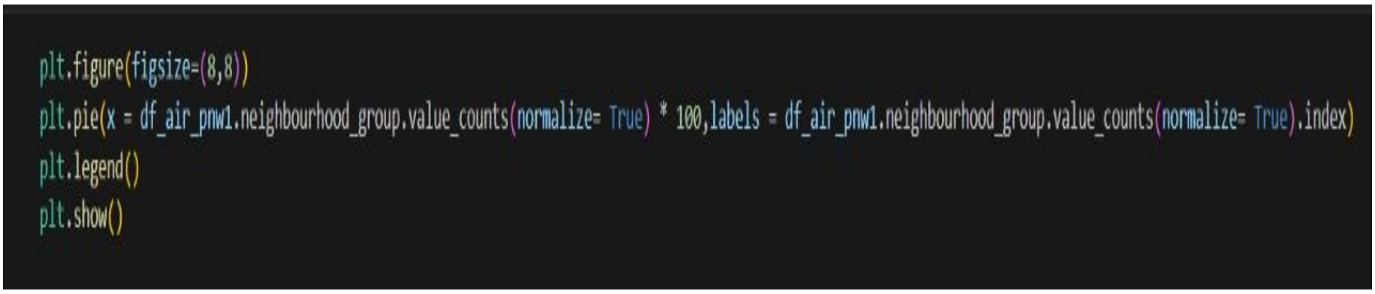
- We plotted boxplot (Availability of Rooms vs. Neighborhood Group) which depicts the distribution of room availability throughout the year across different neighbourhoods, showing variations in room booking frequency.



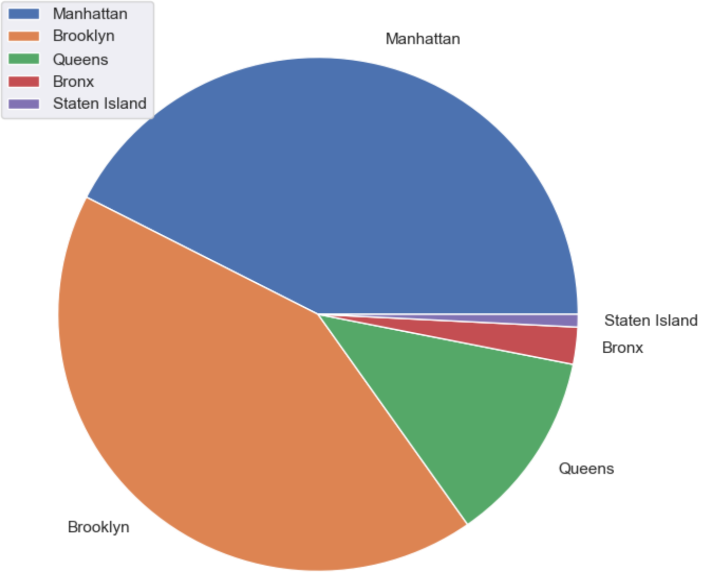
```
f,ax = plt.subplots(figsize=(15,8))
ax=sns.boxplot(x='neighbourhood_group',y='availability_365',data=df_air,palette="bright")
plt.title("Neighbourhood Group vs. Room Availabilty")
plt.xlabel('Neighbourhood groups')
plt.ylabel('Availability(365)')
plt.show()
```

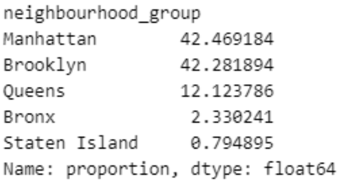


- We plotted pie chart which shows the proportion of listings across NYC's boroughs, illustrating which neighbourhoods have the most or fewest rental properties.



```
plt.figure(figsize=(8,8))
plt.pie(x = df_air_pnw1.neighbourhood_group.value_counts(normalize= True) * 100,labels = df_air_pnw1.neighbourhood_group.value_counts(normalize= True).index)
plt.legend()
plt.show()
```

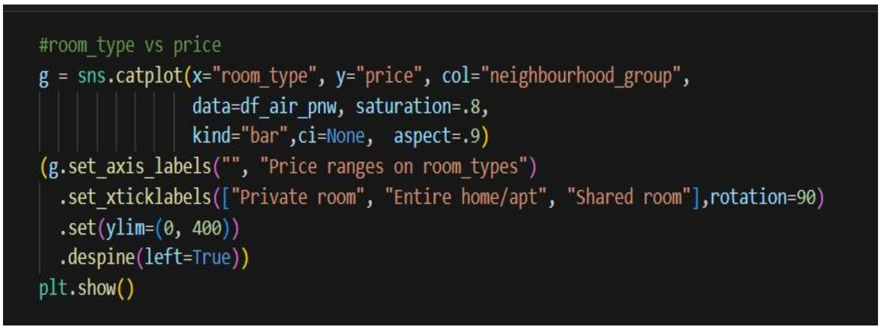




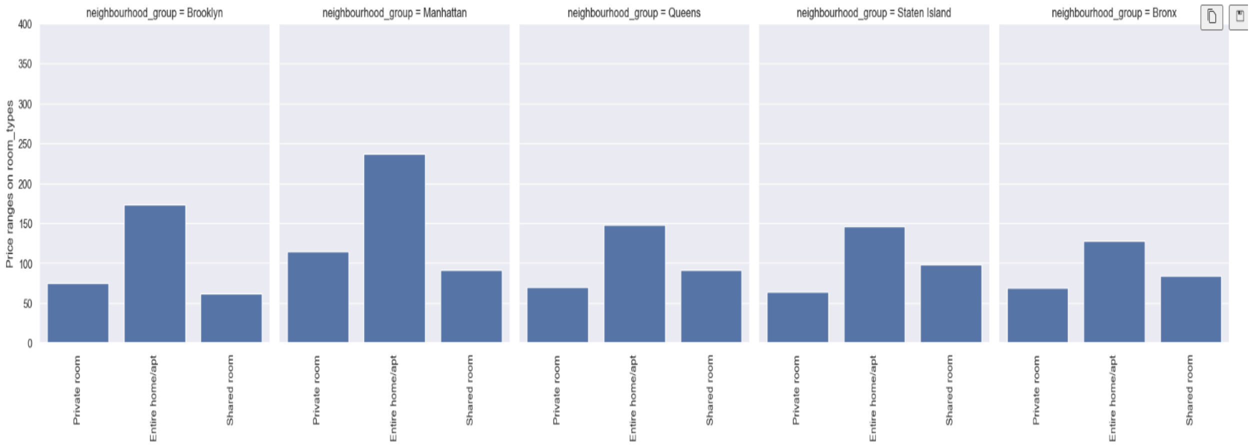
```
neighbourhood_group
Manhattan        42.469184
Brooklyn         42.281894
Queens           12.123786
Bronx             2.330241
Staten Island     0.794895
Name: proportion, dtype: float64
```

3. Cost of living :

- We plotted bar chart (Room Type vs. Price, Faceted by Neighbourhood Group) which compares the average price for different room types across neighbourhood groups, revealing how prices vary for each room type in different areas.

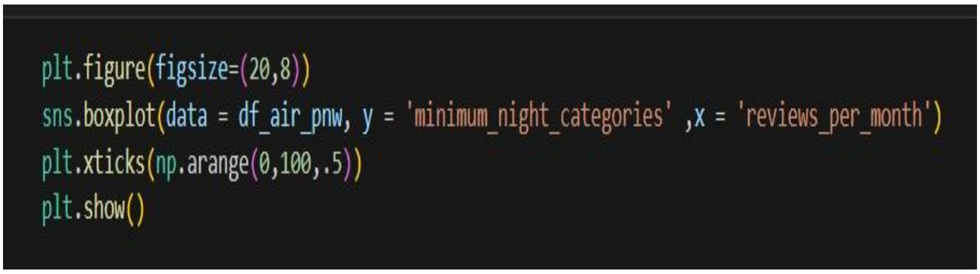


```
#room_type vs price
g = sns.catplot(x="room_type", y="price", col="neighbourhood_group",
                data=df_air_pnw, saturation=.8,
                kind="bar",ci=None,  aspect=.9)
(g.set_axis_labels("", "Price ranges on room_types")
  .set_xticklabels(["Private room", "Entire home/apt", "Shared room"],rotation=90)
  .set(ylim=(0, 400))
  .despine(left=True))
plt.show()
```

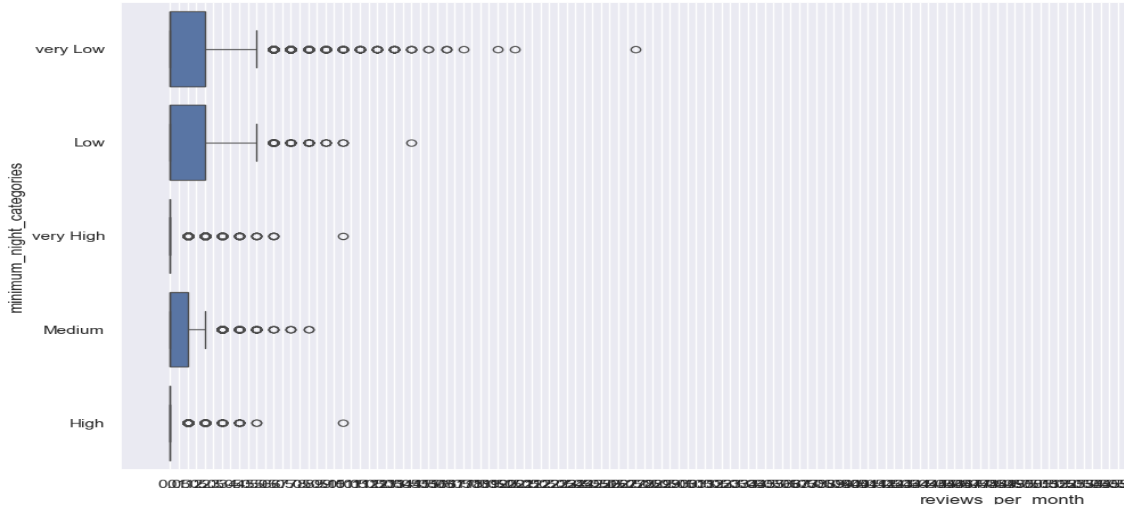


4. Analyzed variation in monthly reviews by room type across neighborhood groups :

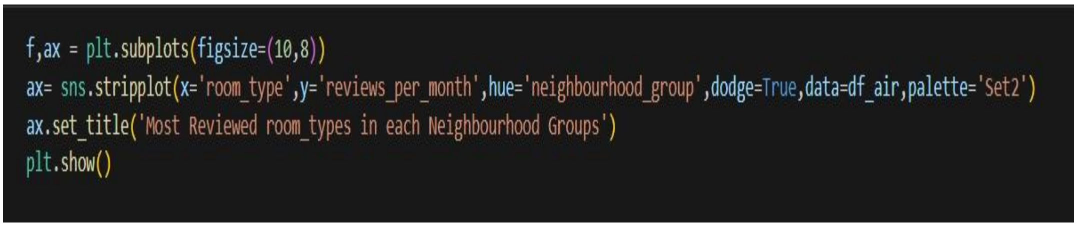
- We created boxplot (Minimum Night Categories vs. Reviews per Month) which illustrates how the frequency of reviews is distributed across different minimum night stay categories, indicating whether longer or shorter stays attract more reviews.



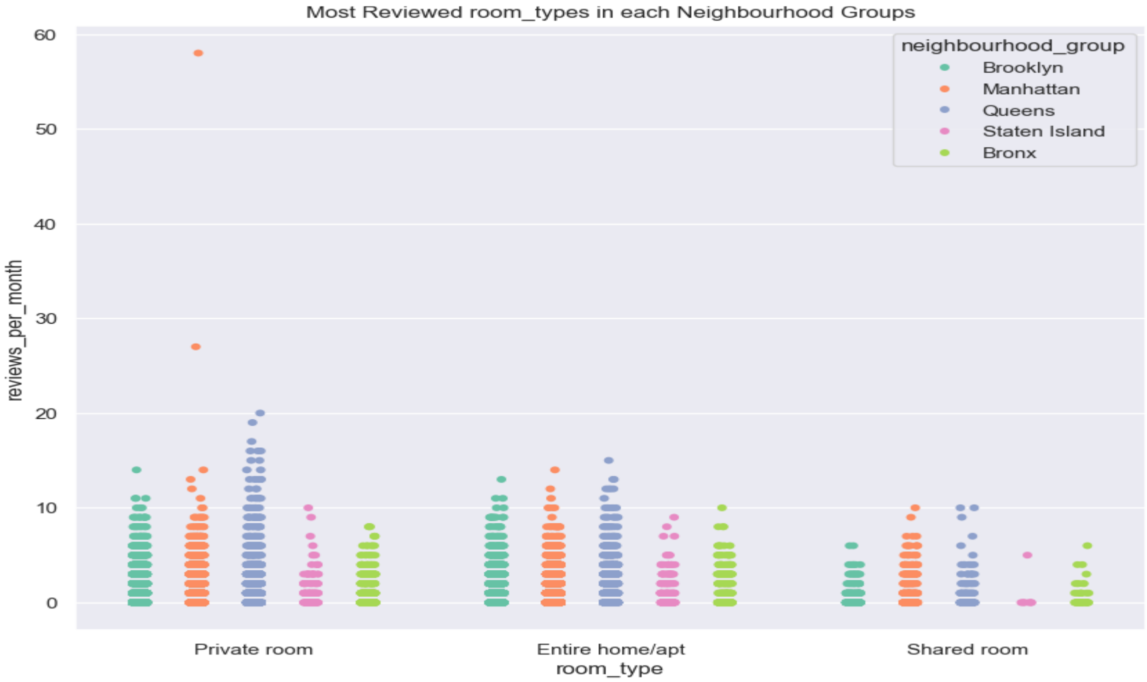
```
plt.figure(figsize=(20,8))
sns.boxplot(data = df_air_pnw, y = 'minimum_night_categories' ,x = 'reviews_per_month')
plt.xticks(np.arange(0,100,.5))
plt.show()
```



- We plotted strip which shows the relationship between room type and review frequency, with colour coding highlighting how this varies across neighbourhood groups.



```
f,ax = plt.subplots(figsize=(10,8))
ax= sns.stripplot(x='room_type',y='reviews_per_month',hue='neighbourhood_group',dodge=True,data=df_air,palette='Set2')
ax.set_title('Most Reviewed room_types in each Neighbourhood Groups')
plt.show()
```


Most Reviewed room_types in each Neighbourhood Groups
reviews_per_month
Private room
Entire home/apt
Shared room
room_type
neighbourhood_group
● Brooklyn
● Manhattan
● Queens
● Staten Island
● Bronx