

Methodology Air BNB Case Study IIIT-B

BY Ayush Kumar Singh, Tushar Prajapati, Tanmoy Bera.

In these case studies, we have seen the past few months, Airbnb has seen a major decline in revenue due to the lockdown imposed during the pandemic.

Also, we have seen that restrictions have started lifting and people have started to travel more. Hence, Airbnb wants to make sure that it is fully prepared for this change.

We have performed data cleaning and visualisation with tools.

- Python
- Tableau

These tools were used to perform some data cleaning and some graphical and visualisation chart.

1. PYTHON: -

In these Python, we have performed cleaning null values.

- First, we call all the necessary data for doing the data framework.

```
|: # importing all the necessary data
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

- Now we call the CSV file to check how many rows and column.

```
data1 = pd.read_csv('AB_NYC_2019.csv')
data1
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	numb
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
...
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70		2
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40		4
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115		10
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55		1
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90		7

48895 rows × 16 columns

You see that there are 48895 rows and 16 columns.

- Now we check the CSV file that how many different int file is present. And how much memory usage.

```
data1.info('all') #calling all the data information

<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
```

- Now we check how many nulls are present in the CSV file.

```
: data1.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
```

If u see that there are 4 different null types present in the CSV file.

- Now we will check what percentage of null values is present in the CSV file.

```
((data1.isnull().sum()*100) / len(data1)).round(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
dtype: float64
```

```
data1['last_review'].fillna(data1['last_review'].mode()[0],inplace=True)
data1['reviews_per_month'].fillna(data1['reviews_per_month'].mode()[0],inplace=True)
```

- Cleaning all the data of percentage and filtering the data we get :-

```
((data1.isnull().sum()*100) / len(data1)).round(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  0.00
reviews_per_month  0.00
calculated_host_listings_count  0.00
availability_365  0.00
dtype: float64
```

```
data1['name'].fillna(data1['name'].mode()[0],inplace=True)
data1['host_name'].fillna(data1['host_name'].mode()[0],inplace=True)
```

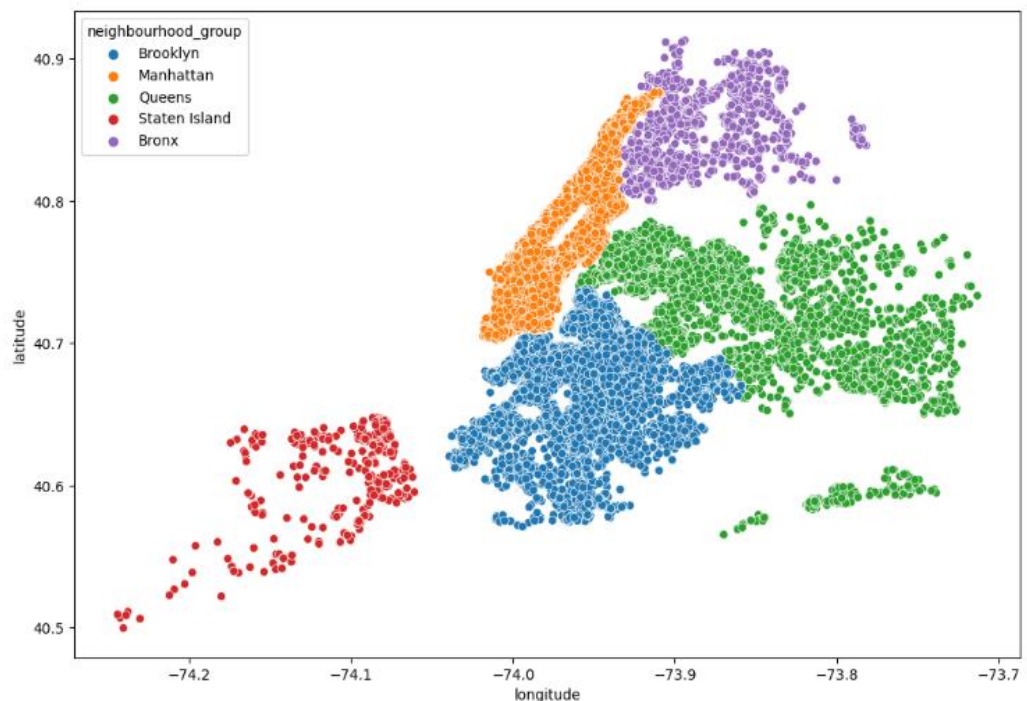
```
((data1.isnull().sum()*100) / len(data1)).round(2)
```

```
id          0.0
name        0.0
host_id     0.0
host_name   0.0
neighbourhood_group  0.0
neighbourhood  0.0
latitude    0.0
longitude   0.0
room_type   0.0
price       0.0
minimum_nights  0.0
number_of_reviews  0.0
last_review  0.0
reviews_per_month  0.0
calculated_host_listings_count  0.0
availability_365  0.0
dtype: float64
```

- Now we have performed all the data cleaning parts.
- At last in Python we have made one scatter plot for these for better-knowing data in region-wise.

```
# Lets check the overall map spread of airbnb :
```

```
plt.figure(figsize=(12,8))
sns.scatterplot(data1['longitude'],data1['latitude'],hue=data1['neighbourhood_group']);
```



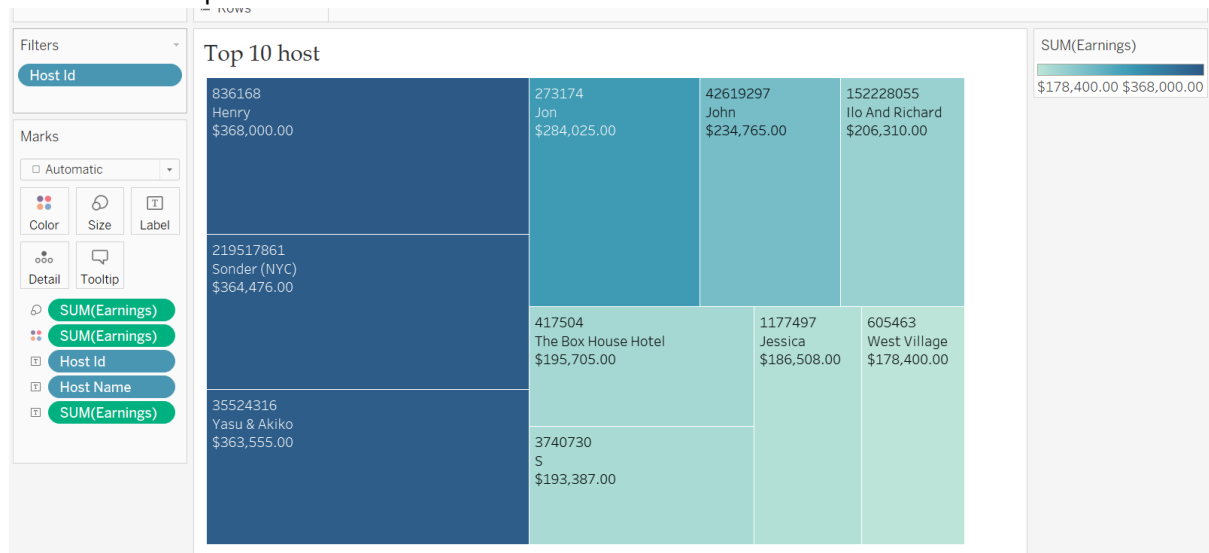
2. TABLEAU: -

In this tableau, we perform a visualisation chart of all the necessary diagrams we put here, so on that basic of charts u will know what actual data is going to show how much Air BNB Case increases or decreases.

In this pdf, we only tell you about how we put data and how we present it in a tableau.

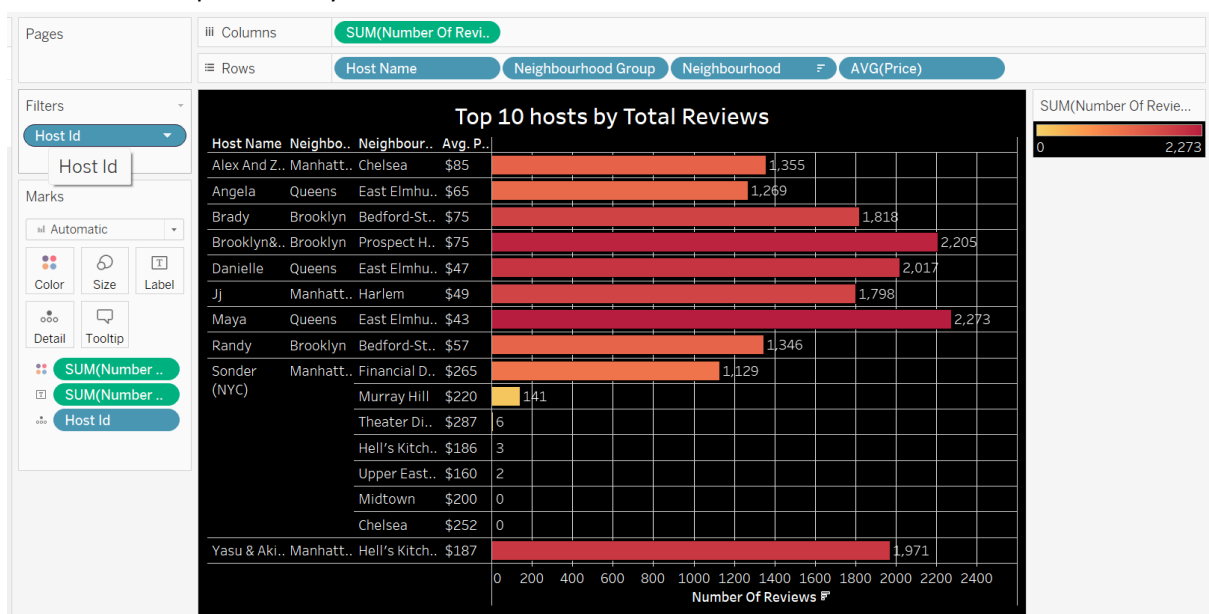
➤ TOP HOST

We identified the top 10 Host Ids, Host Names with a count of Host Ids using the tree map.



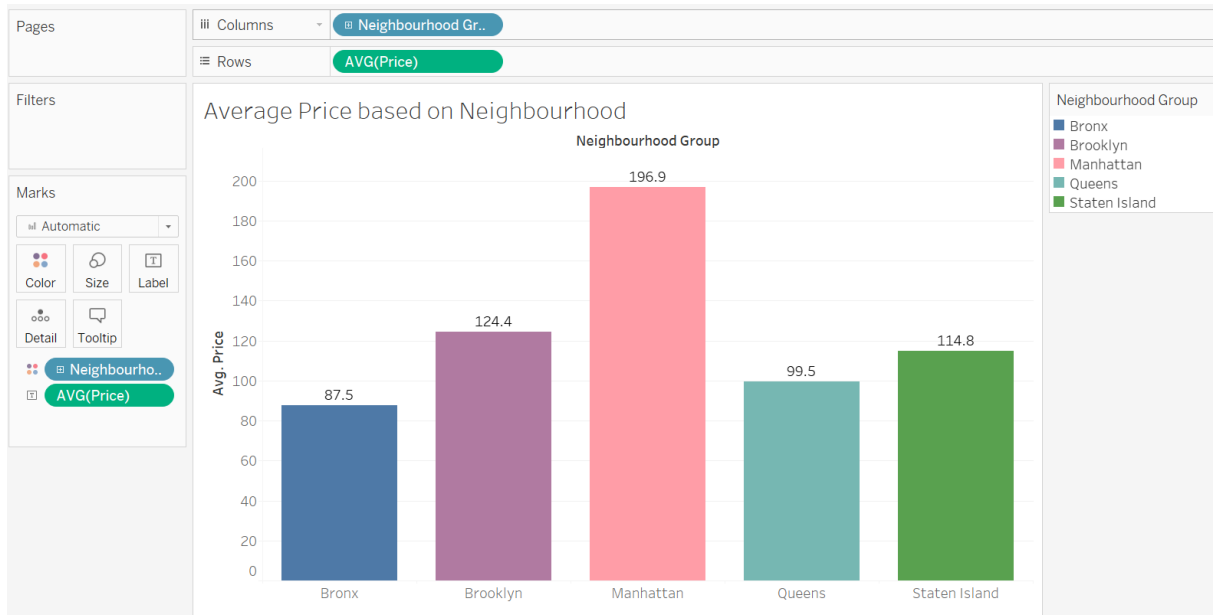
In the above image, we have done some filters in the Host id we put some filters and select a top and put a field by the top 10 only through filtering data and we get the top 10 host values.

➤ Top 10 host by review



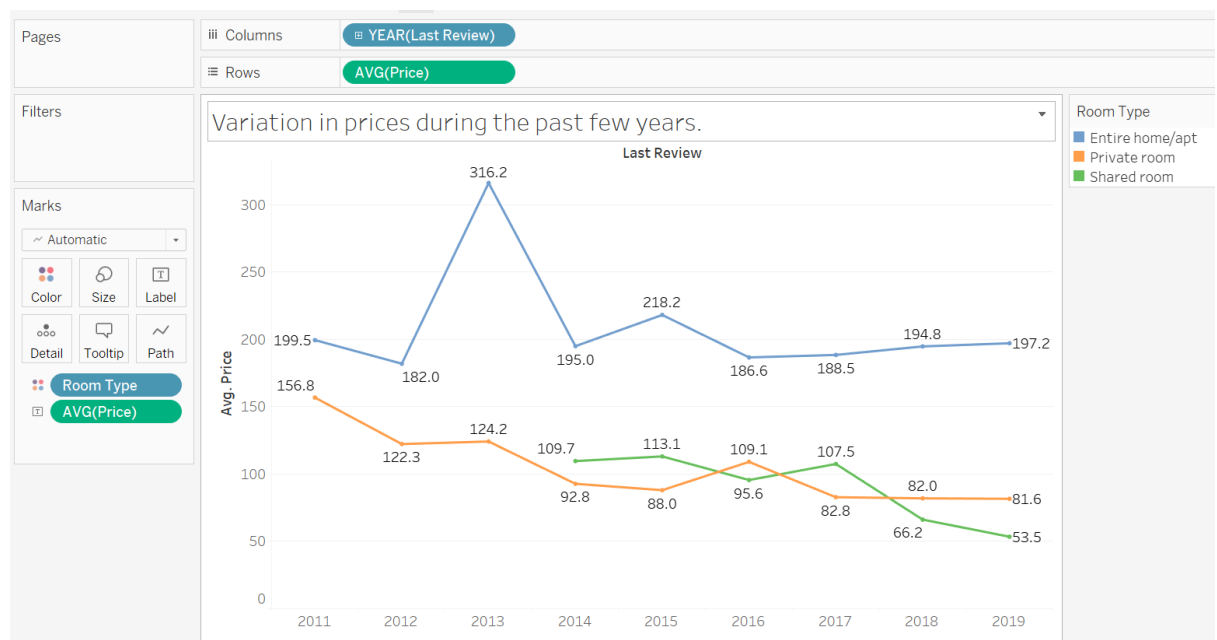
Same things but the graph visualisation is different in these we have shown in which region top neighbourhood, neighbourhood group and average price generate total reviews.

➤ **Average prices based on neighbourhood group.**



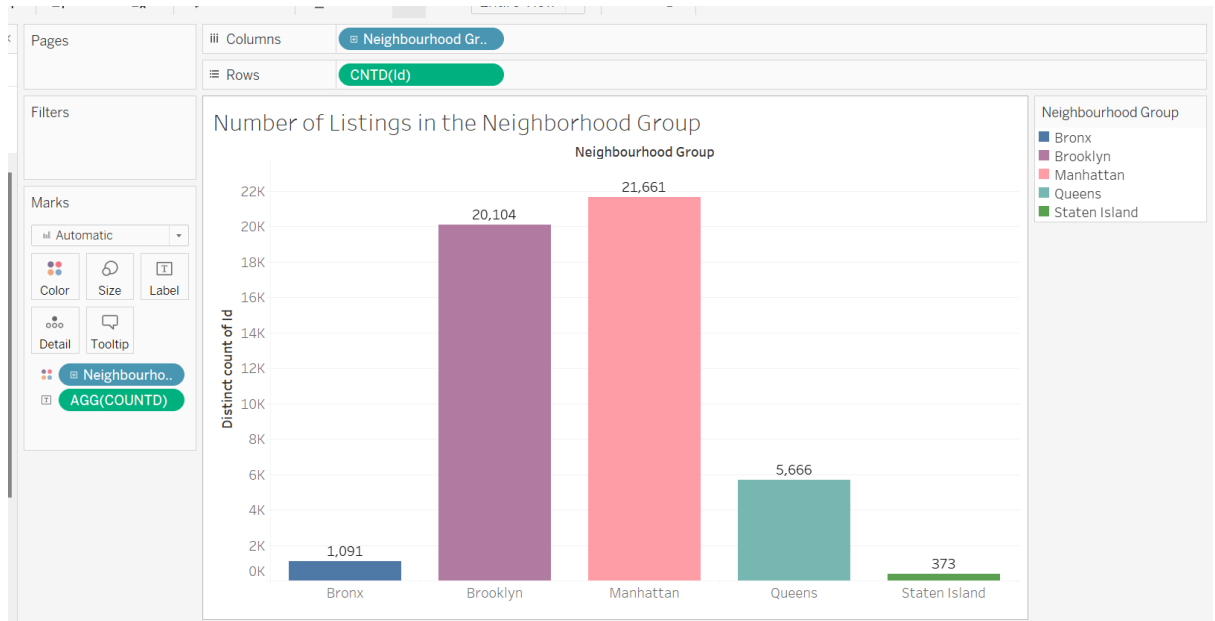
In these, we present an in which neighbourhood group presents an Average price related to state.

➤ **Variation in price during past few years.**



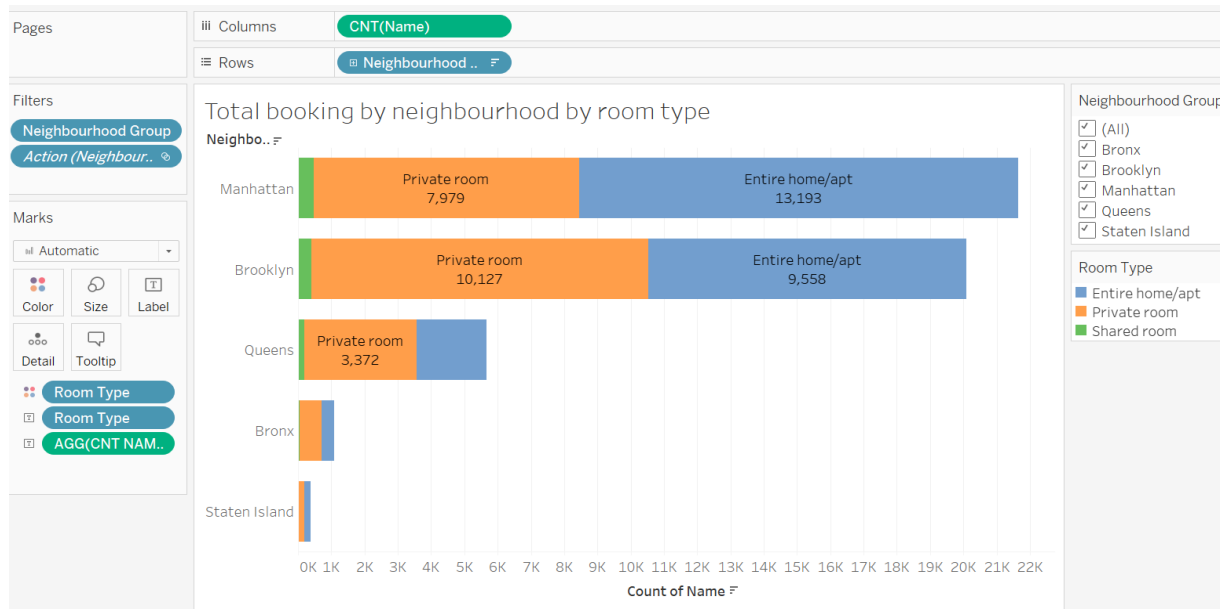
In these charts, we present the prices in the past few years. In these, we put years wise in columns and Average price in rows, colours in room type so that we can see a variation easily and price (avg) in detail.

➤ Number of listings in the neighbourhood group :-



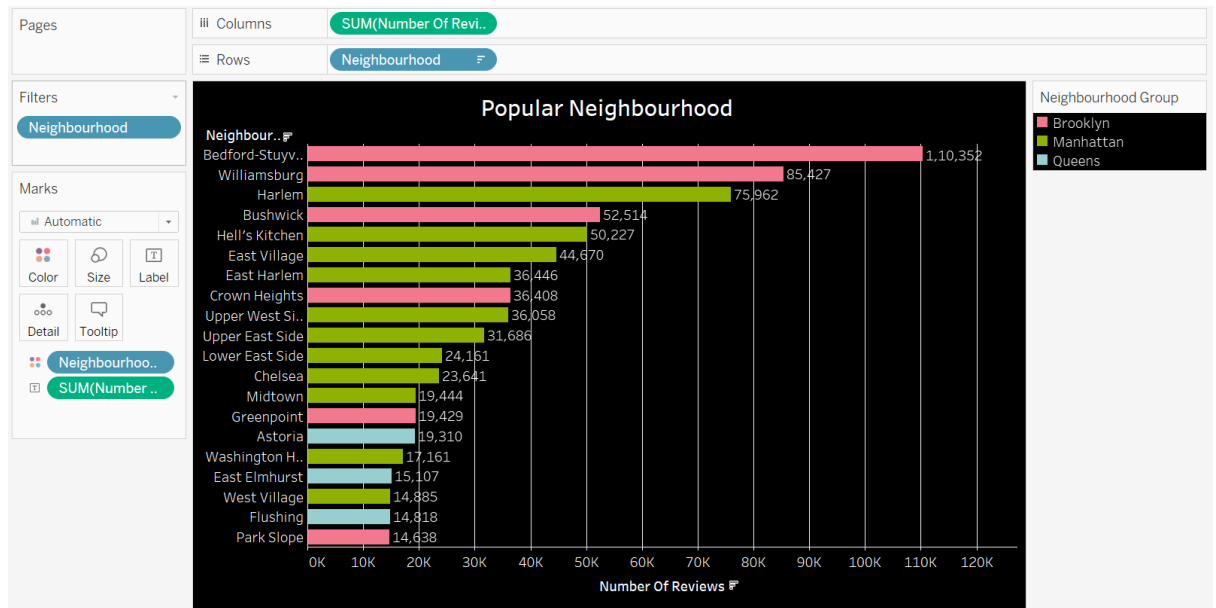
In this chart we put the id in rows and then double-clicking on the id we make a count of the id then the neighbourhood group is put in columns the graph will present in a bar chart and then colour in the neighbourhood group. Label in the count of id for showing which part is highest in the neighbourhood by id.

➤ Total Bookings in Neighbourhood by room type.

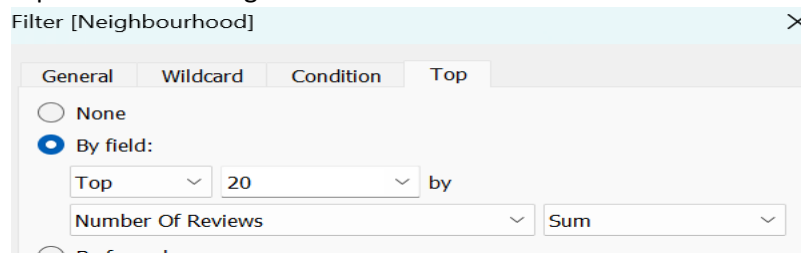


In these charts, we make a count of names in columns and neighbourhood groups in rows. In the filter, we have done in general click on use all and apply in neighbourhood groups and colour in room type so that we easily present a which count of name using which type of rooms and label in room types and label in count name.

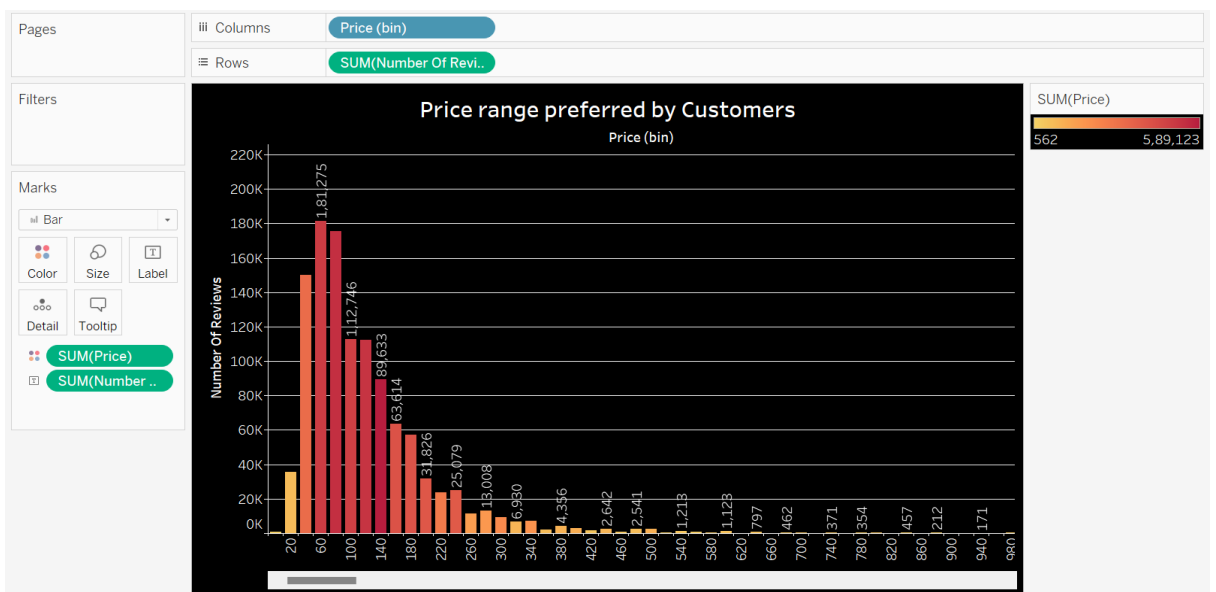
➤ Popular neighbourhood groups :-



In this chart, we put the number of reviews as the sum in columns and the neighbourhood in rows. In the filter part we have done neighbourhood we just put top 20. Color in neighbourhood and label it in Sum of number of reviews



➤ Popular Price Range Preferred by Customers.



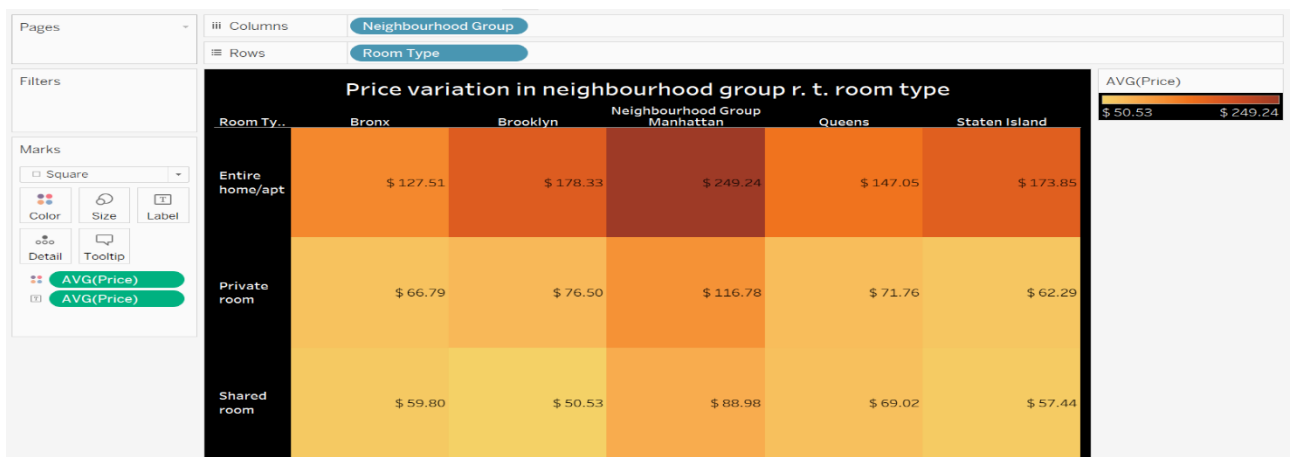
In this chart, we put the price of the bin in columns and the sum of the number of reviews in rows. The colour is the sum of prices so we should see a colour variation of prices related to customer reviews.

➤ **Popular properties as per reviews.**



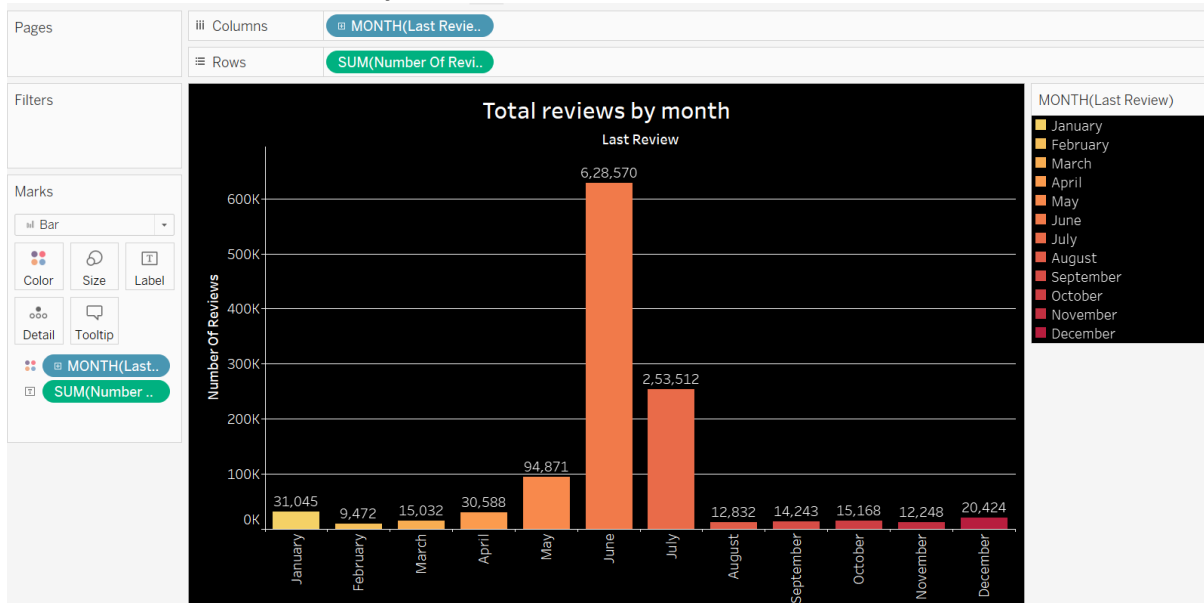
In these charts, we put the sum number of reviews in columns and name in rows. In filter, we have put a custom value list so unknown name not shown.

➤ **Variation of price with respect to neighbourhood group & room type :-**



In these, we select a neighbourhood group in columns and a room type in rows. The average price in colours for better seeing which rooms have high value booked. And the Average of price in label.

➤ Total reviews by month :-



In these charts we put the last review in the column as mentioned in months by clicking the right mouse we get a dialogue box to appear and then select months. In rows, we have put the sum of the number of reviews.

Year	2015
Quarter	Q2
• Month	May
Day	8
More	▸

We drag the month of the last review in colour to a better showing in which month the highest reviews got. And label the sum of several reviews.

- So that is all but you will get all these things in our ppt file
- There are two ppt files which are as follows:-
- ❖ PPT 1:-
 - This PPT 1 is related to our customer in that file what is needed in customer preference and what is customer reviews on the basic of room type, and room availability.
 - Understand customer preference and customer experience in Airbnb listings.
 - Improve business strategies and estimate customer preferences to revive the business and generate more revenue in the post-COVID period.
- ❖ PPT 2:-
 - This PPT 2 is related to our reviews based on customers how they stay enjoyment in their respective rooms.
 - Airbnb can concentrate on promoting shared rooms with targeted discounts to increase bookings.

END

THANK YOU