Storytelling Case Study: Airbnb, NYC

Authored By :

Chandra Deepak
Parag Behera
Jayant Singhal
Shubhranshu Shekhar Dash

Storytelling Case Study: Airbnb, NYC

Problem Statement:

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.

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

As a data analyst at Airbnb, we would need to understand the past business (provided in a dataset) and provide suggestion to various stakeholders.

Target Audience I:

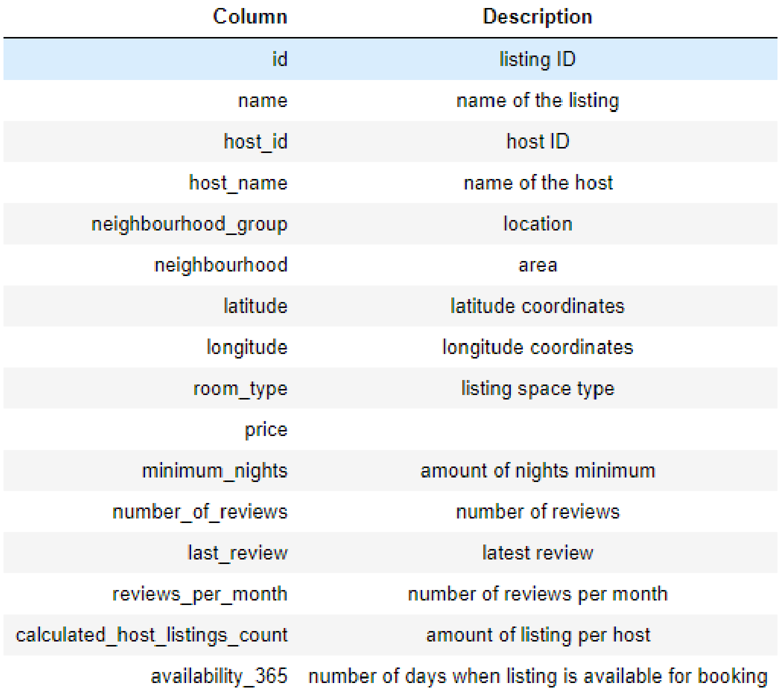
- Data Analyst Manager
- Lead Data analyst

Target Audience II:

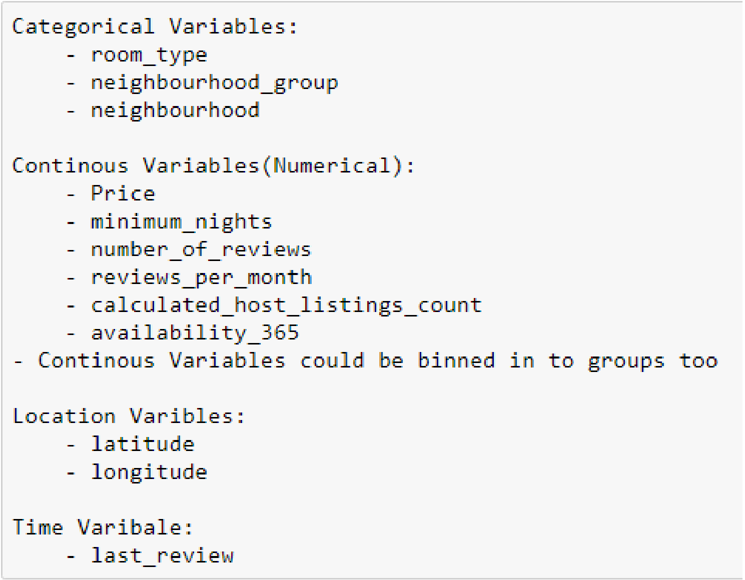
- Head of Acquisitions and Operations, NYC
- Head of User Experience, NYC

Understanding on the data set.

We had analyzed the “Newyork Airbnbs Dataset” which was provided in the case study with the below details.



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



```
Categorical Variables:
    - room_type
    - neighbourhood_group
    - neighbourhood

Continous Variables(Numerical):
    - Price
    - minimum_nights
    - number_of_reviews
    - reviews_per_month
    - calculated_host_listings_count
    - availability_365
- Continous Variables could be binned in to groups too

Location Varibles:
    - latitude
    - longitude

Time Varibale:
    - last_review
```

Working on the Dataset

We used the python coding to understand on:

1. Have a look on the dataset.
2. Understand on the datatypes of the data.
3. Check the NULL values in the data and the find the %age of Null values to decide what should be done in each case where we have the NULL data.
4. Data cleaning for the NULL value data.
5. Prepare the dataset (in excel format) to feel into the analytics tool.

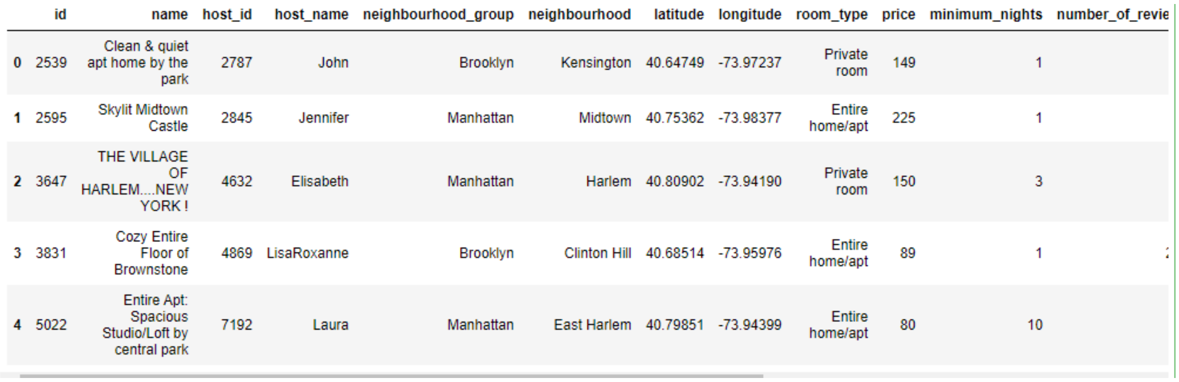
We used Tableau for the data analysis and find insights:

1. Univariate analysis.
2. Bivariate analysis.
3. Multivariate analysis.
4. Maps
5. Other inferences.

We will go through the entire process of the analysis in sequence

1. Python analysis on - look on the dataset

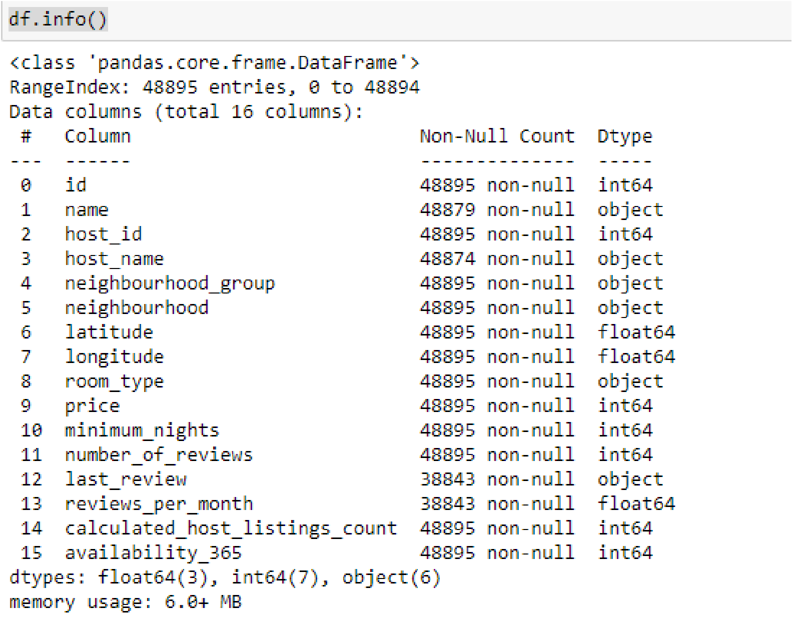
df = pd.read_csv("AB_NYC_2019.csv")
df.head()



	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	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

At his step we tried to check what are the data present and how do the dataset looks like. This is just get a feel before we step into the real data cleaning and manipulation.

2. Python analysis on – understand the datatypes:

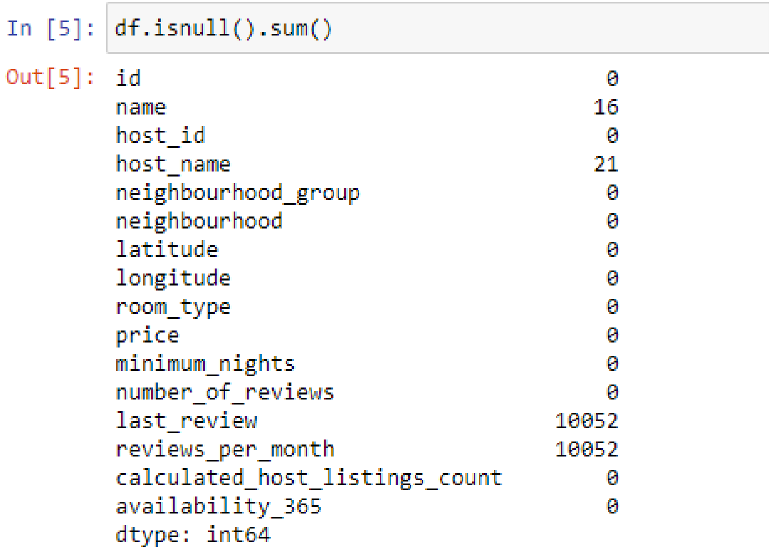


```
df.info()
```

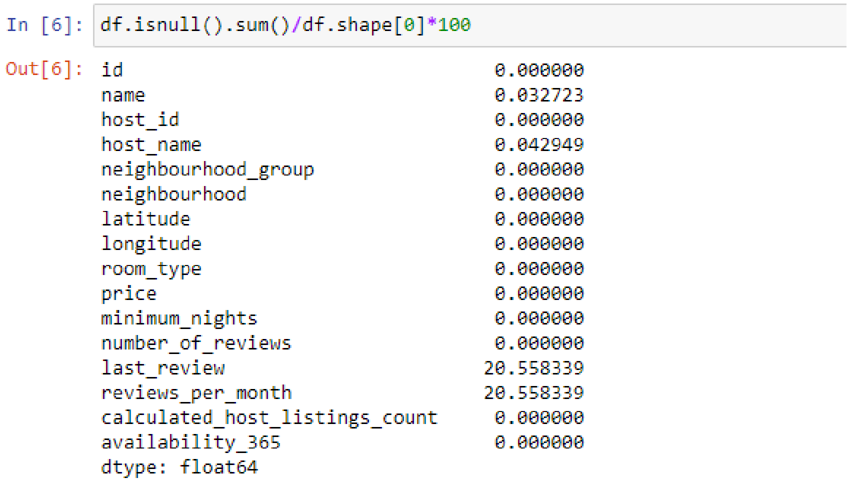
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   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
```

We could see here is that all the data types are correct and no futher change or manupulation of the data types are required. Hence we didn’t proceed with the change of any data types.

3. Python analysis on – Null value check:



```
In [5]: df.isnull().sum()

Out[5]: 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
```



```
In [6]: df.isnull().sum()/df.shape[0]*100

Out[6]: 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
```

From the NULL value check , we could see that out of all the fields given for the analysis, only below 4 columns had NULL values.

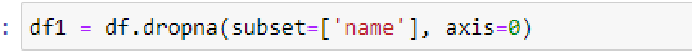
Name
Host Name
Last Review
Reviews per month

4. Python analysis on –data cleaning:

Name:

Only 0.03% of the data set had the NULL values. This might have been error during the capturing of the data. We felt that this can be ignored during analysis.

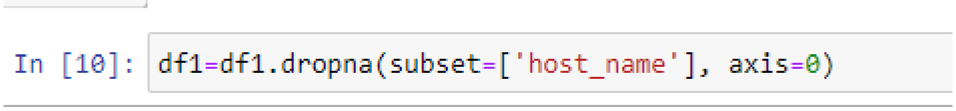
Hence, we decided to delete the rows with Null values.



```
: df1 = df.dropna(subset=['name'], axis=0)
```

Host_name :

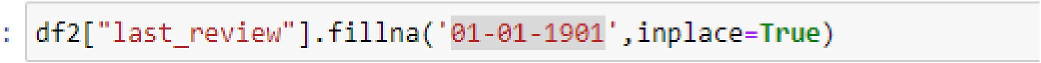
Only 0.03% of the data set had the NULL values. This might have been error during the capturing of the data. We felt that this can be ignored during analysis.
Hence, we decided to delete the rows with Null values.



```
In [10]: df1=df1.dropna(subset=['host_name'], axis=0)
```

Last_review:

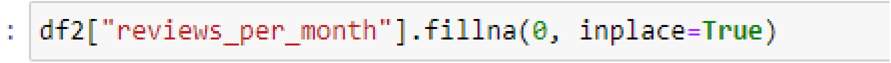
The last_review column had around 20% null value data. We felt that we should not delete the data with NULL values as might lose out some important data from other columns. Hence we decided to assign a date "01-01-1901" , a different date from the dataset date provide to identify that these were the NULL values.



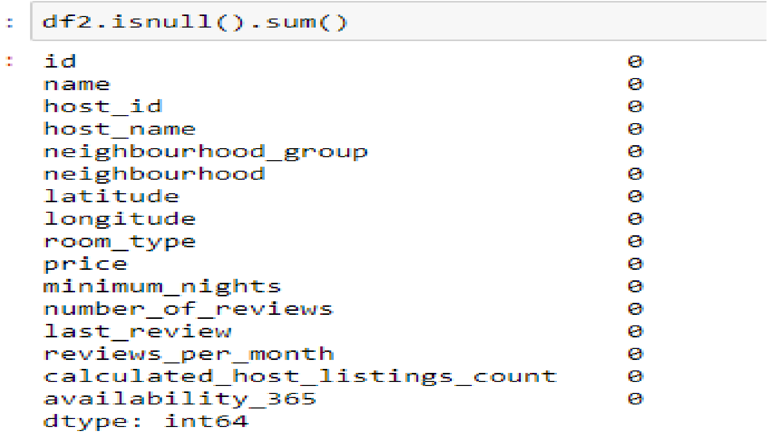
```
: df2["last_review"].fillna('01-01-1901',inplace=True)
```

reviews_per_month

The reviews_per_month column had around 20% null value data. We felt that we should not delete the data with NULL values as might lose out some important data from other columns. Hence we decided to assign a 0 to the NULL values.



```
: df2["reviews_per_month"].fillna(0, inplace=True)
```



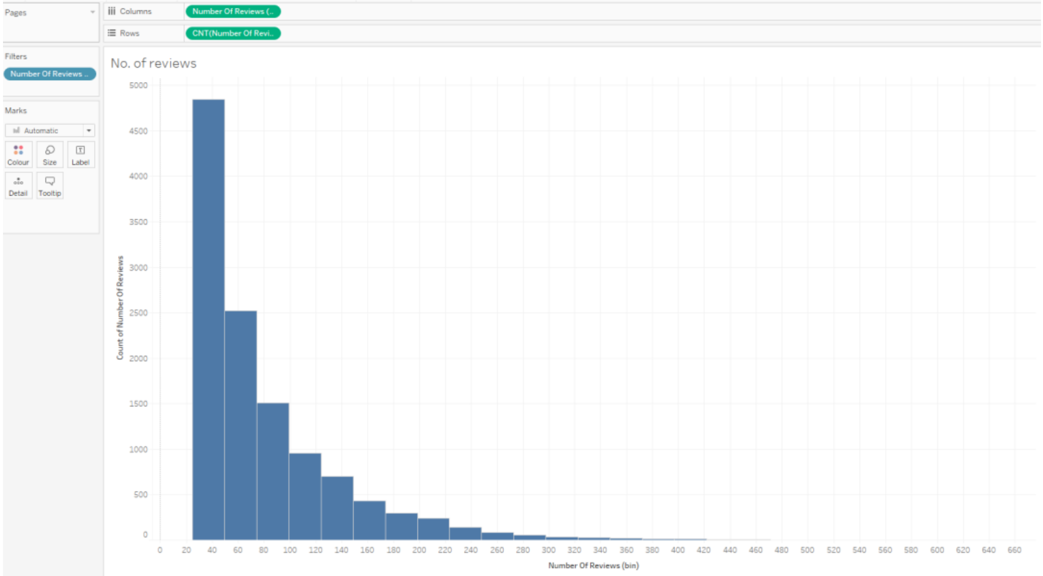
```
: df2.isnull().sum()

: id                                0
  name                              0
  host_id                           0
  host_name                         0
  neighbourhood_group               0
  neighbourhood                     0
  latitude                          0
  longitude                         0
  room_type                         0
  price                             0
  minimum_nights                    0
  number_of_reviews                 0
  last_review                       0
  reviews_per_month                 0
  calculated_host_listings_count    0
  availability_365                  0
  dtype: int64
```

5. Python analysis on –exporting cleaned dataset into excel:

The cleaned dataset was then exported to excel so that it could be used by Tableau (used in our case study) to analyze the data and trend.

TABLEAU ANALYSIS

We have done various analysis in tableau to find the different insights.



Columns: Number Of Reviews (..

Rows: CNT(Number Of Revi..

Filters: Number Of Reviews ..

Marks: Automatic

Fig 1. Insight of univariate analysis on the number of reviews and its count.

We excluded the 0 which had no significance. This figure tell us maximum “Host ID” has 25 reviews and the number of reviews are lesser for rest of the properties. If we consider the reviews, we can see that reviews were more in 0 values as compared to other values.

Hence we need to encourage customers to roved more feedback and provide appropriate reviews to help other customer to narrow down the search.

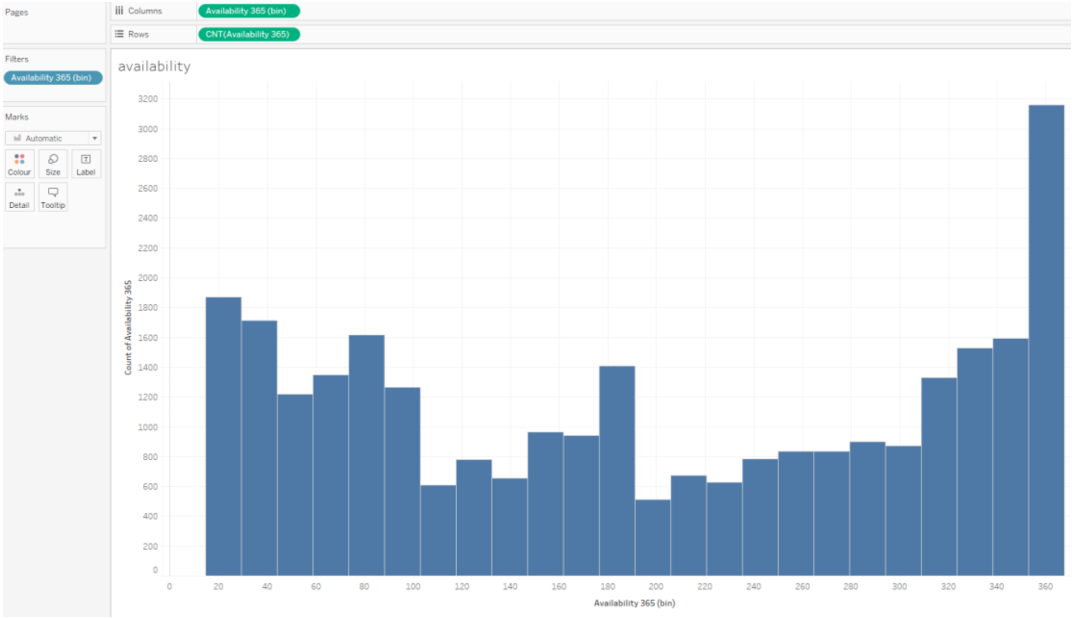

Columns: Availability 365 (bin)
Rows: CNT(Availability 365)
Filters: Availability 365 (bin)
Marks: Automatic

Fig 2: Availability of properties around the year.

From this we could see that :

- Many of the properties are listed for 0 days. Airbnb can take action for those properties' availability
- Most of the properties has 360 days availability.

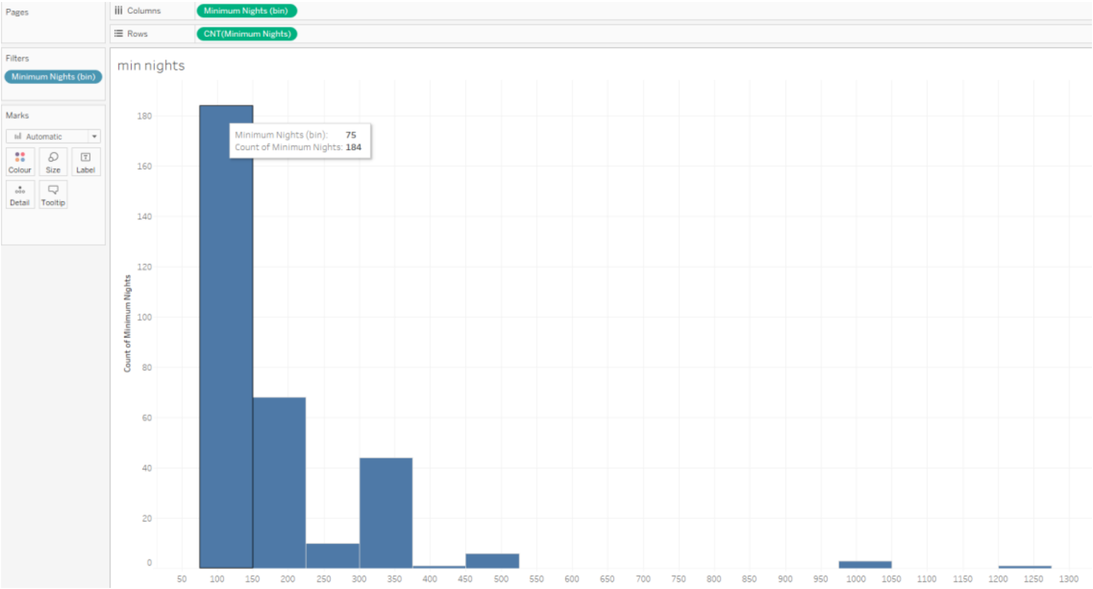
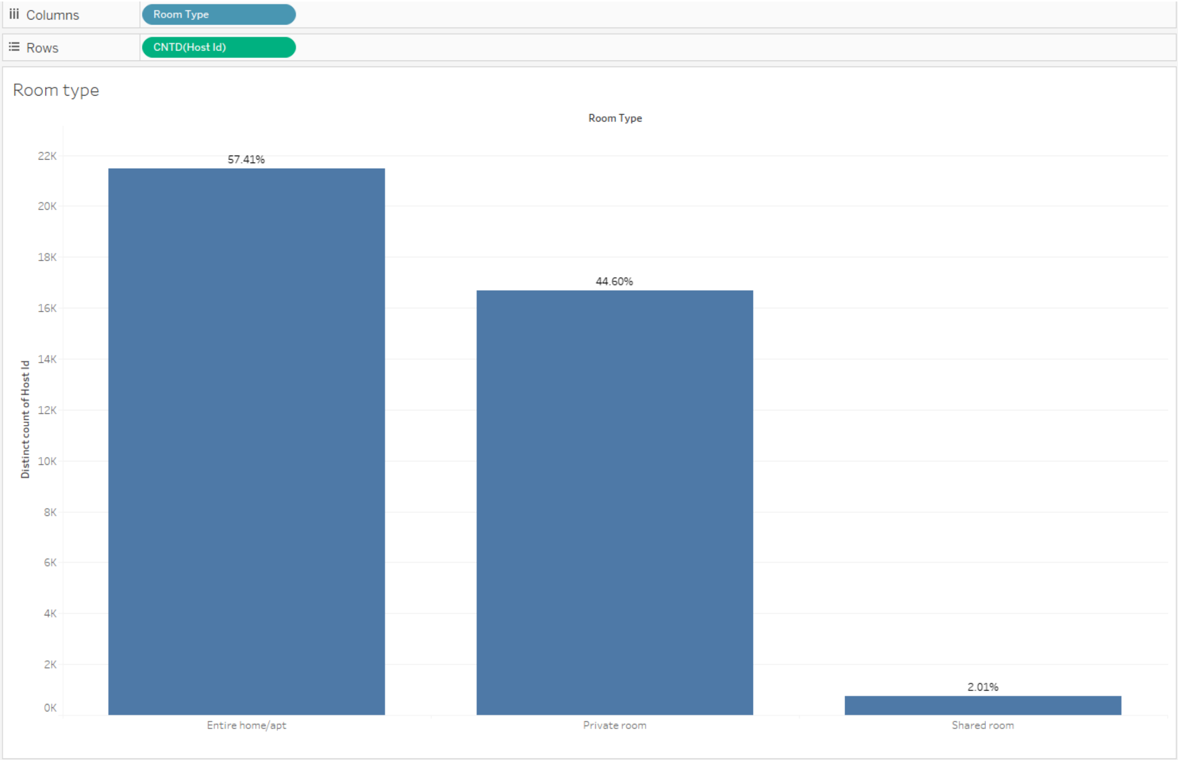

Columns: Minimum Nights (bin)
Rows: CNT(Minimum Nights)
Filters: Minimum Nights (bin)
Marks: Automatic
Minimum Nights (bin): 75
Count of Minimum Nights: 184

Fig 3: Minimum nights – Univariate analysis

From this figure, we could see that :

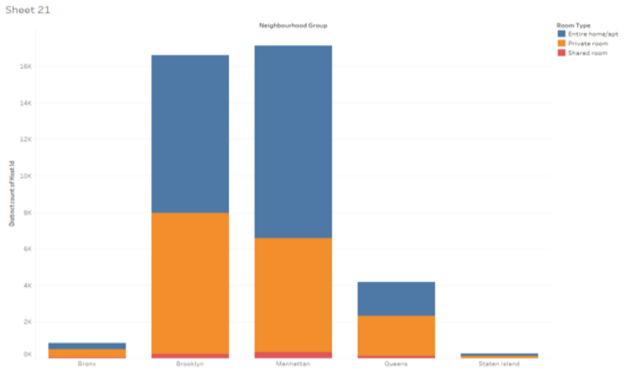
- Properties listed with around 75 night are the most preferred ones.

- And then most preferred properties are the ones which are available for 150 and 300 nights.
- Rest of the properties are not much preferred by customers.

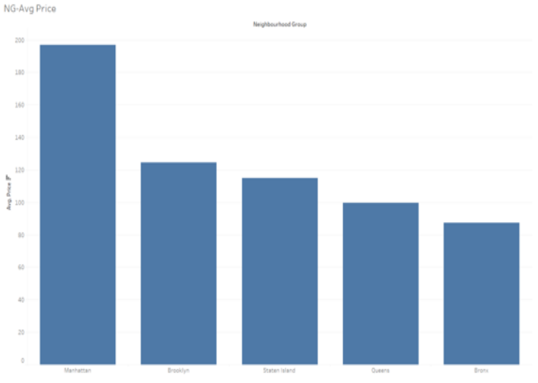


Columns: Room Type
Rows: CNTD(Host Id)

- “Entire home/apt” and “Private rooms” remain the most preferred room types which constitutes of 98% of occupancy.
- Shared rooms are preferred less (just 2% of the total booking)



Distinct count of Host Id for each Neighbourhood Group. Colour shows details about Room Type.

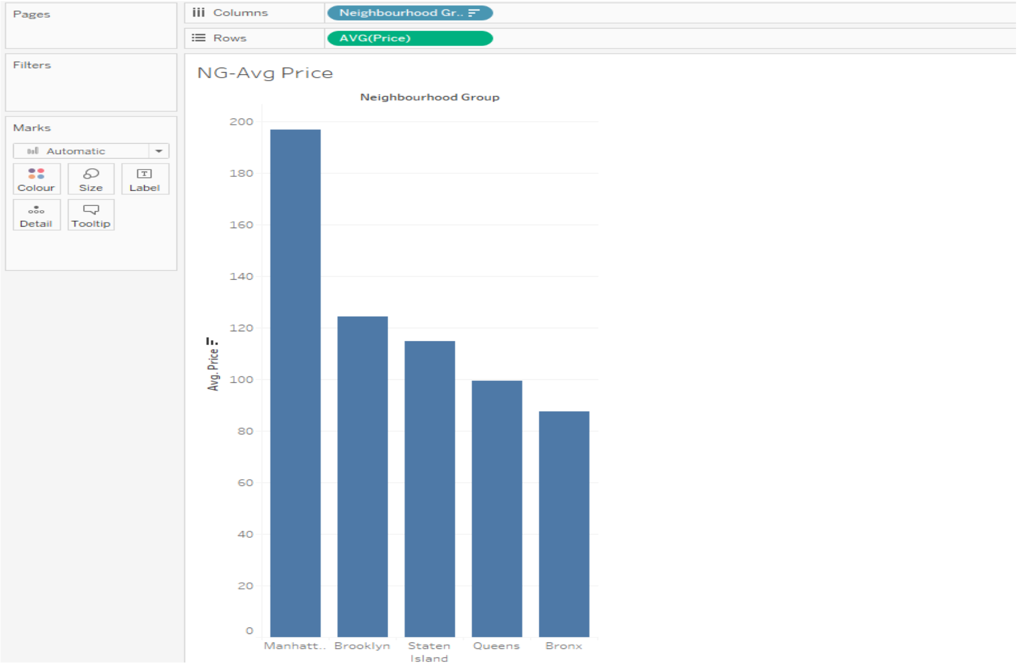


Average of Price for each Neighbourhood Group.

❑ Manhattan has the highest average price for all types of room types.

❑ Entire home and Private rooms are most frequently booked for all regions.

❑ The average prices of the properties are least in Bronx.



Columns: Neighbourhood Gr..

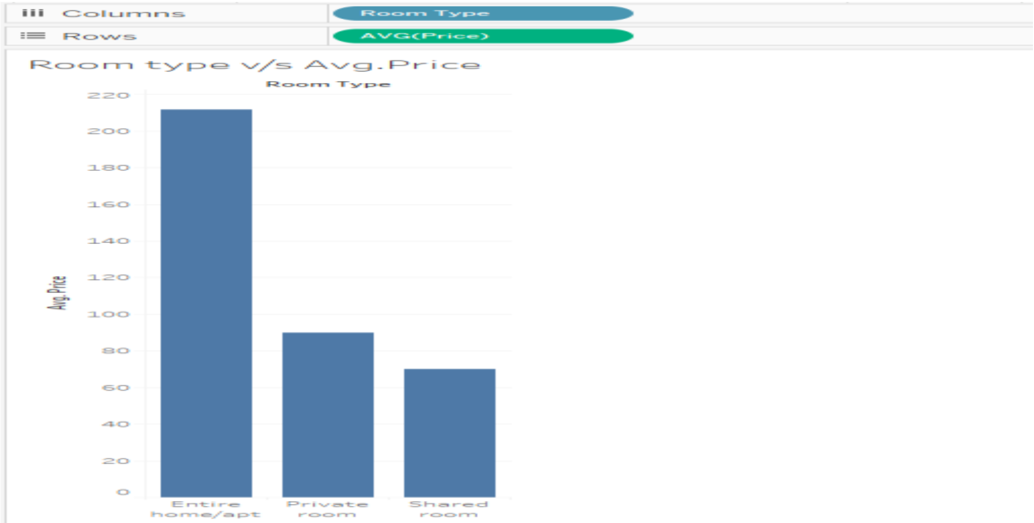
Rows: AVG(Price)

Fig 4 : Bivariate – Neighbourhood group v/s average price

We could see that :

■ Manhattan has the highest average price as compared to other areas.

■ Bronx has the minimum average price



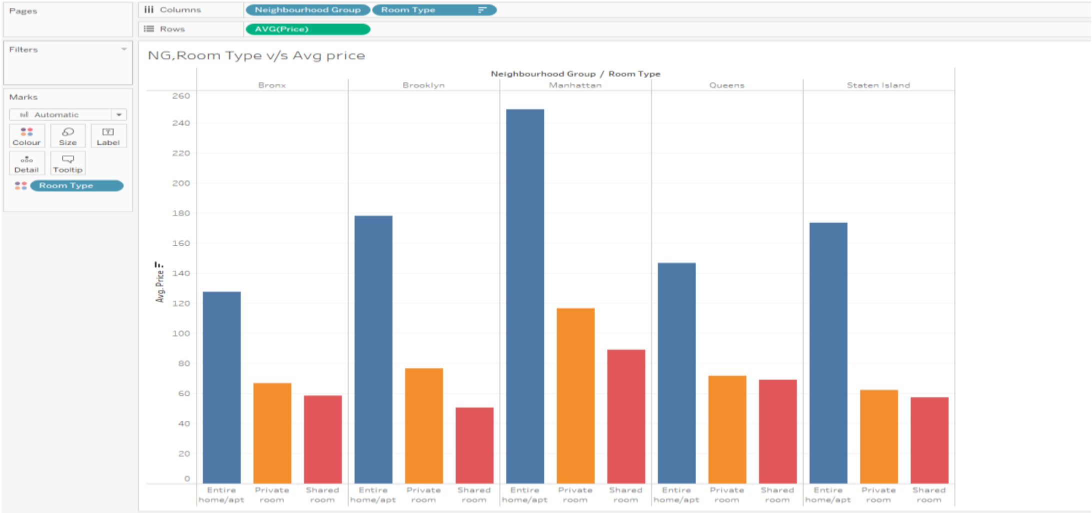
Columns: Room Type

Rows: AVG(Price)

Fig 5: room type v/s average price

Based on the analysis on the room type in accordance with the price, we can infer that:

- Entire home/apartment are the costliest option.
- Then we have the private room and shared rooms are the least costliest option for customers.



Columns: Neighbourhood Group, Room Type

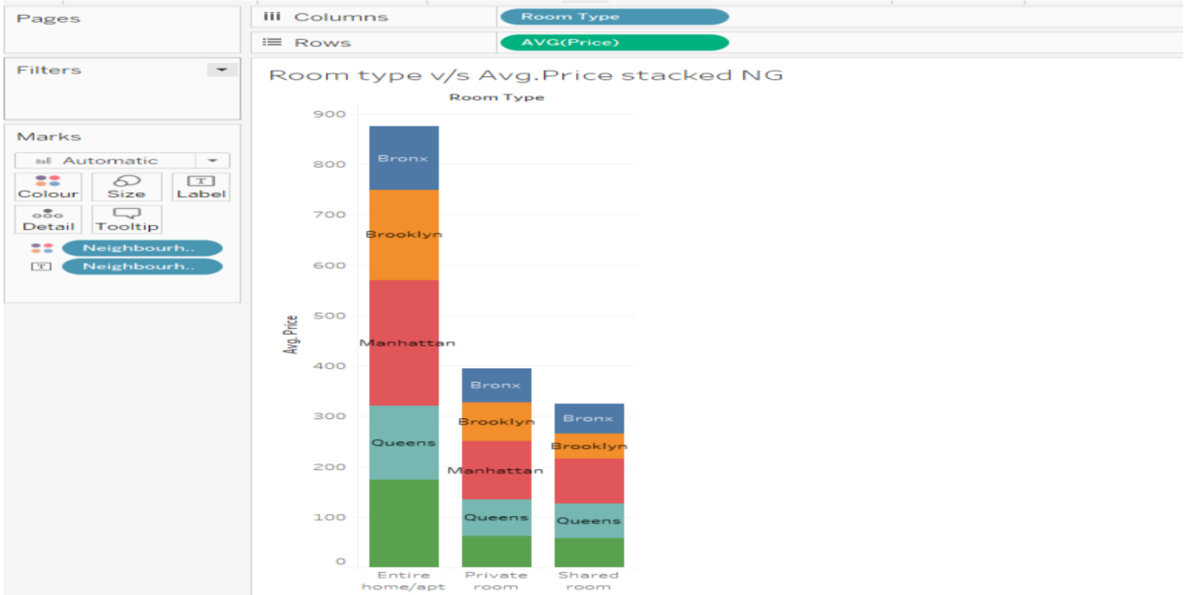
Rows: AVG(Price)

Marks: Automatic; Colour: Room Type

Fig 6 : Multivariate – Neighborhood group v/s room type v/s average price

This analysis clearly tells us :

- If entire home/apartment is rented out , then this is the most costly properties. This holds true for all the neighborhood groups.
- Private rooms are second costly and shared rooms are the least cost options for the customers.



Columns: Room Type

Rows: AVG(Price)

Marks: Automatic; Colour: Neighbourh..; Label: Neighbourh..

Fig 7: Room type v/s average price

From this analysis, we can see that :

- Properties listed in Manhatted constitutes the maximum price.
- This means that customer has to really think before they go for a longer vacation into those areas.
-

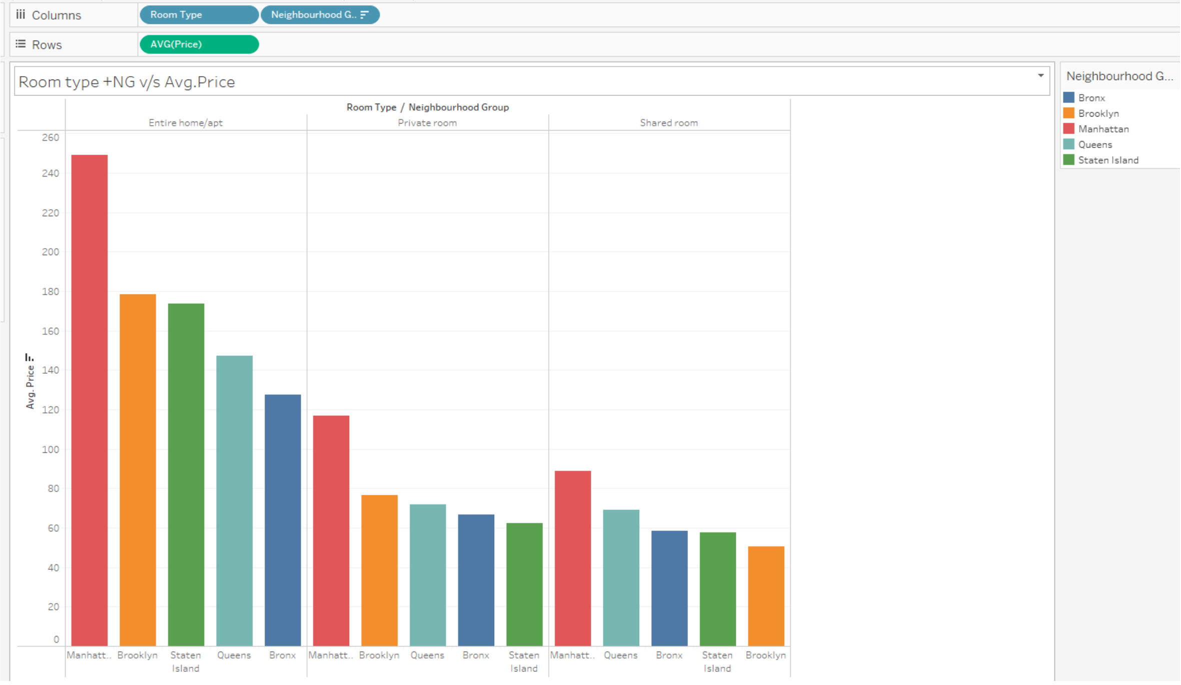


Fig 8: room type v/s neighborhood group v/s average price

We wanted to check if the cost is similar based on the neighborhood group. So what we got to understand from the graph is that:

- Yes, all types of room has the highest cost in Manhattan.
- But the same doesn't holds true for private room or shared rom. Example, the private rooms in Brooklyn are second highest costlier after Manhattan but shared rooms are much cheaper (least average price among all neighborhood group in NYC)

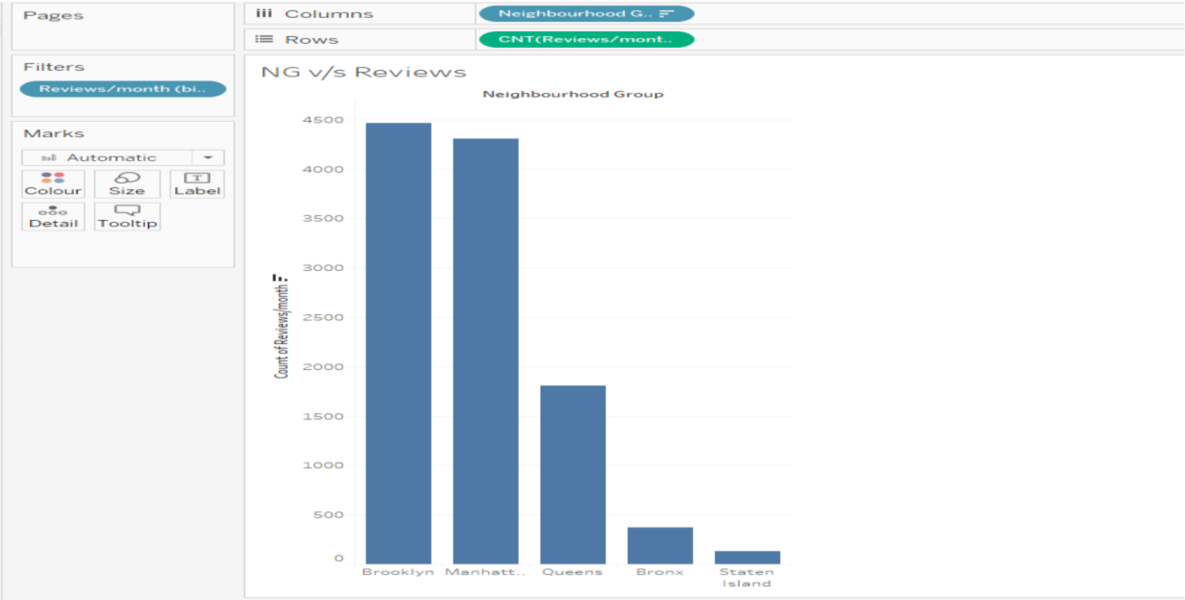


Fig 9: Neighborhood group v/s reviews

Taking into consideration, each review is positive, we can infer from the graph is that:

- Brooklyn is the most preferred location for stay and many customer has given positive feedbacks.
- Manhattan, even though it is the costliest area, is also one of the preferred location for stay in NYC.

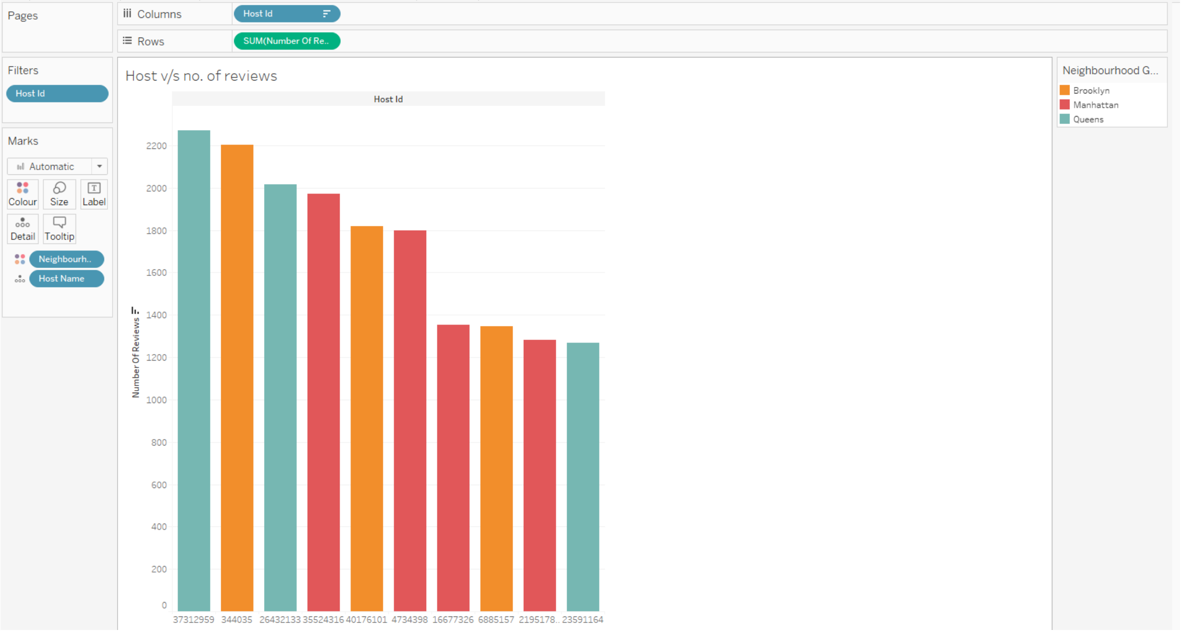
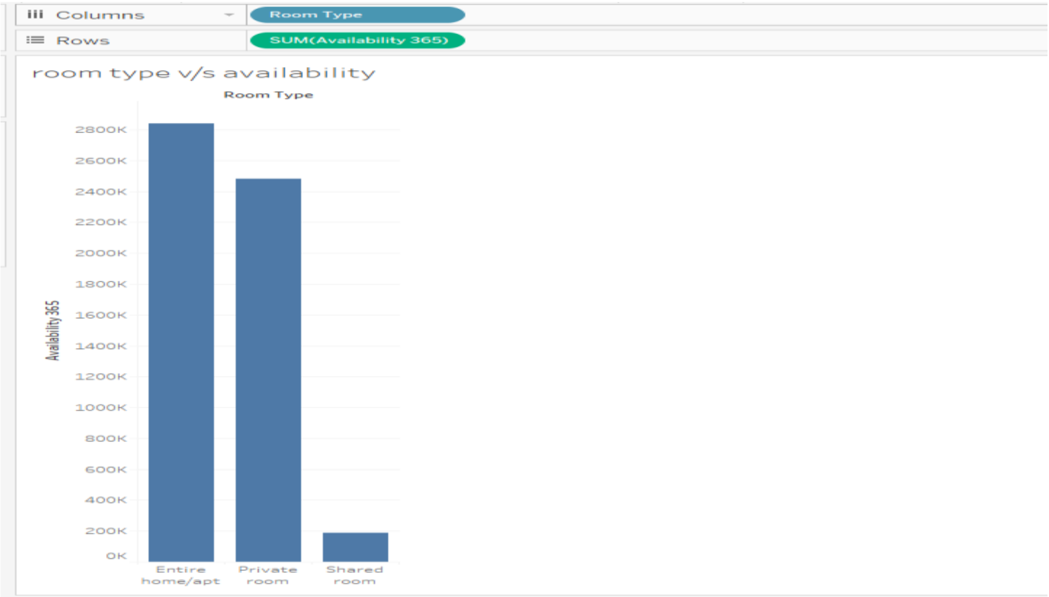


Fig 10: Top 10 hosts v/s reviews

We tried to check the top 10 host which has highest number of reviews. We could find the analysis is that :

- Queens, Brooklyn and Manhattan has the hosts which has maximum (top 10) review counts.
- Staten Island and Bronx has lesser reviews. Meaning, might be people are not more interested to stay in those areas.



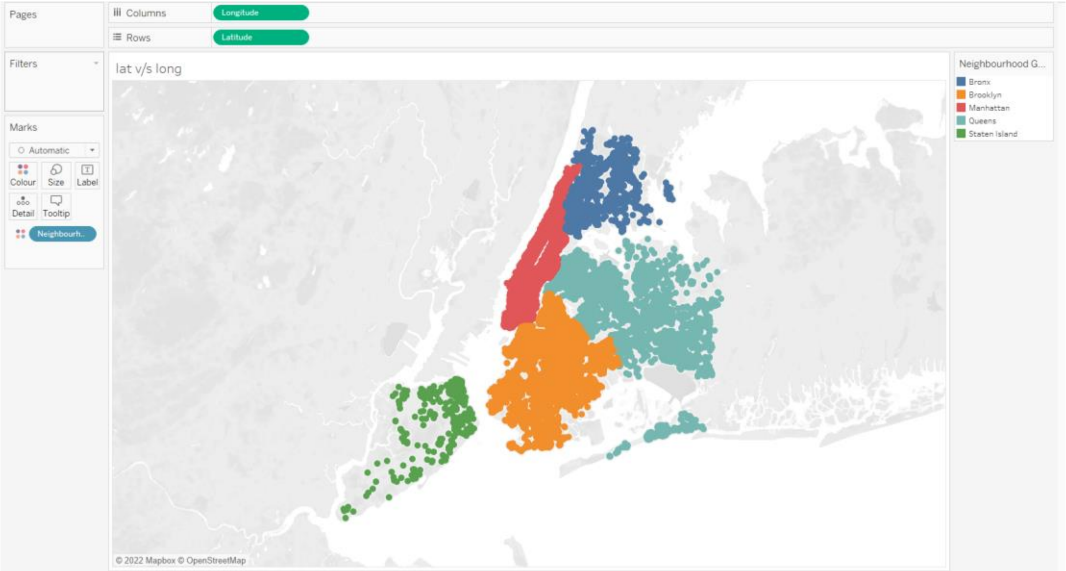
Columns: Room Type

Rows: SUM(Availability 365)

Fig 11: room type v/s availability

When we checked the room type v/s availability, we can see that :

- Entire home/apt are the most available properties.
- But looks like shared room are less available and considering the lesser cost, may be customer might look for it and Airbnb can plan to take more rooms for customers.



Columns: Longitude

Rows: Latitude

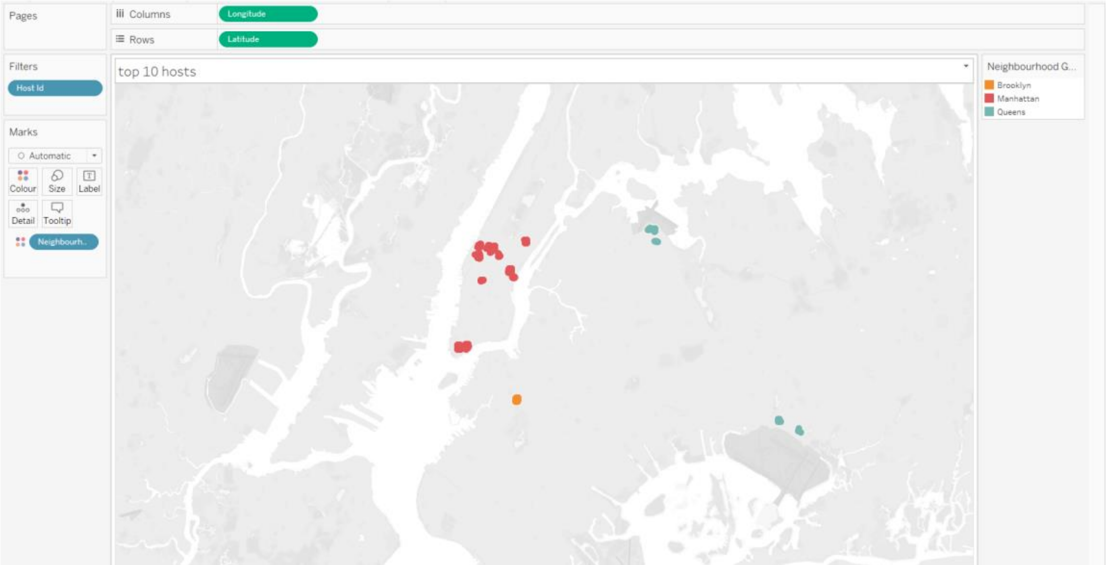


Fig 12: top 10 host

When we checked the location of the top 10 hosts, we can see that :

- Maximum host are present in Manhattan.
- Followed by Queens and then Brooklyn

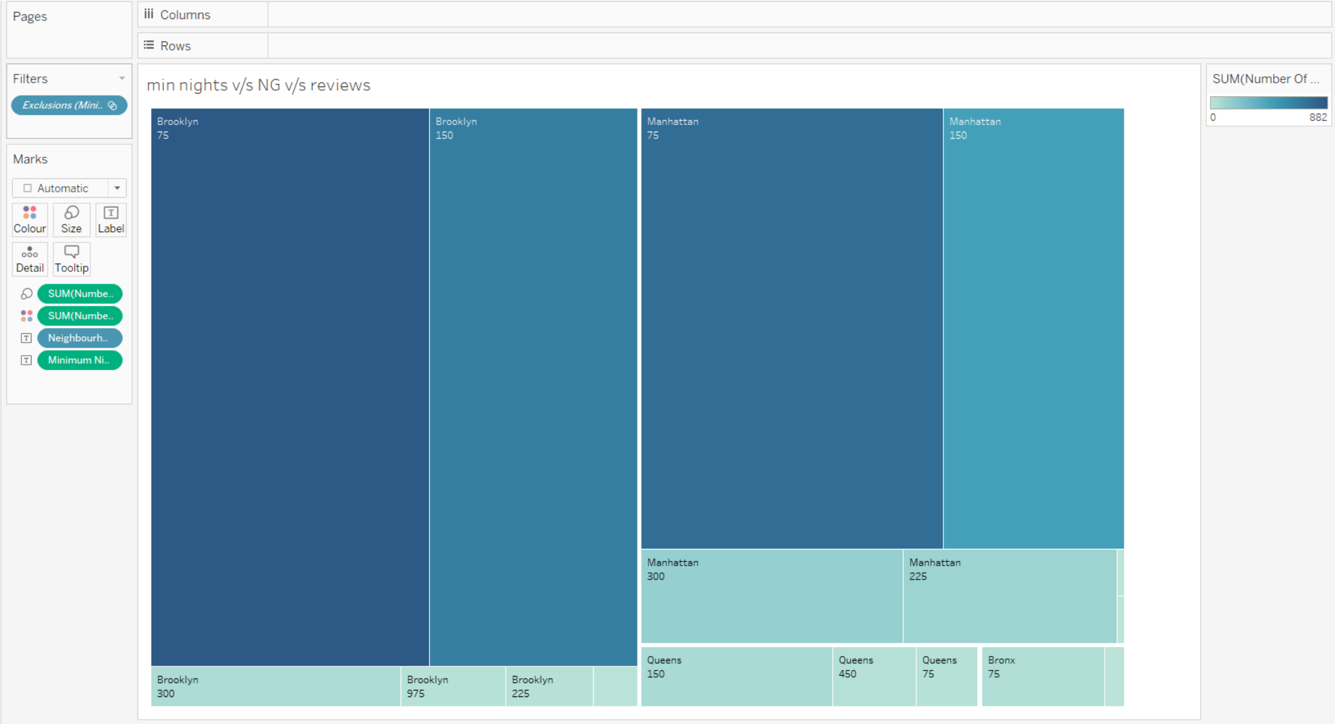


Fig : 13 Heat map of min nights v/s neighborhood group v/s reviews

After removing the reviews with count 0, we have plotted a heat map to understand the most preferred areas of stay for the customers.
From the graph we could understand that :

- Brooklyn with room availability of 75 and 150 nights are the most preferred ones for stay.
- Similarly, Manhattan with 75 and 150 nights stay are also the next preferred options for the customers.

Thank you