AIRBNB Case Study

IIIT-B

METHODOLOGY DOCUMENT PPT 1:

IN THE CASE STUDY WE HAVE USED JUPITER NOTEBOOK TO PERFORM INITIAL ANALYSIS OF THE DATA AND TABLEAU FOR DATA ANALYSIS AND VISUALIZATION.

INITIAL ANALYSIS USING JUPITER NOTEBOOK: DATA SET USED: ***AB_NYC_2019.CSV***

NUMBER OF ROWS: 48895

NUMBER OF COLUMNS: 16

By
Pabitra Kumar Pradhan
Ujwal Kesharwani
Vedha Priya Kumaresan

In [2]:
```
import pandas as pd, numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [7]:
```
df = pd.read_csv("AB_NYC_2019.csv")
df.head()
```

Out[7]:

	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	

In [8]:
```
df.describe()
```

Out[8]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.

In [9]:
```
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
```

check missing value in data set

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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 [11]:
```
## we have missing value but that dont effect our data set
df.drop(['id','name','last_review'],axis=1,inplace=True)
```

In [12]:
```
df.shape
```

Out[12]:
```
(48895, 13)
```

In [13]:
```
df.fillna({"reviews_per_month":0},inplace=True)
```

In [14]:
```
df.reviews_per_month.isnull().sum()
```

Out[14]:
```
0
```

In [15]:
```
# now we check unique values in data set
df.room_type.unique()
```

Out[15]:
```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

In [16]:
```
len(df.neighbourhood.unique())
```

Out[16]:
```
221
```

In [17]:
```
df.neighbourhood_group.unique()
```

Out[17]:
```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

Step 2: Data Wrangling:

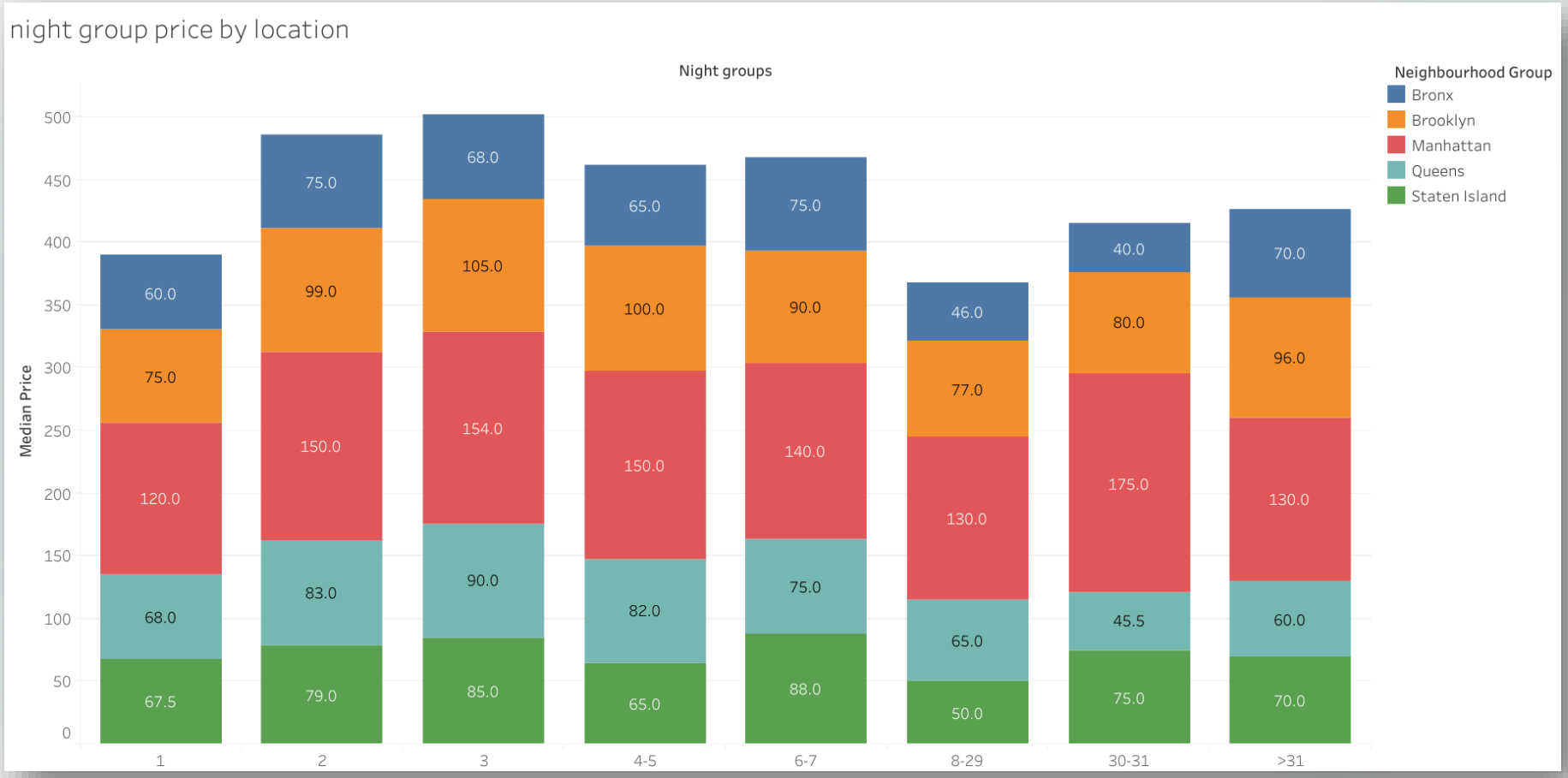
- Checked the Duplicate rows in our dataset and no duplicate data was found.
- Checked the Null Values in our dataset. Columns like name, host-name, last review and review-per-month have null values.
- We've dropped the column name as missing values are less and dropping it won't have significant impact on analysis.
- Checked the formatting in our dataset. Identified and review outliers.

Data Analysis and Visualizations using Tableau:

- We have used tableau to visualize the data for the assignment. Below are the detailed steps used for each visualization.

1. Booking Price with respect to minimum nights.

- We create a bar chart to understanding Booking Price with respect to minimum nights.
- We added Location to the colors Marks card to highlight the different Location in different colors and count of Host Id and median price to the size.

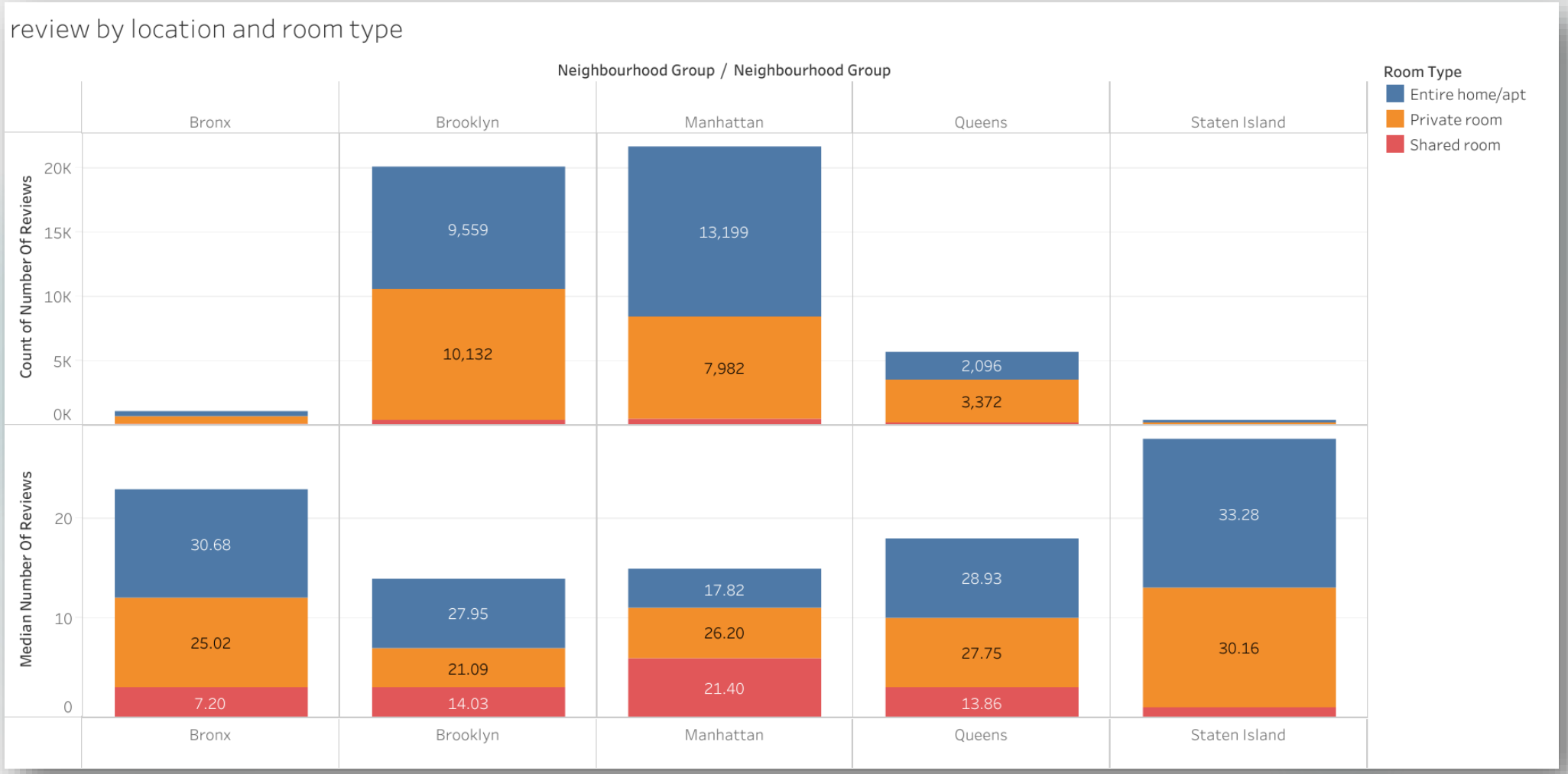


2. Customer Booking w r t minimum nights:

- 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 ids booked for each neighborhood group.

3. Reviews by Location And Room type

- We create a two bar chart for Reviews by location and room type.
- We add a room type in color.
- We add count of number of reviews and avg. number of reviews in row side.
- We add Neighborhood group in columns.



4. Reviews Room and Minimum Night

- We create a Heat Map for Reviews, room type, Minimum nights and Neighborhood group.
- We add Number of reviews in colors and Label.
- We add Neighborhood group columns.
- We add Room type and Night group in rows.

location by night group room type rating

Neighbourhood Group

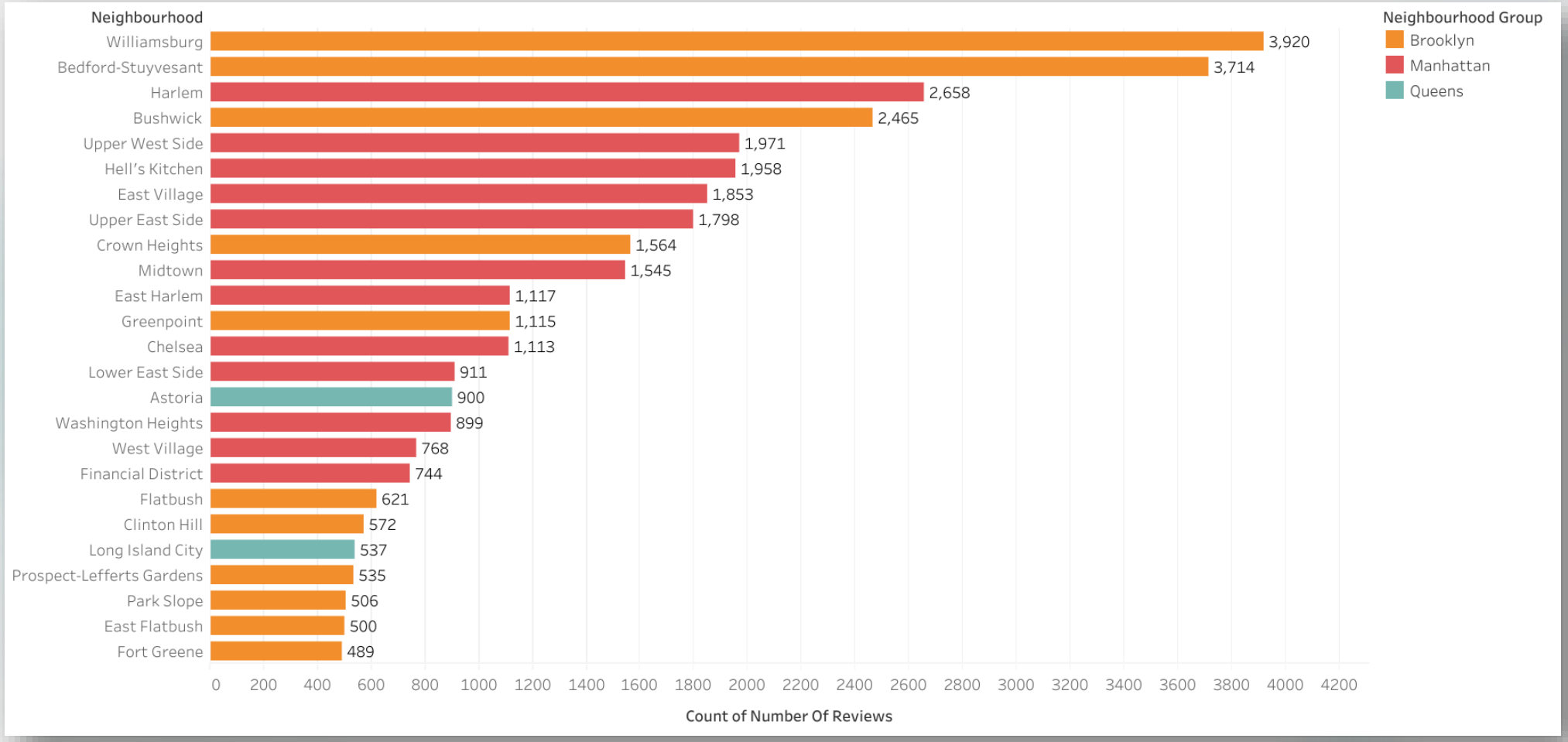
Avg. Number Of Reviews

0.00 50.18

Room Type	Night groups	Bronx	Brooklyn	Manhattan	Queens	Staten Island
Entire home/apt	1	33.93	32.22	23.98	41.47	36.41
	2	37.27	37.21	23.77	32.16	40.56
	3	35.11	32.19	22.72	28.99	27.18
	4-5	22.30	21.79	17.74	18.86	35.75
	6-7	5.06	10.45	12.16	9.77	16.00
	8-29	3.78	8.72	7.28	7.22	1.33
	30-31	4.60	16.83	6.21	11.73	11.56
	>31	5.20	14.69	9.56	20.93	0.00
Private room	1	24.79	25.89	30.54	37.75	20.73
	2	29.44	27.07	32.71	27.26	50.18
	3	23.82	21.21	28.07	19.41	25.88
	4-5	26.09	15.59	20.84	17.46	25.57
	6-7	9.59	7.15	11.00	13.82	1.33
	8-29	12.21	7.08	10.00	10.79	0.50
	30-31	7.87	7.73	8.57	4.07	12.75
	>31	0.00	7.77	9.43	10.67	1.00
Shared room	1	7.78	20.42	22.76	19.73	4.50
	2	7.13	14.18	25.60	8.80	0.75
	3	9.00	11.74	35.53	4.44	0.00
	4-5		24.26	6.83	8.57	1.00
	6-7	5.00	1.00	15.67	2.00	
	8-29	2.00	3.42	2.31	0.00	
	30-31		1.64	18.75	2.17	
	>31	0.00	1.00	0.56	0.50	

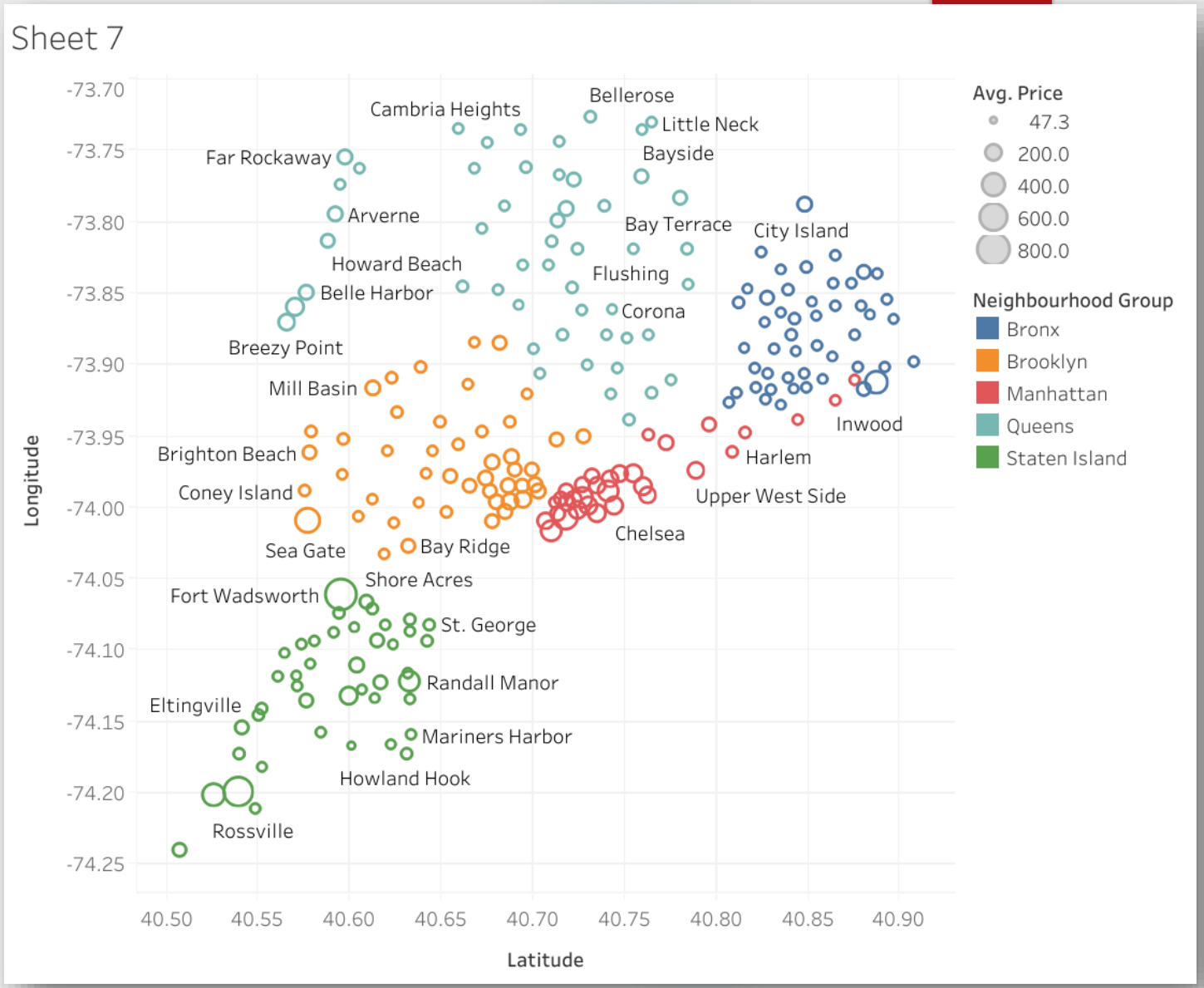
5. Top 25 Neighborhood

- We create a horizontal bars for top 25 Neighborhood.
- We add a neighborhood in colors and row.
- We add a count of Number of Reviews in Labels and Columns.



6. Price variation with respect to Geography

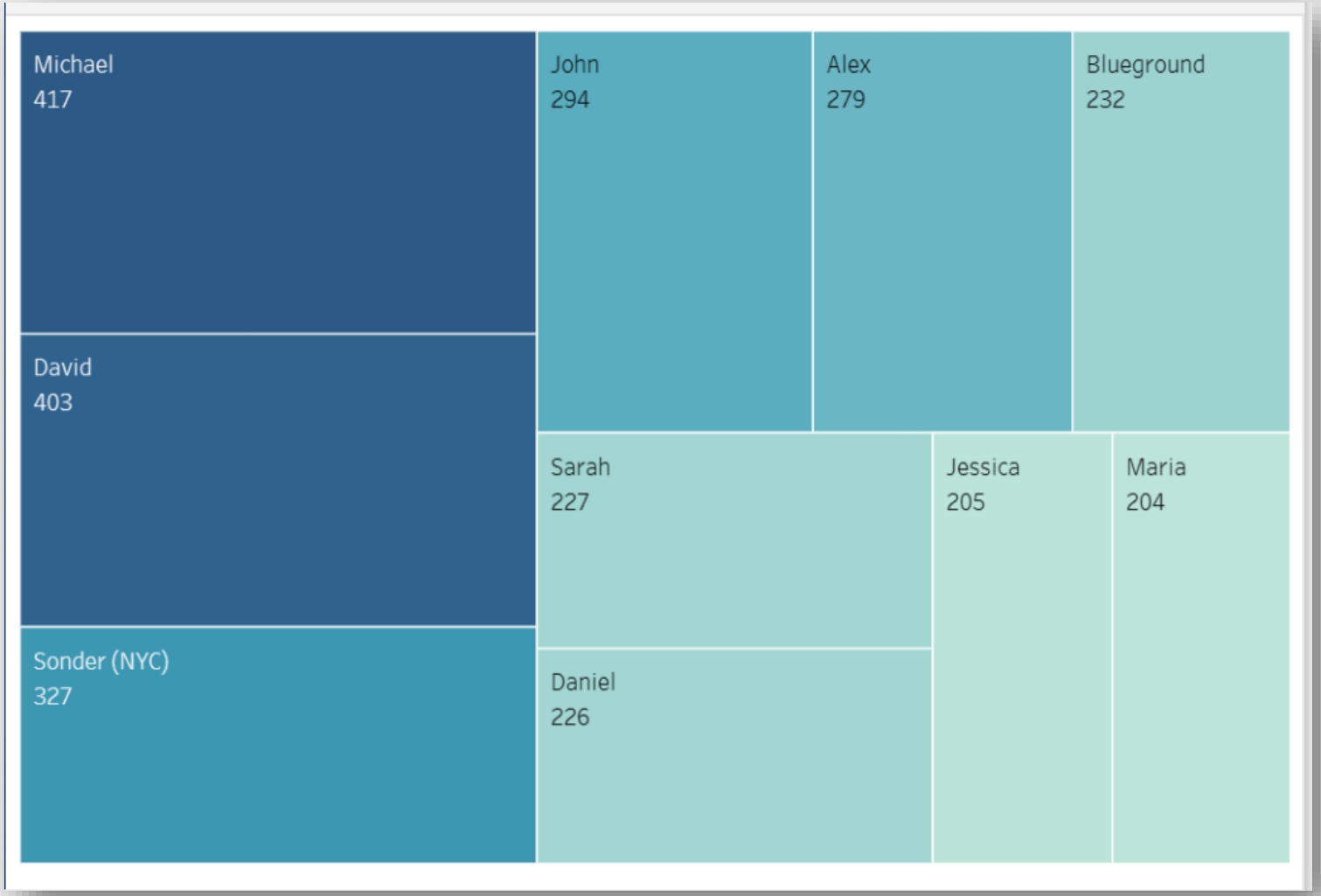
- We create a Map for Price variation with respect to Geography.
- We add a Neighborhood in Detail , Label and Colors.
- We add a avg. of Price in Size.
- We add a Latitude and Longitude.



Methodology Document PPT 2:

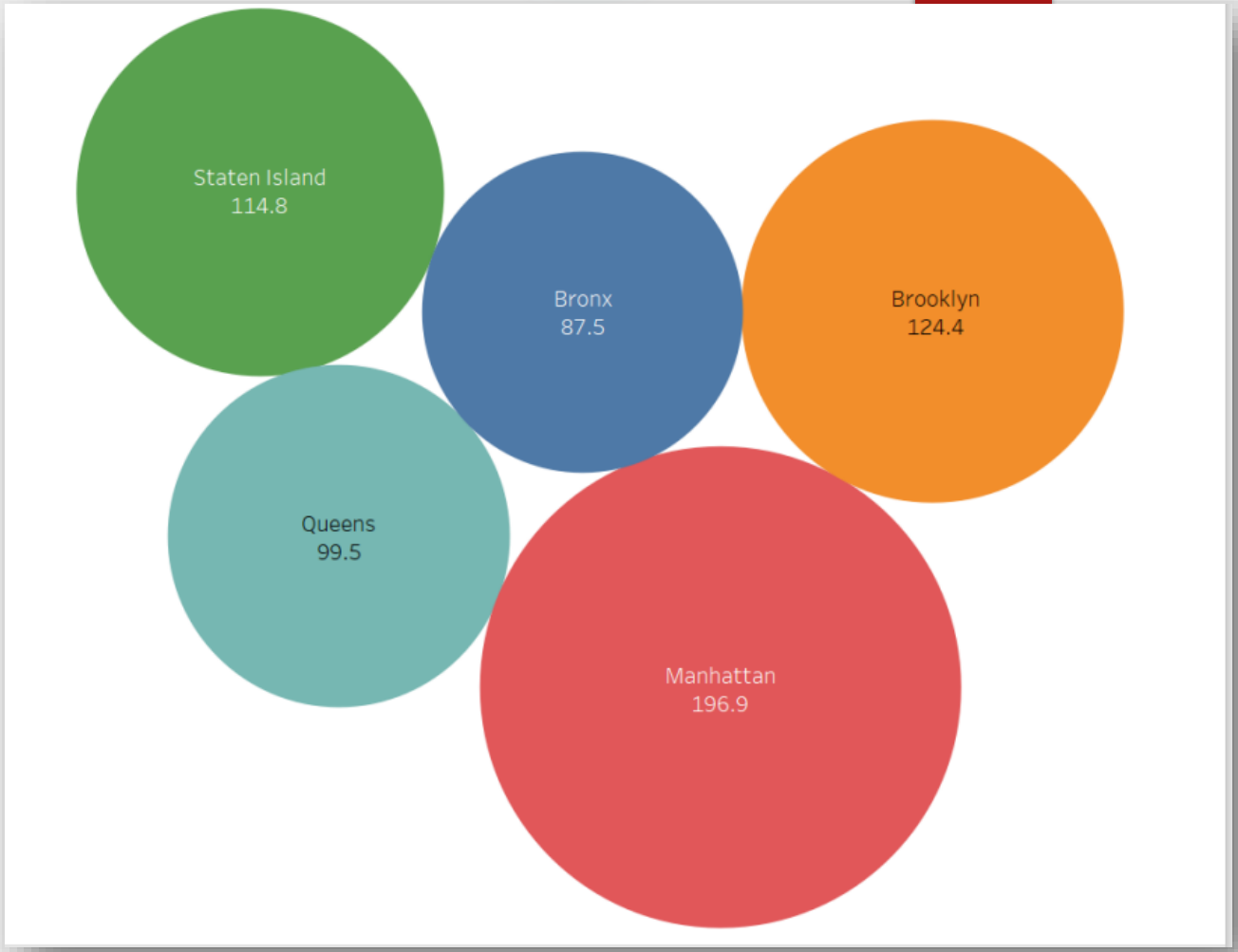
1. Top 10 Host

- We created tree map for top 10 host.
- • We added count of host id on Size, Label, Colors.
- • We added host name on Label and filter top 10 by count.



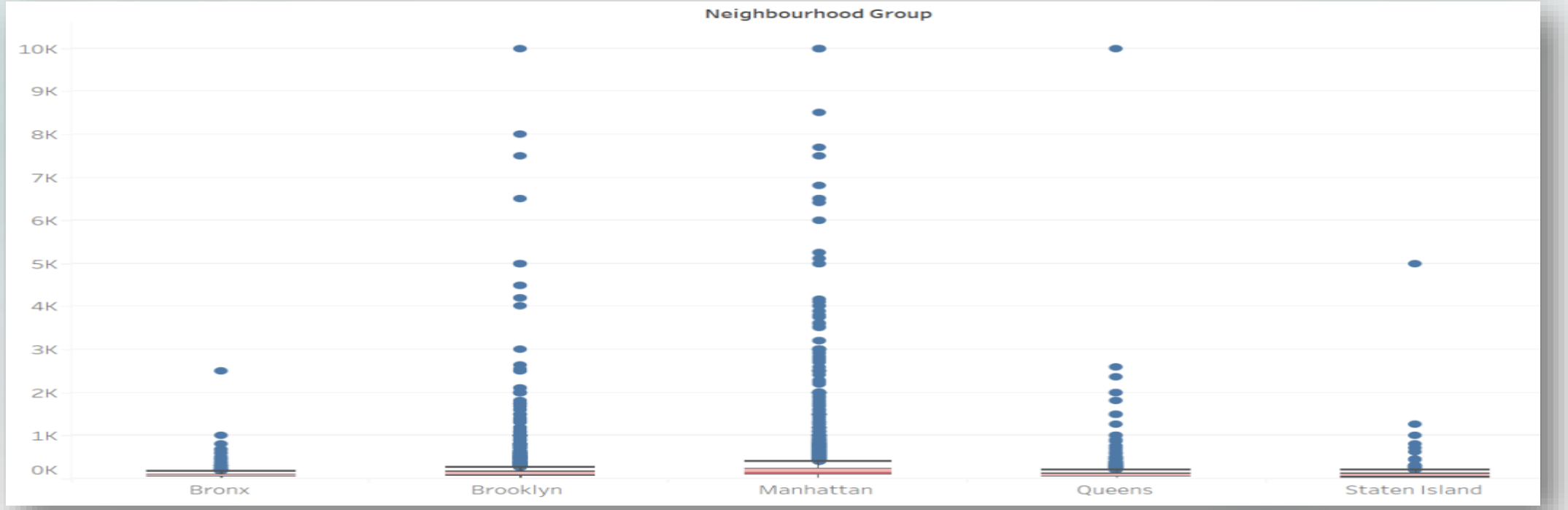
2. Average price of Neighborhood groups.

- The average price of listed properties in Manhattan is around 196.9, which is highest among all neighborhoods.
- Average price for Brooklyn is second highest i.e. 124.4.
- Bronx appears to be an affordable neighborhood as the average price is almost half than Manhattan's average price



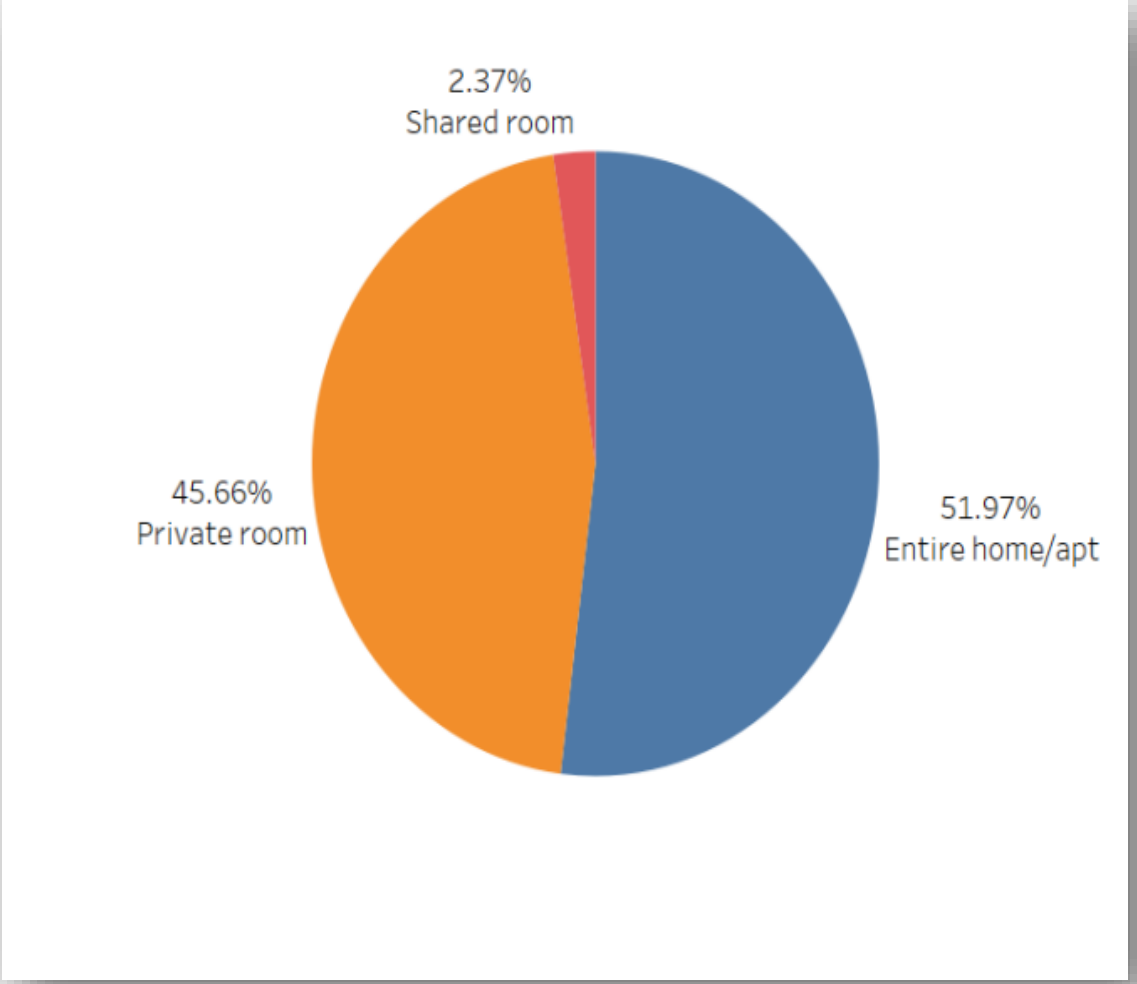
3. Price Analysis Neighborhood wise.

- Most of the outliers in Price column are for Brooklyn and Manhattan.
- Also, Manhattan has the highest range of prices for the listings.
- Bronx is the cheapest of them all.
- We can see the median price of all neighborhood groups lying between $ 80 to $ 300.
- Price was highly positively skewed so median was very close the lower quartile with some outliers as seen in the boxplot below



4. Room type

- There are three types of rooms - Entire home/Apartment, Private room & shared room.
- Overall, customers appear to refer private rooms (45%) or entire homes (52%) in comparison to shared rooms (2.4%).
- Airbnb can concentrate on promoting shared rooms with discounts to increase bookings and also acquire more private listings.



Tools used:

- Data cleaning and preparation: ***Jupyter notebook – Python***
- Visualization and analysis: ***Tableau***
- Data Storytelling: ***Microsoft PPT***