Airbnb Case Study - Methodology

Introduction:

This case study is about finding important insights based on various attributes in dataset to increase their revenue after the

Problem Statement:

For the past few months, Airbnb has seen a major decline in revenue due to the lockdown imposed during the pandemic. 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

Objective:

The main objective of this case study is to understand the customer preferences, get a better understanding of Airbnb listings with respect to various parameters and give insights to improve their business.

Data Information:

- Decline in the revenue could be for two major reasons, either the sites hosted on the platform are not able to provide better user experience or there could be a competitor in the market capturing the market share.
- Keeping the above in mind, we first try to work on the first reason as that is something internal to the company and can have the data in hand to identify the reasons behind the dropping of the revenue. Hence, we use the information of the hosted places on the platform to see where and what can be done to improve the end consumer experience.
- The data would majorly include the location and region of the hosted places, in our case we are targeting Borough (New York City) — the Bronx, Brooklyn, Manhattan, Queens and Staten Island, followed by their hosts details, prices of the hosted sites and reviews received by the end consumer.

To whom are we presenting?

- **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.
- **Head of Acquisitions and Operations, NYC:** This head looks after all the property and hosts 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 optimize the order of property listing in certain neighbourhoods and cities in order to get every property the optimal amount of traction.

Methodology:

Data Understanding and Preparation:

1. First, we have understood the data of the dataset in python.
2. Then we have handled the missing values. It is observed that there are null values are present in name, host_name, last_review, reviews_per_month
3. reviews_per_month also has the missing values we have replaced these values with '0' respectively and other columns are not efficient to the dataset so we have decided to drop those columns.
4. There are no missing values present in reviews_per_month columns, also we have checked for the unique values present in the columns, it shows that there are no duplicate values are present in the dataset.
5. Then separated the columns of dataset into categorical and numerical data types.
6. Then identified and reviewed outliers present in the dataset.

Data Understanding and Preparation:

Before we start the basic understanding of the data in hand, we imported relevant libraries available in Python. Below are the libraries that we imported,

We started with Understanding the Data in hand provided by running basic functions to load and interpret the variables, data types of the variables, dimensions and size of the data frame. Below is the code used for the same

Airbnb Case Study-Analysis

```
In [1]: # Import all the important libraries
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
        from scipy import stats
        import datetime as dt
        import plotly
        import plotly.express as px

In [2]: # Ignore warnings
        import warnings
        warnings.filterwarnings("ignore")

In [3]: pd.set_option("display.max_columns", None)

        pd.set_option("display.max_rows", None)

        plt.style.use("seaborn-dark-palette")

In [4]: # import the data
        data = pd.read_csv(r"C:\Users\Lenovo\Downloads\AB_NYC_2019.csv")
```

Activate Windows
Go to Settings to activate Windows.

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

Dataset Description

The above understandings lead us to perform basic Numeric and Categorical analysis in depth by using the following function along with some basic wear and tear.

```
In [5]: # read the data
        data.head()
```

Out[5]:

	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 [6]: # Let's have a look on the dimensions of the data
        data.shape
```



```
Out[6]: (48895, 16)
```

```
In [7]: # Let's check the data type
        data.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
```



```
In [8]: # The above description shows that there are some numerical as well as the categorical variables are present,
        # so we have to perform both the analsis numerical as well as the categoraical
```

```
In [9]: # for numerical analysis
        data.describe()
```

Out[9]:

	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.

Handling Missing Values and Outliers:

Then we moved to handle missing values and outliers in the dataframe. Starting with the missing values, we identified two columns having equal missing values which were last_review and reviews_per_month. Also, other two columns having quiet minimal

missing values which were host_name of and name of the place.

```
In [10]: # Let's check the missiong values and the outliers
         data.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]: # It is observed that there are null values are present in name, host_name, last_review, reviews_per_month
         # Certain columns are not efficient to the dataset so let us drop those columns
         data.drop(['id','name','last_review'], axis = 1, inplace = True)
```



```
In [12]: # Let us check wheather the columns are dropped or no
         data.head(5)
```

Out[12]:

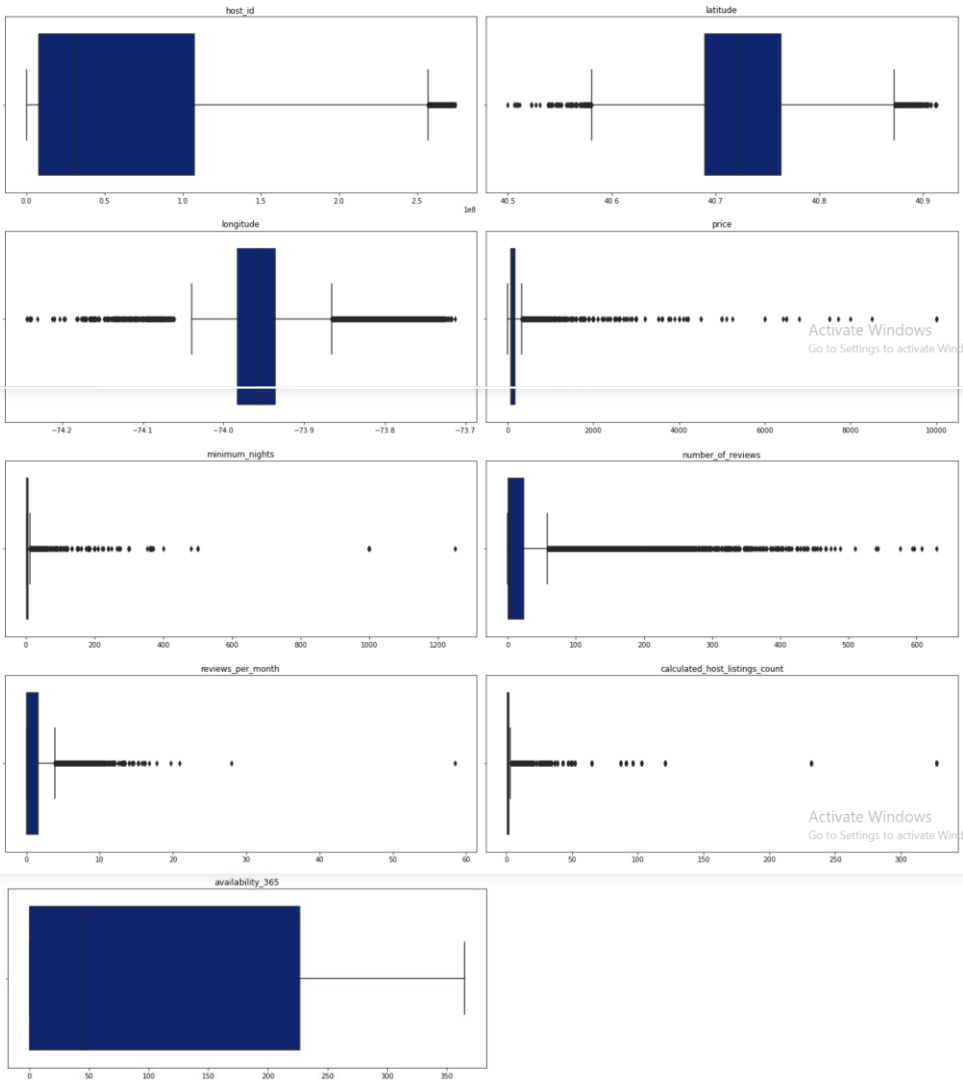
	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_month
0	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	0.21
1	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	0.38
2	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	NaN
3	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	4.64
4	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	0.10

```
In [13]: # reviews_per_month also has the missing values let us replace these values with '0' respectively
         data.fillna({'reviews_per_month':0}, inplace = True)

In [14]: data.reviews_per_month.isnull().sum()

Out[14]: 0
```



```
In [15]: # There are no missing values present in reviews_per_month column
         # Now check the uniques values present in the columns

         data.room_type.unique()

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

In [16]: len(data.room_type.unique())

Out[16]: 3

In [17]: data.neighbourhood_group.unique()

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

In [18]: len(data.neighbourhood_group.unique())

Out[18]: 5

In [19]: len(data.neighbourhood.unique())

Out[19]: 221

In [20]: # Extracting Numeric columns:

         int_cols = data.select_dtypes(include=["int64", "float64"]).columns

In [21]: # Tagging them:

         list(enumerate(int_cols))

Out[21]: [(0, 'host_id'),
          (1, 'latitude'),
          (2, 'longitude'),
          (3, 'price'),
          (4, 'minimum_nights'),
          (5, 'number_of_reviews'),
          (6, 'reviews_per_month'),
          (7, 'calculated_host_listings_count'),
          (8, 'availability_365')]

In [22]: # Plotting the spread of outliers:

         plt.figure(figsize=([20,22]))

         for n,col in enumerate(int_cols):
             plt.subplot(5,2,n+1)

             sns.boxplot(data[col])

             plt.xlabel("")

             plt.ylabel("")
```

```
    plt.ylabel("")
    plt.title(col)
    plt.tight_layout()
```



In [23]:
```
# Capping (statistical) outliers
# outlier treatment for price:
Q1 = data.price.quantile(0.10)
Q3 = data.price.quantile(0.90)
IQR = Q3 - Q1
data = data[(data.price >= Q1-1.5*IQR) & (data.price <= Q3 + 1.5*IQR)]
```

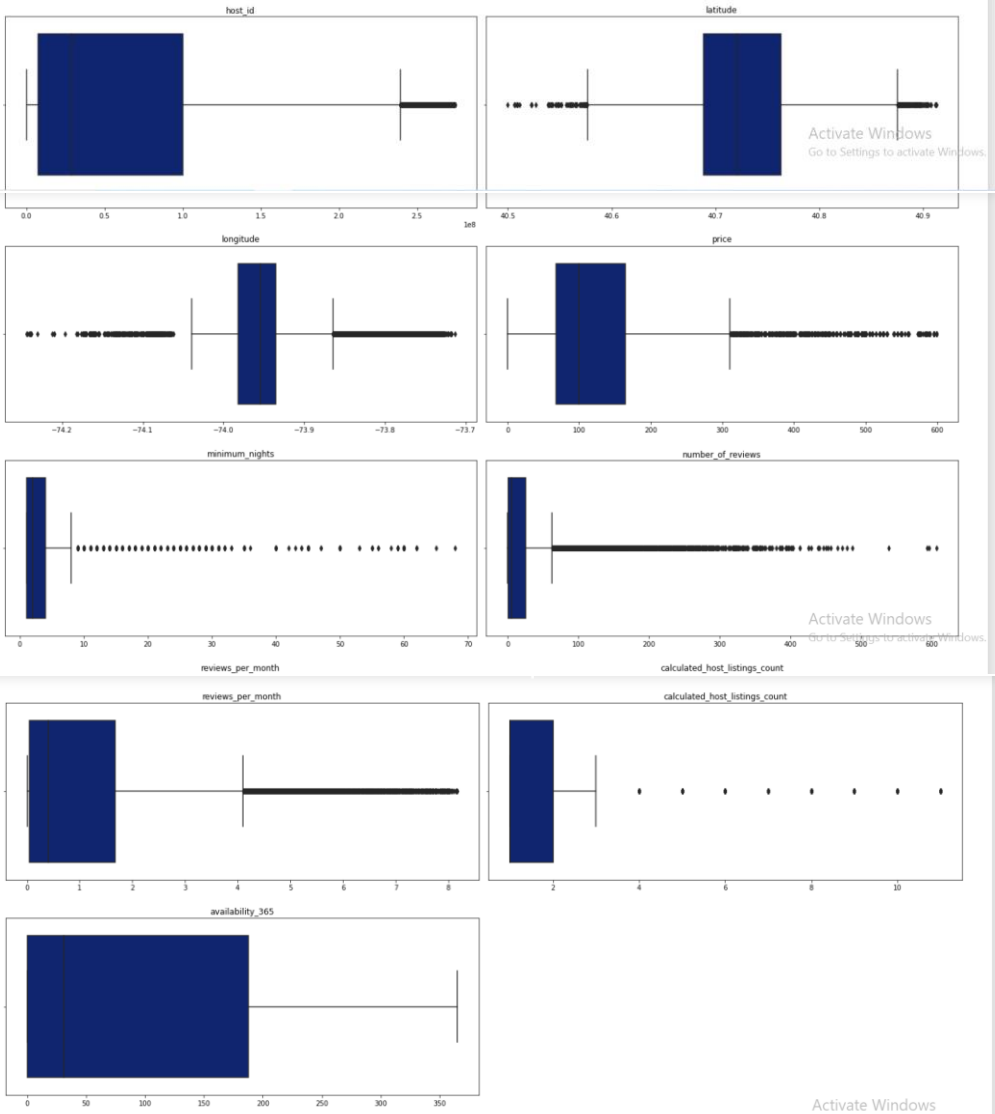

In [24]:
```
# outlier treatment for minimum_nights:
Q1 = data.minimum_nights.quantile(0.10)
Q3 = data.minimum_nights.quantile(0.90)
IQR = Q3 - Q1
data = data[(data.minimum_nights >= Q1-1.5*IQR) & (data.minimum_nights <= Q3 + 1.5*IQR)]
```

In [25]:
```
# outlier treatment for reviews_per_month:
Q1 = data.reviews_per_month.quantile(0.10)
Q3 = data.reviews_per_month.quantile(0.90)
IQR = Q3 - Q1
data = data[(data.reviews_per_month >= Q1-1.5*IQR) & (data.reviews_per_month <= Q3 + 1.5*IQR)]
```

In [40]:
```
# outlier treatment for calculated_host_listings_count:
Q1 = data.calculated_host_listings_count.quantile(0.10)
Q3 = data.calculated_host_listings_count.quantile(0.90)
IQR = Q3 - Q1
data = data[(data.calculated_host_listings_count >= Q1-1.5*IQR) &
```

```
(data.calculated_host_listings_count <= Q3 + 1.5*IQR)]
```

In [41]:
```
plt.figure(figsize=([20,22]))

for n,col in enumerate(int_cols):
    plt.subplot(5,2,n+1)

    sns.boxplot(data[col])

    plt.xlabel("")

    plt.ylabel("")

    plt.title(col)

    plt.tight_layout()
```



In [28]:
```
# Univariate Analysis:
```

Analyzing Methods:

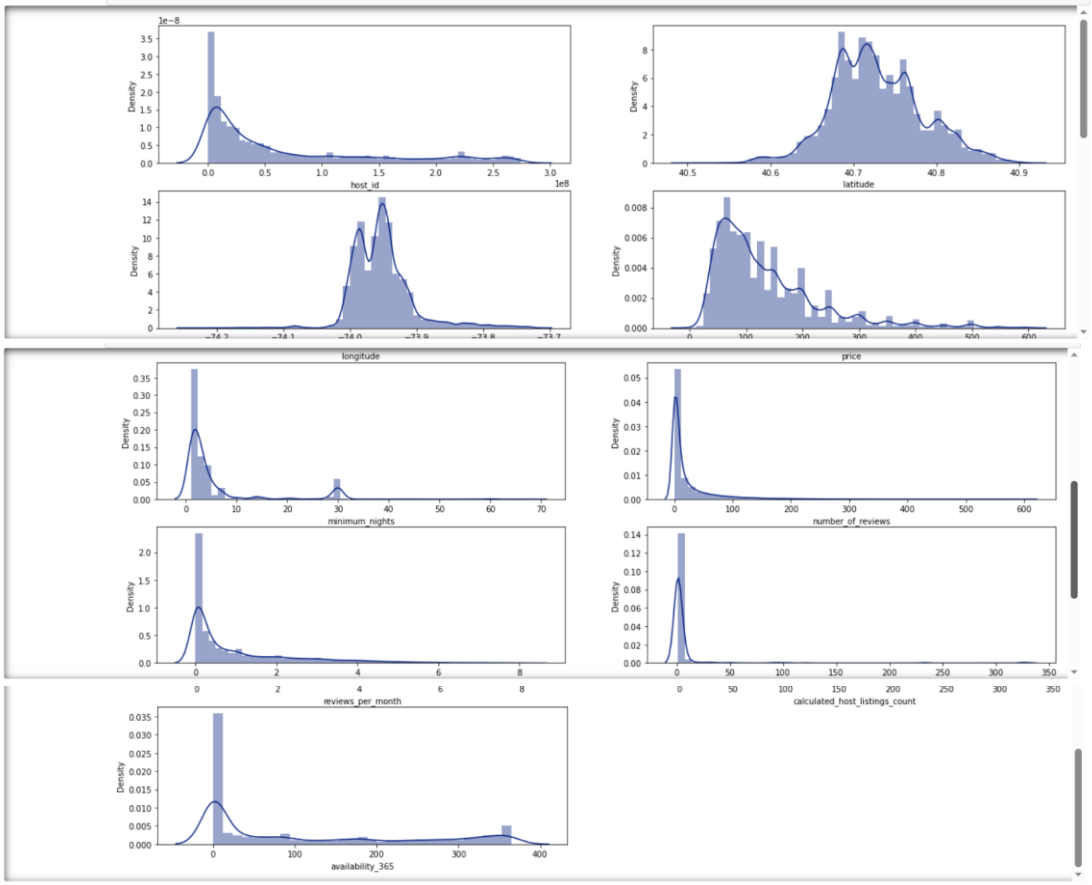
Univariate Analysis: We started our general Univariate Analysis of Numeric and Categorical columns. For numeric columns we used a Distribution plot from seaborn and for categorical columns we used a Countplot from the same library seaborn. Below are the codes for the same.

```
# We started our general Univariate Analysis of Numeric and Categorical columns.
# For numeric columns, we used a Distribution plot from seaborn and for categorical columns,
# we used a Countplot from the same library seaborn. Below are the codes for the same.
```

In [29]:
```
# Extracting and Tagging the Numeric Columns:

int_cols = data.select_dtypes(include=['int64', 'float64']).columns

list(enumerate(int_cols))
```

Out[29]:
```
[(0, 'host_id'),
 (1, 'latitude'),
 (2, 'longitude'),
 (3, 'price'),
 (4, 'minimum_nights'),
 (5, 'number_of_reviews'),
 (6, 'reviews_per_month'),
 (7, 'calculated_host_listings_count'),
 (8, 'availability_365')]
```

In [30]:
```
# Plotting the Numeric Variables Distribution:

int_cols = data.select_dtypes(include=['int64', 'float64']).columns

plt.figure(figsize=[20,18])

for n,col in enumerate(int_cols):

    plt.subplot(5,2,n+1)

    sns.distplot(data[col])
```



In [31]:
```
# Findings:

# The Highest price range seems to be between 30 dollars to 150 dollars per day stay for most of the sites hosted.
```

Findings:

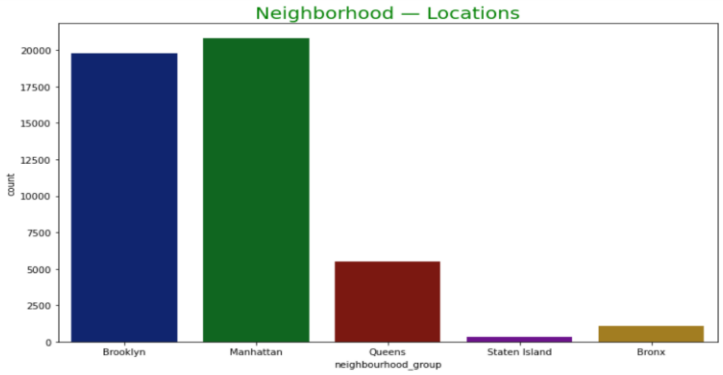
- The Highest price range seems to be between 30 dollars to 150 dollars per day stay for most of the sites hosted.
- Still we can see there are many sites which cost more than 200 dollars per day and can even go upto 500 dollars.

In [31]:
```
# Findings:

# The Highest price range seems to be between 30 dollars to 150 dollars per day stay for most of the sites hosted.

# Still, we can see there are many sites that cost more than 200 dollars per day and can even go up to 500 dollars.
```

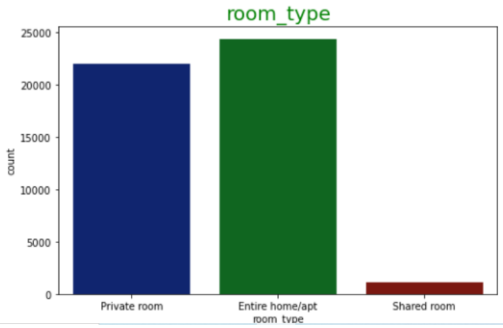
In [32]:
```
# Extracting and Tagging the Numeric Columns:

cat_cols = data.select_dtypes(include=['object']).columns

list(enumerate(cat_cols))
```

Out[32]:
```
[(0, 'host_name'),
 (1, 'neighbourhood_group'),
 (2, 'neighbourhood'),
 (3, 'room_type')]
```

In [33]:
```
# Checking the count of Neighborhood Groups

plt.figure(figsize=[12,7])

sns.countplot(data.neighbourhood_group)

plt.title('Neighborhood — Locations', fontdict={'fontsize': 20, 'fontweight': 5, 'color': 'Green'})

plt.show()
```

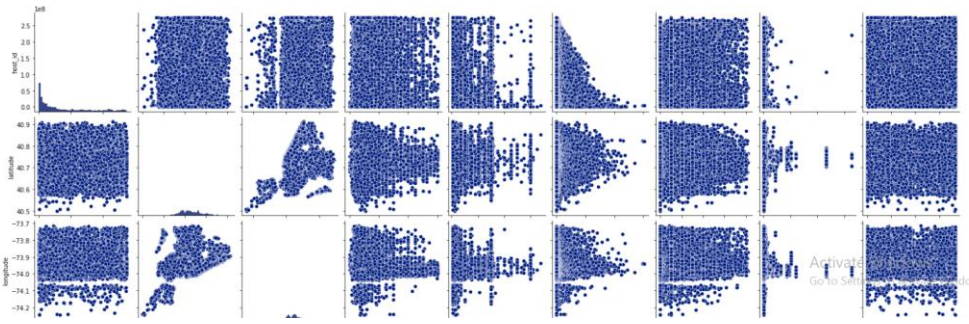



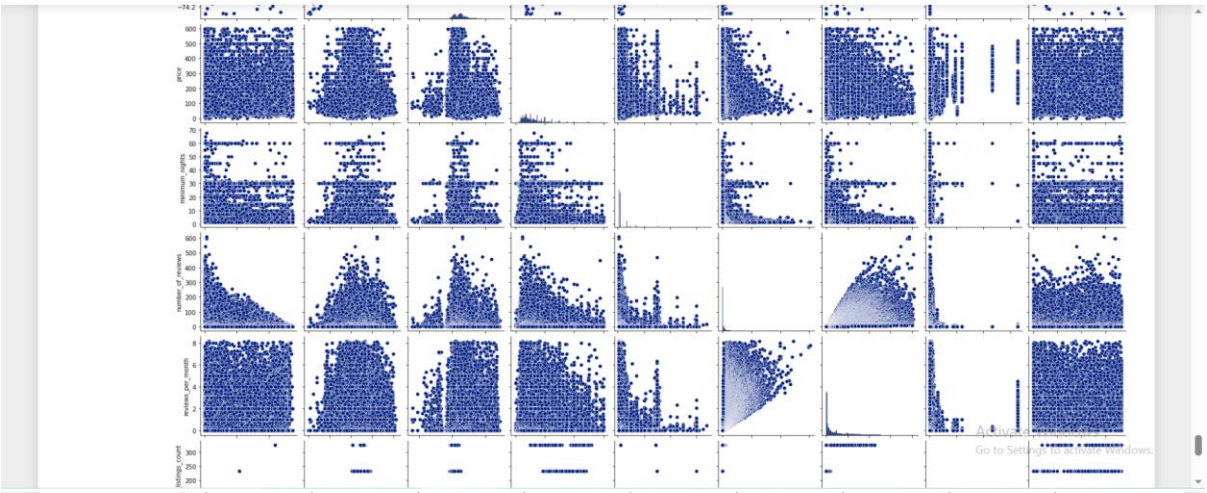
In [34]:
```
# Checking the count of Room Type

plt.figure(figsize=[8,5])

sns.countplot(data.room_type)

plt.title('room_type', fontdict={'fontsize': 20, 'fontweight': 5, 'color': 'Green'})

plt.show()
```



In [35]:
```
# Bi-Multivariate Analysis:

# Here we first plotted a pairplot of all the numeric columns using seaborn library in Python itself.
# Below is the code for the same.
```

In [36]:
```
sns.pairplot(data)

plt.show()
```






```
In [38]: # Parallel to we created all other Bivariate and Multivariate plots using Tableau.

In [39]: data.to_csv('AIRBNB_modified.csv',index=True)
```

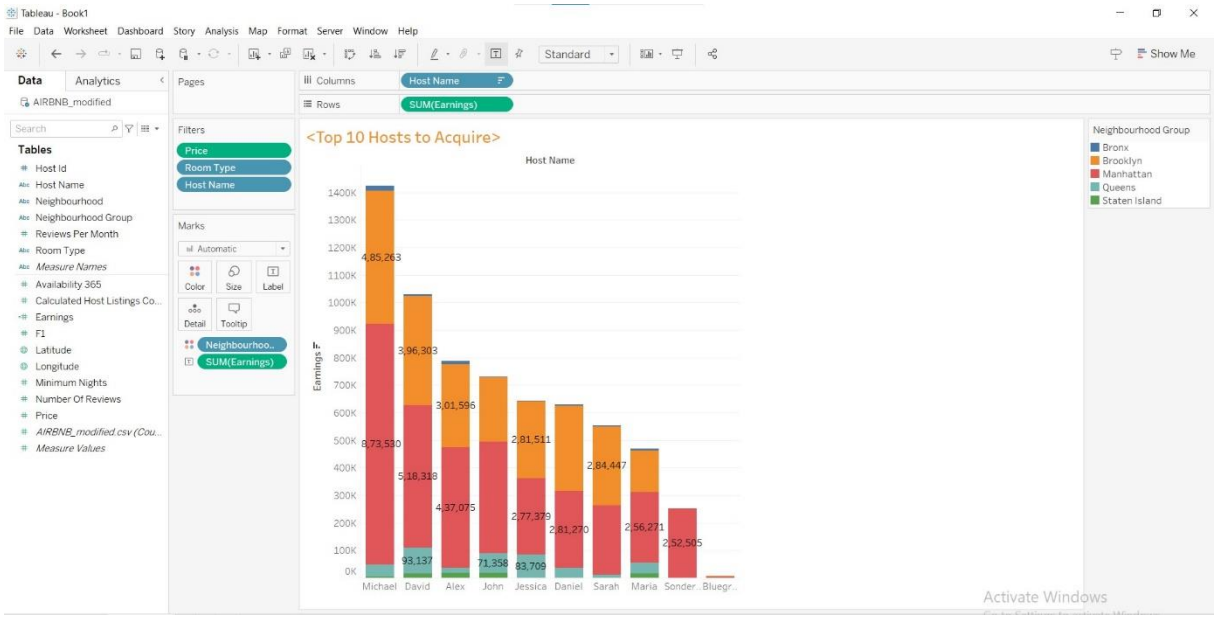
Data Visualization:

The visualization part we have did in tableau for various attributes to increase the revenue of Airbnb such as

1. Which type of hosts to acquire more and where?
2. The categorization of customers based on their preferences.
3. What are the Neighbourhoods they need to target?
4. What is the pricing ranges preferred by customers?
5. The various kinds of properties that exist w.r.t. customer preferences.
6. Adjustments in the existing properties to make it more customer oriented.
7. What are the most popular localities and properties in New York currently?
8. How to get unpopular properties more traction? And so on...

Tableau Analysis:

1. **Top 10 hosts with respect to earnings and Neighbourhood group:**

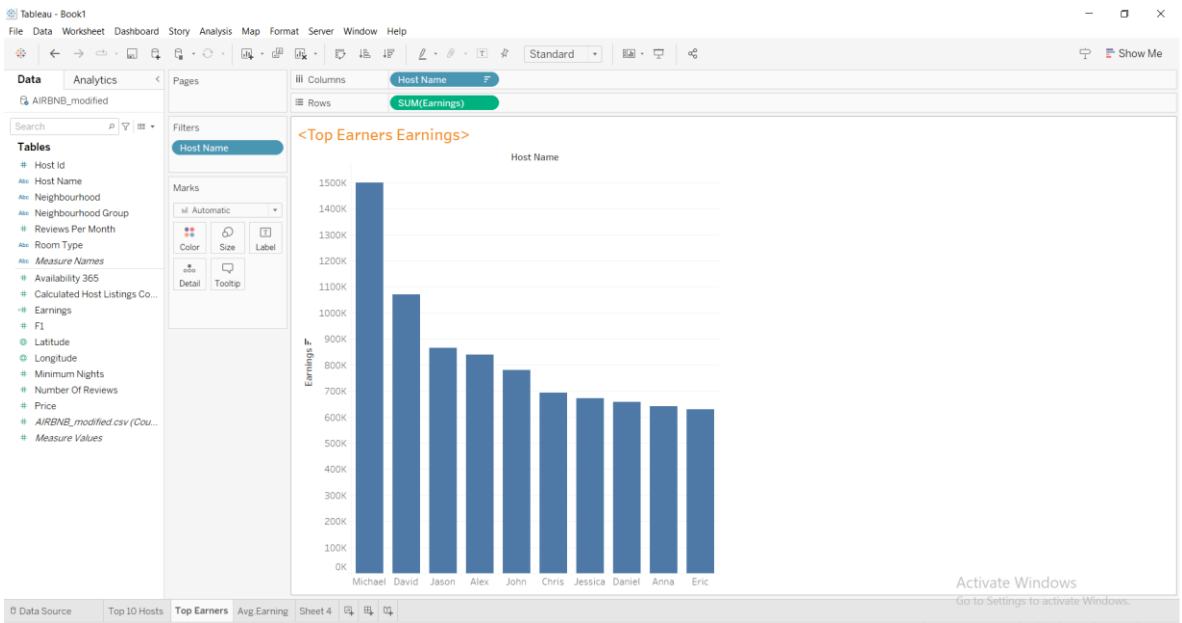


The graph depicts the top 10 hosts who are earning more. We have observed a single host (Michael) having a greater number of properties overall and has more earnings. This is because Manhattan has the highest influx of tourists and financial enthusiasts visiting city all year around and he has highest revenue from Manhattan.

Created a calculated field for earnings: [price]*[Number of Reviews]

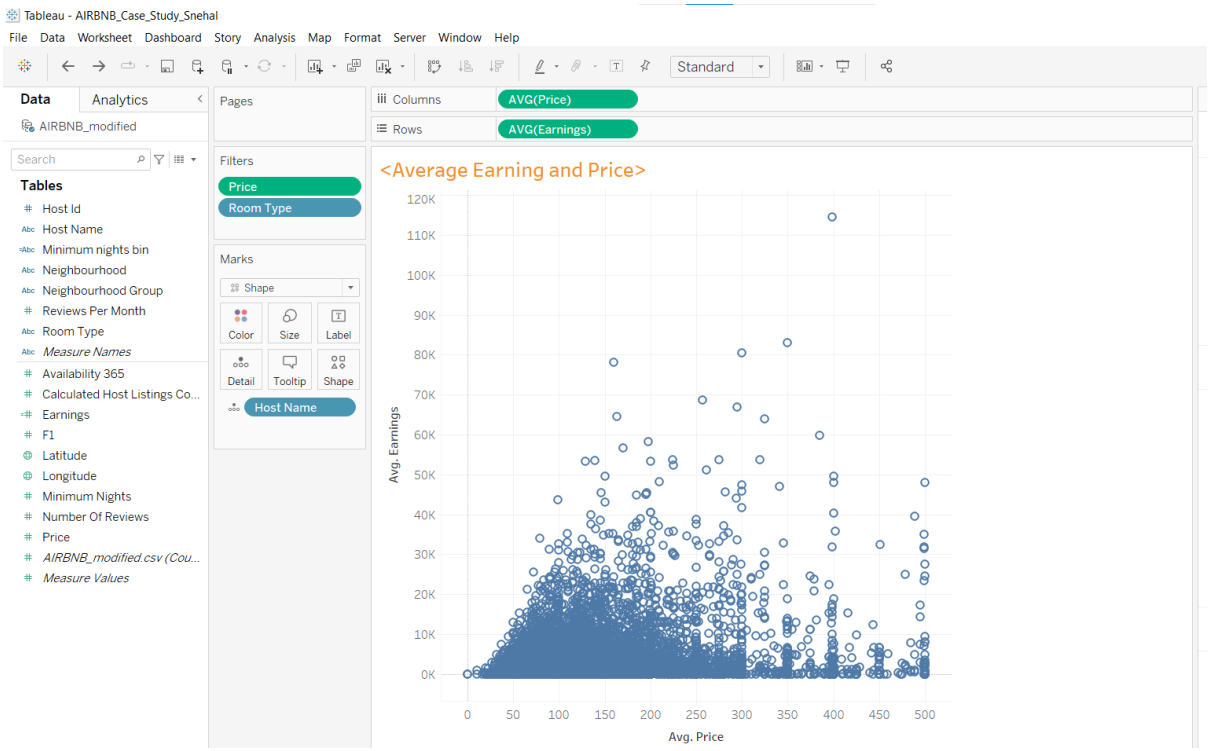
2. Top Earners Earnings –

Michael is earning more in a single day.



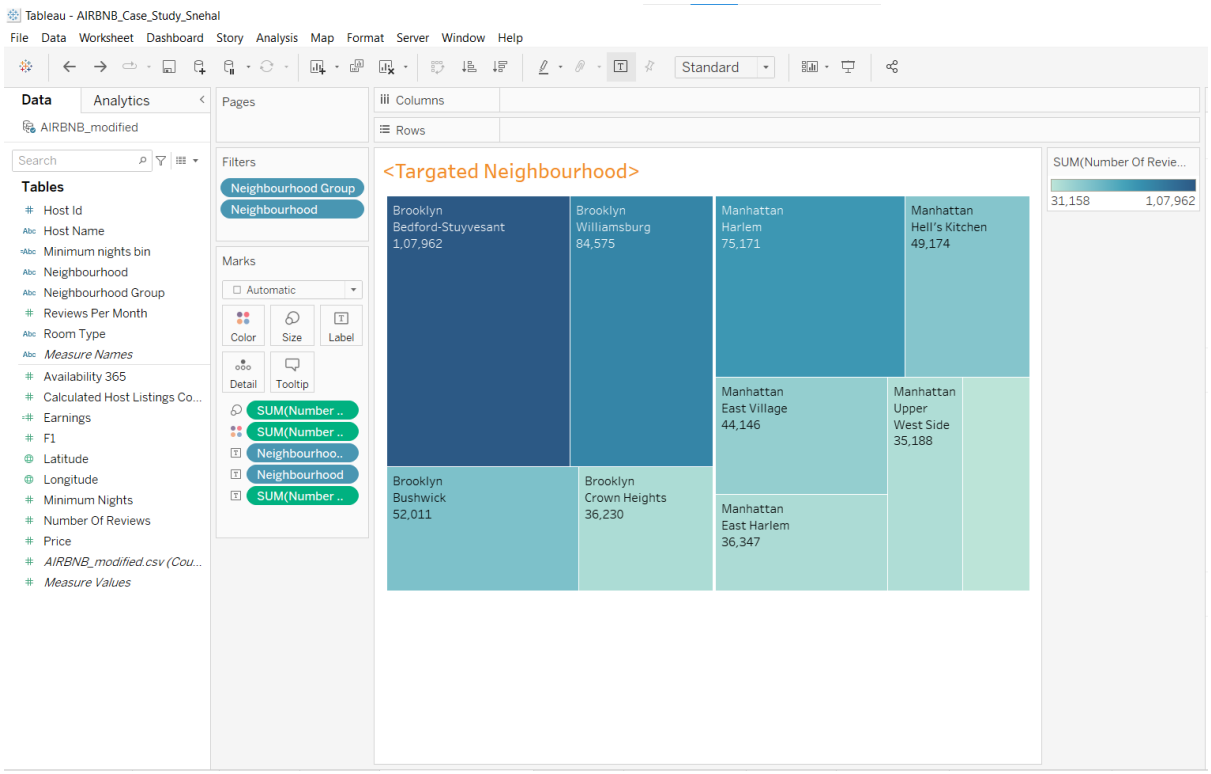
3. Average earning and price:

According to this graph, the majority of individuals would have spent between $40 and $250. A typical host makes between $6000 and $7000 per year. The hosts who charge 170 or more as the standard fee make around 10,000.



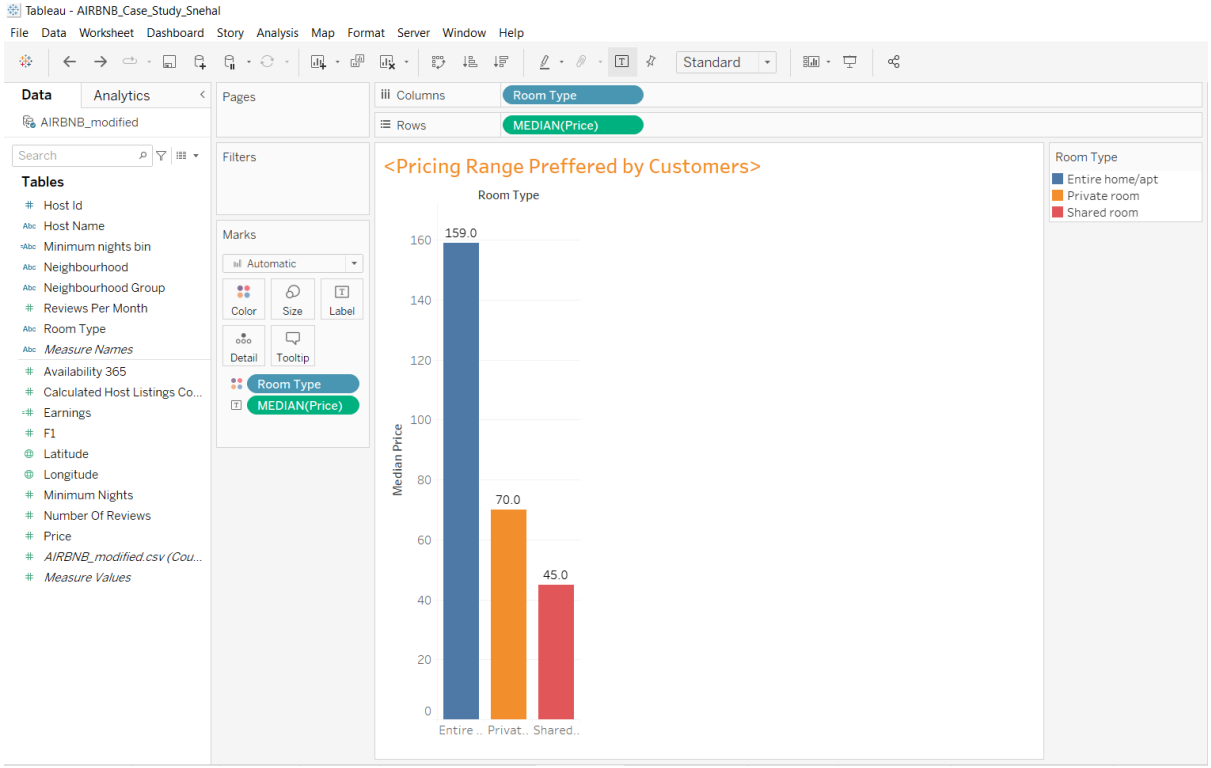
4. Targeted Neighbourhood:

These are the most targeted Neighbourhoods by customers. This may be mostly visited as they are near to beach and the facilities are best in this locality.



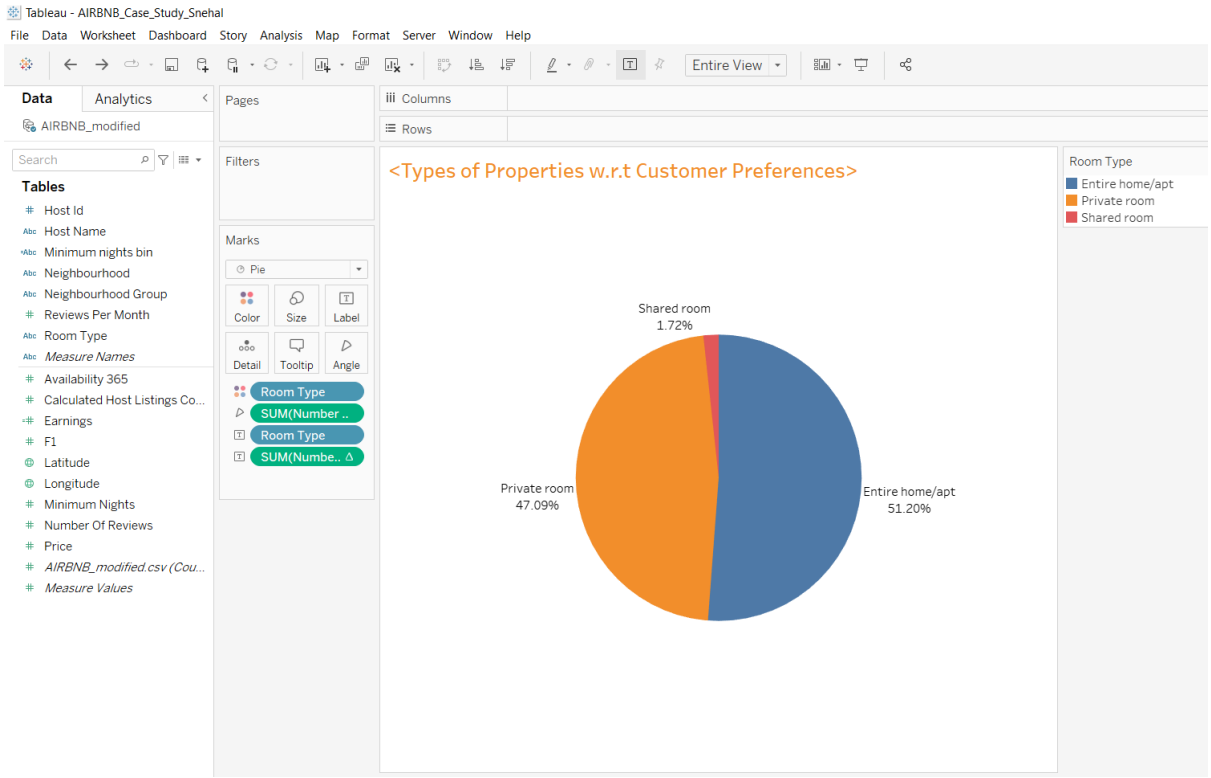
5. Pricing range preferred by customer on basis of room type:

On basis of room type, the customers prefer entire home mostly. The pricing range is 159 for entire home, then private room 70 and least is shared room 45.



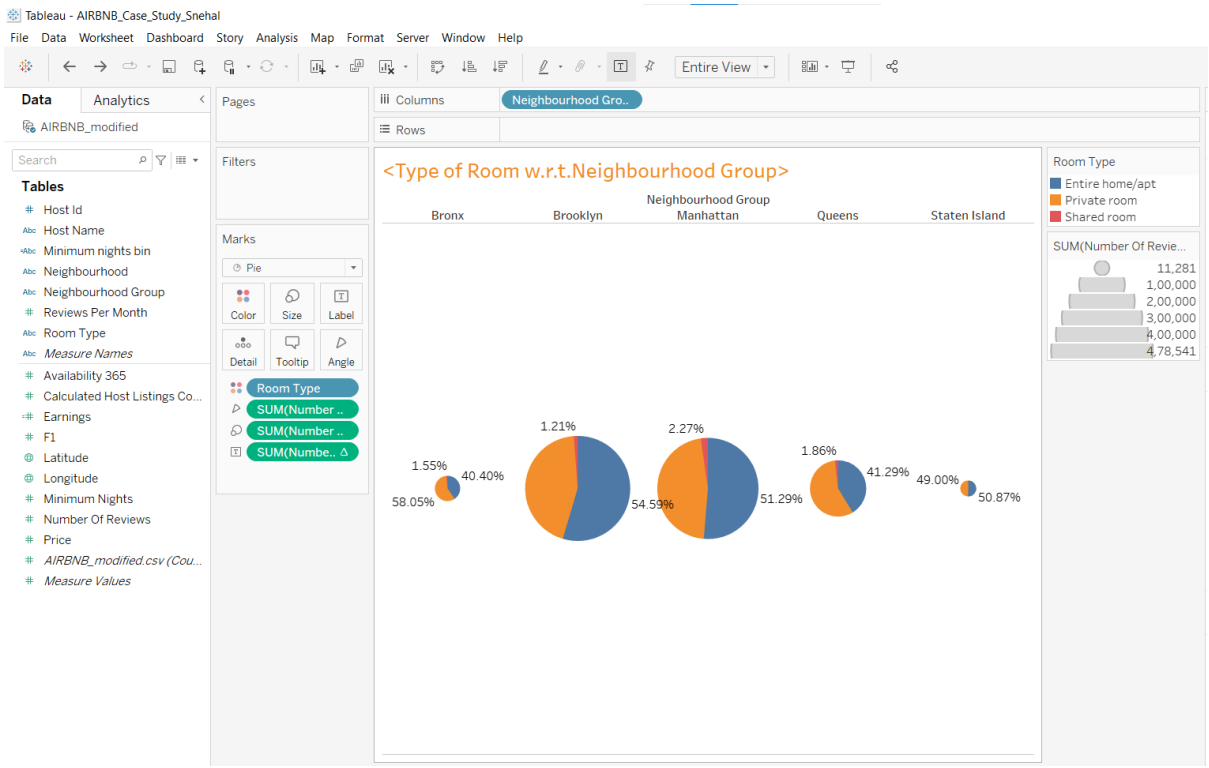
6. Types of properties preferred by customers:

The customers mainly prefer entire home (51.20%) and then private room (47.09%). So, the airbnb can increase this type of rooms more and provide discounts to attract customers more. Shared rooms are preferred less by the customers.



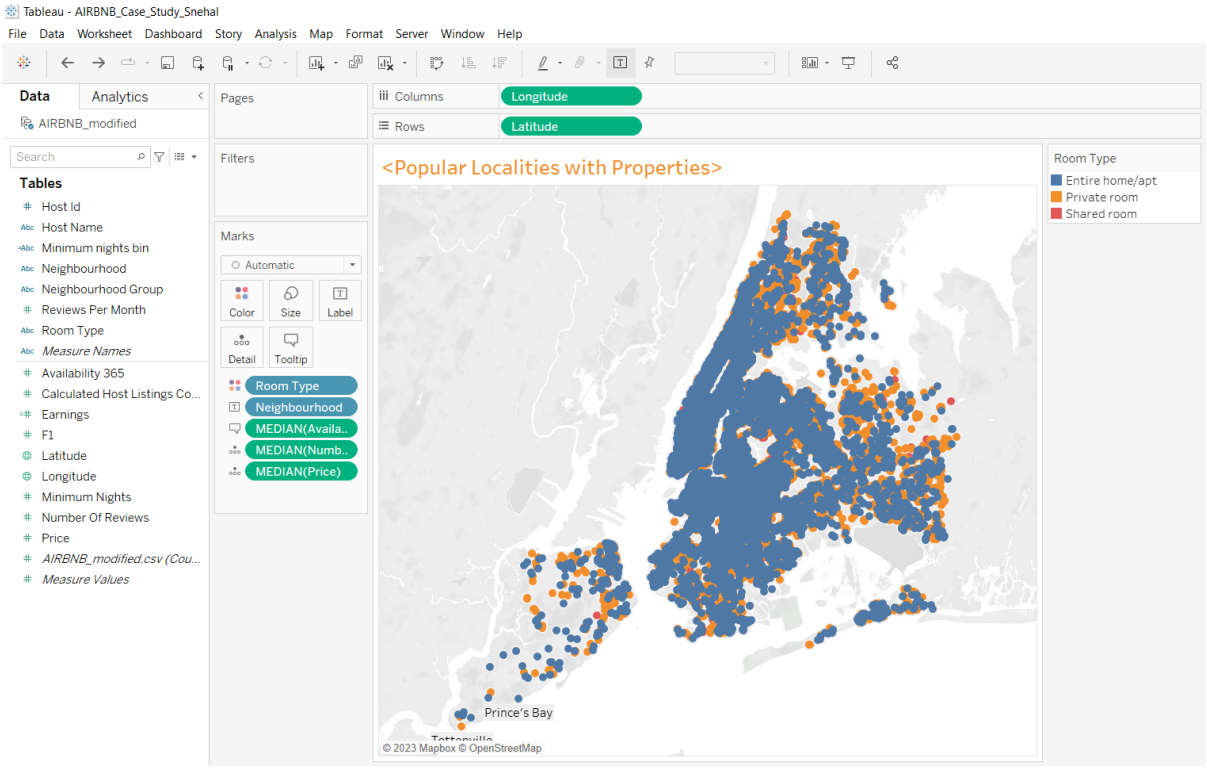
7. Type of room type preferred by customer's w.r.t Neighbourhood group:

This graph shows that in Manhattan, Brooklyn, Staten Island entire home is preferred by customers whereas in Bronx and Queens private rooms are preferred.



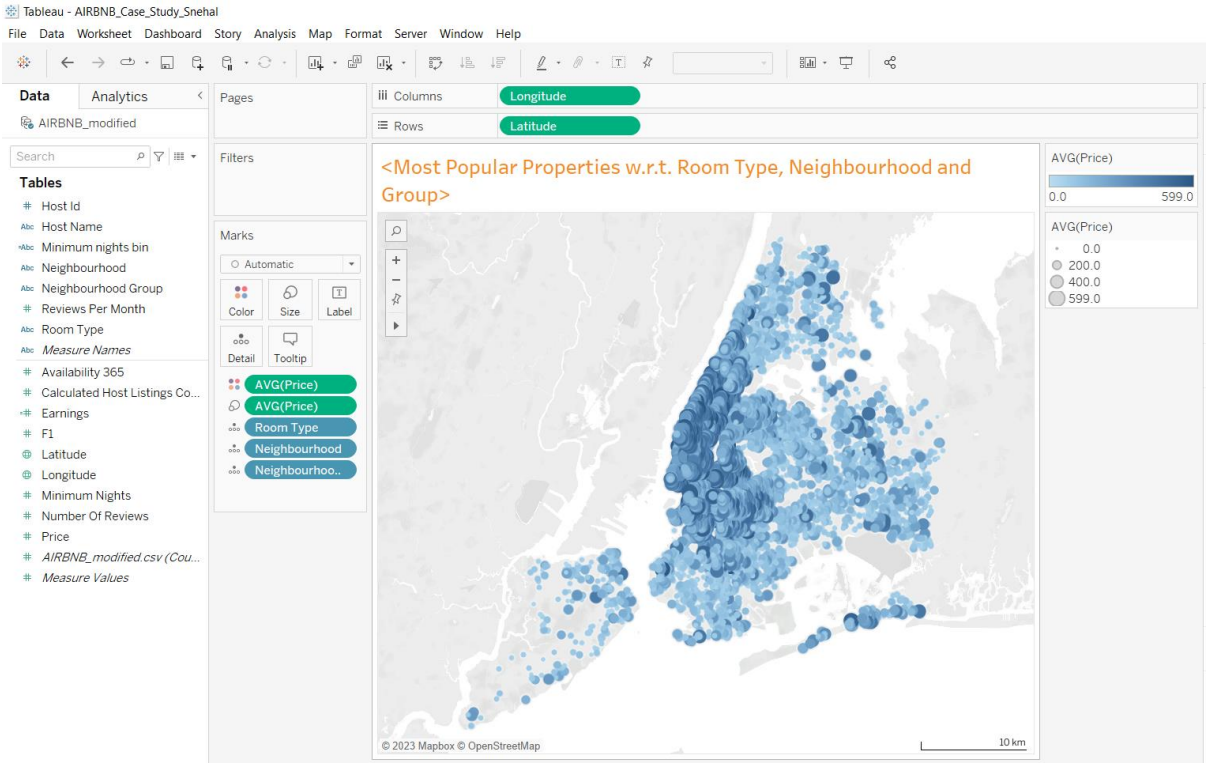
8. Popular localities and properties:

This map shows that Manhattan, Brooklyn, are popular than Queens, Bronx and Staten islands for entire home. They are shown as darker side in the map.



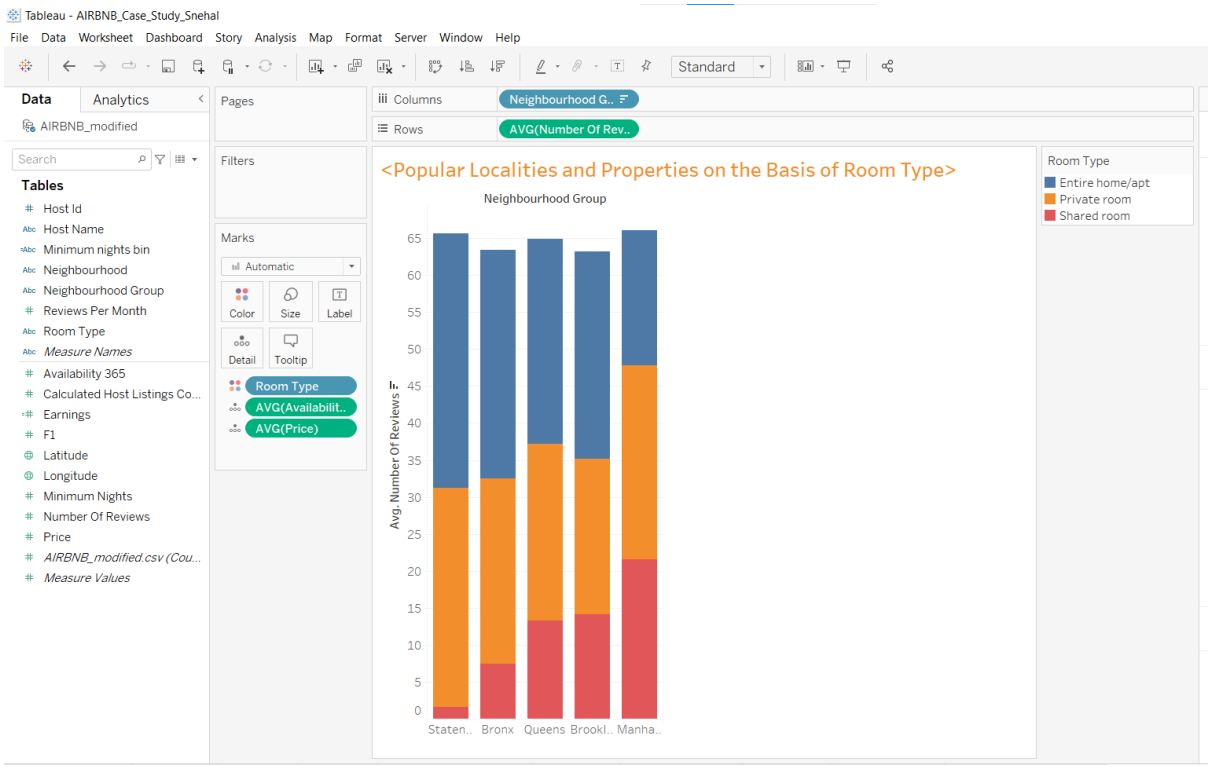
9. Most popular properties w.r.t room type, Neighbourhood and group:

According to the map Manhattan, Brooklyn, Queens are popular than Bronx and Staten islands for entire home. They are shown as darker side in the map.



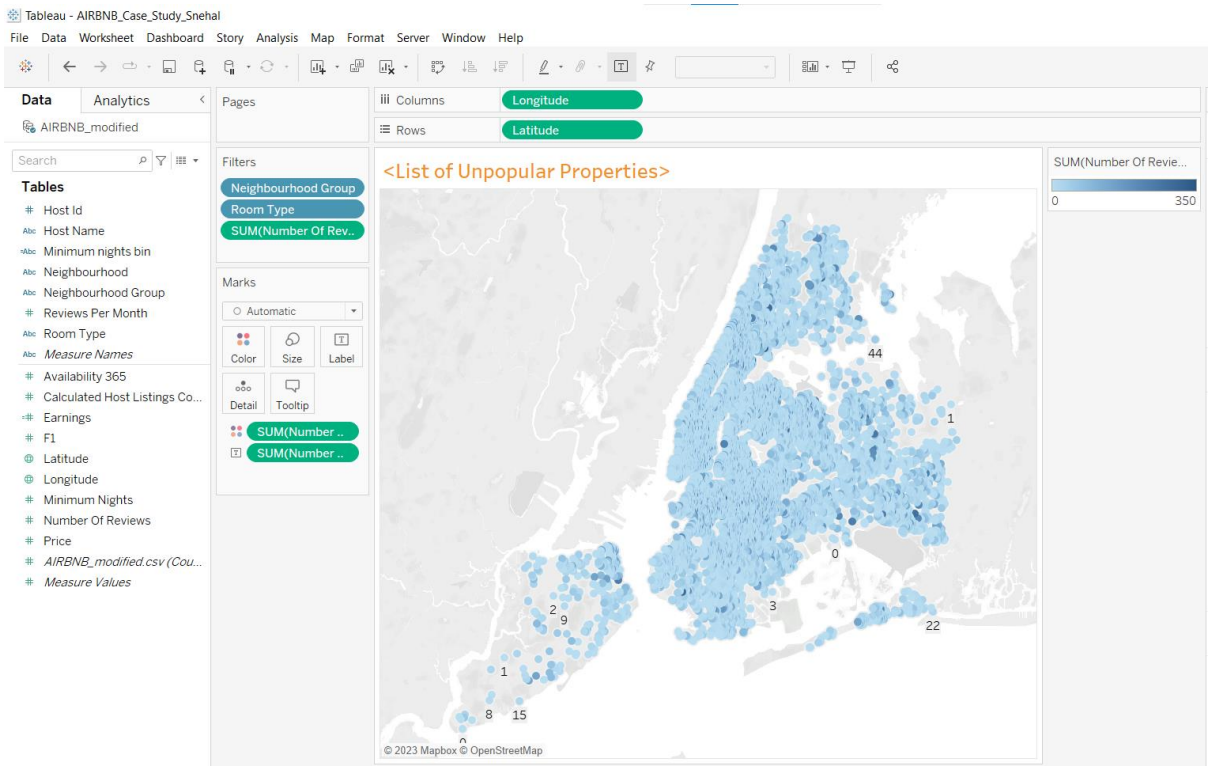
10. Popular localities and properties on basis of room type:

This graph, show which room type are preferred by customers in different Neighbourhood based on number of reviews and price. Only in Manhattan, shared room is preferred more as it is industrial area and it hub. So, the employees prefer it more.



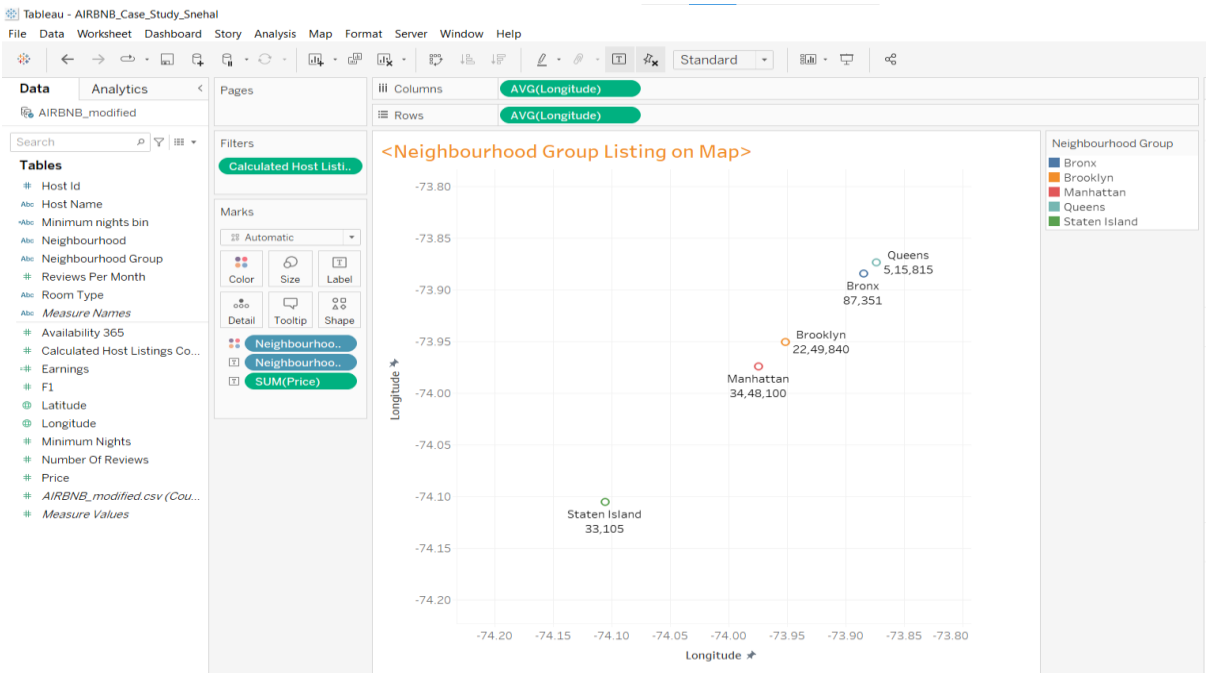
11. List of Unpopular Properties:

This map shows the top 10 unpopular properties as they are far from city and didn't have any tourist spots.



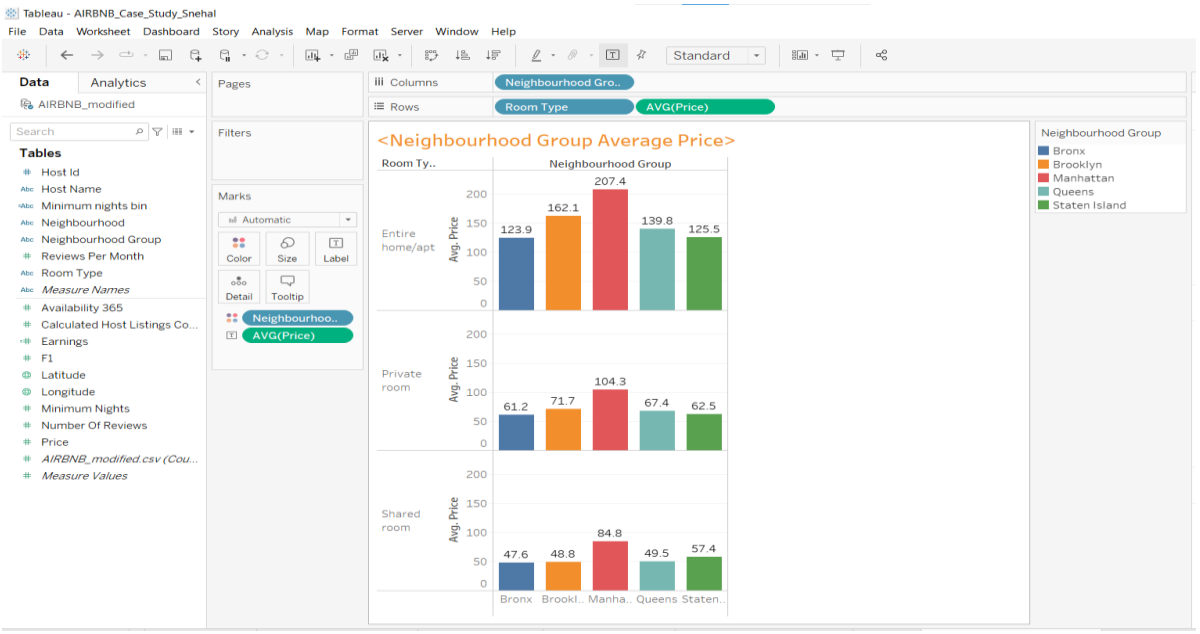
12. Neighbourhood group listing:

The map shows that Manhattan is higher and affordable for high class people for its tourist spots. Then Brooklyn is preferred by customers compare to Bronx and Queens.



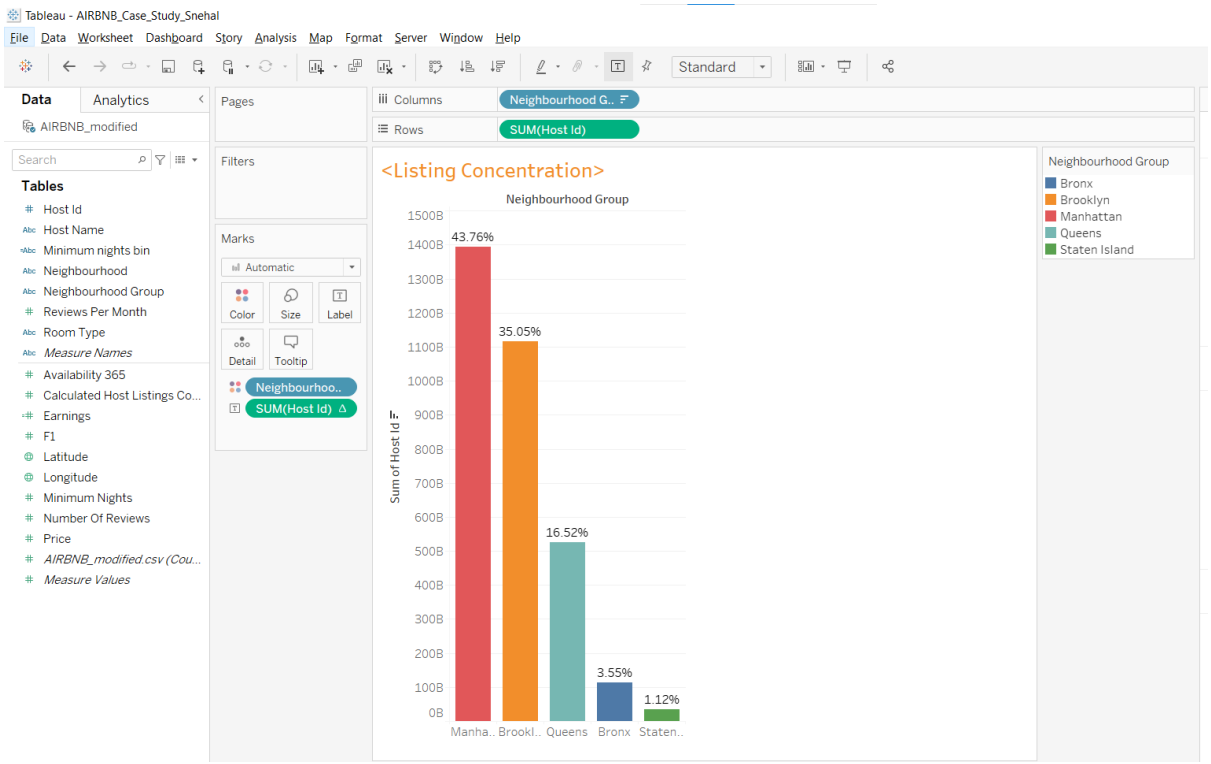
13. Neighbourhood group average price:

In this graph, Manhattan is preferred in all room types. And Bronx is least preferred mostly by customers in entire home and private rooms. In all the Neighbourhoods other than Manhattan they have to make new marketing strategies to increase their revenue.



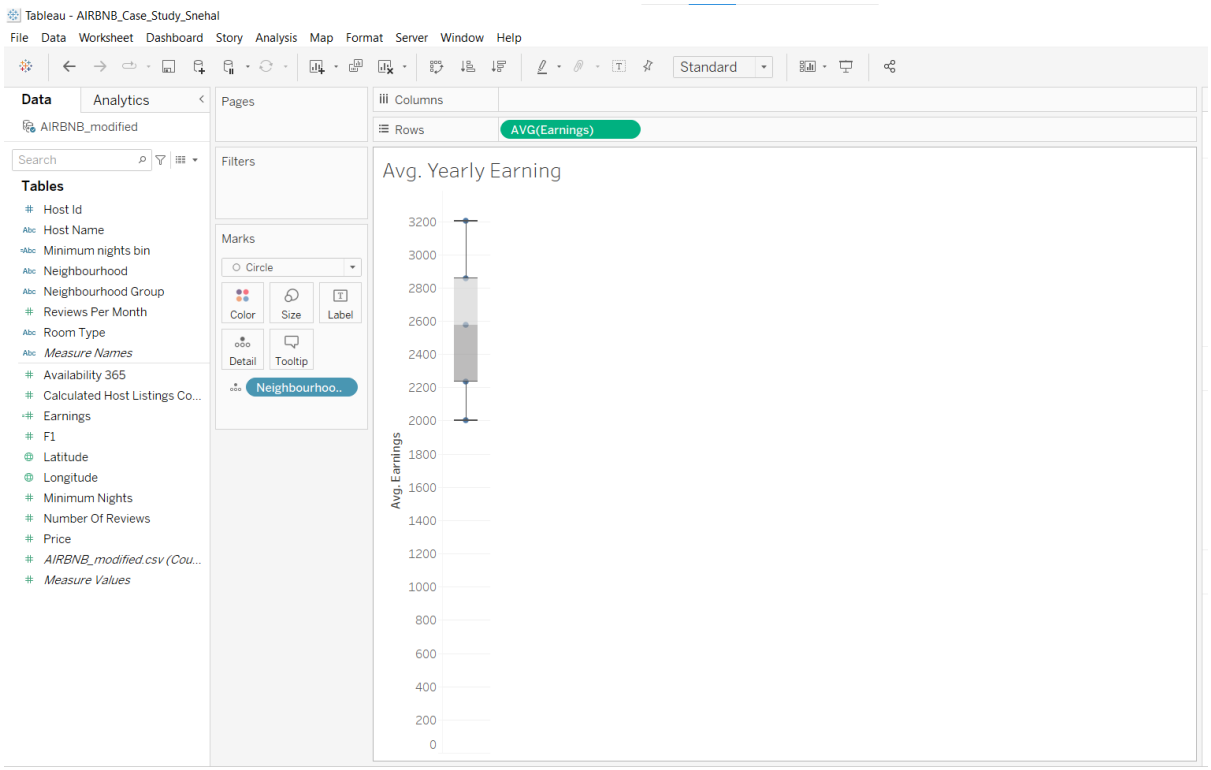
14. Listing Concentration in New York:

The Airbnb has good presence in Manhattan, Brooklyn, and Queens. Listings are maximum in Manhattan (43.76%), Brooklyn (35.05%) as they have highest population density and tourism spots.



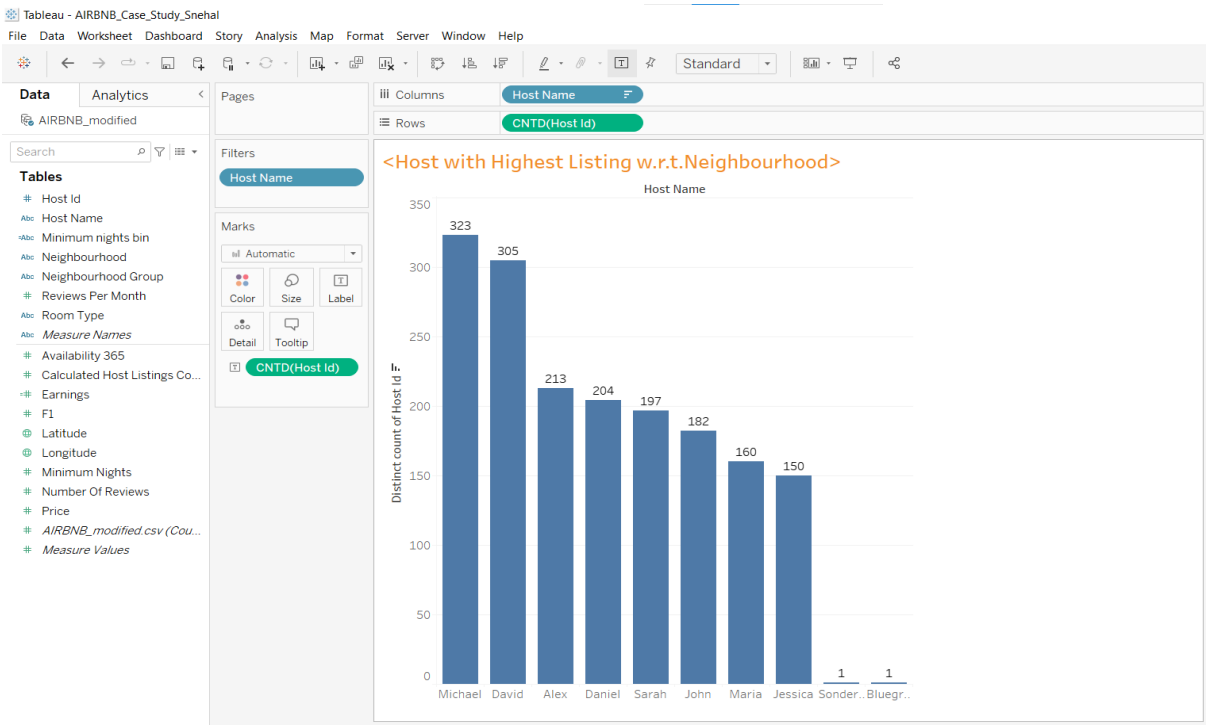
15. Average Yearly Earning:

This graph shows the yearly earning in each neighbourhood group.



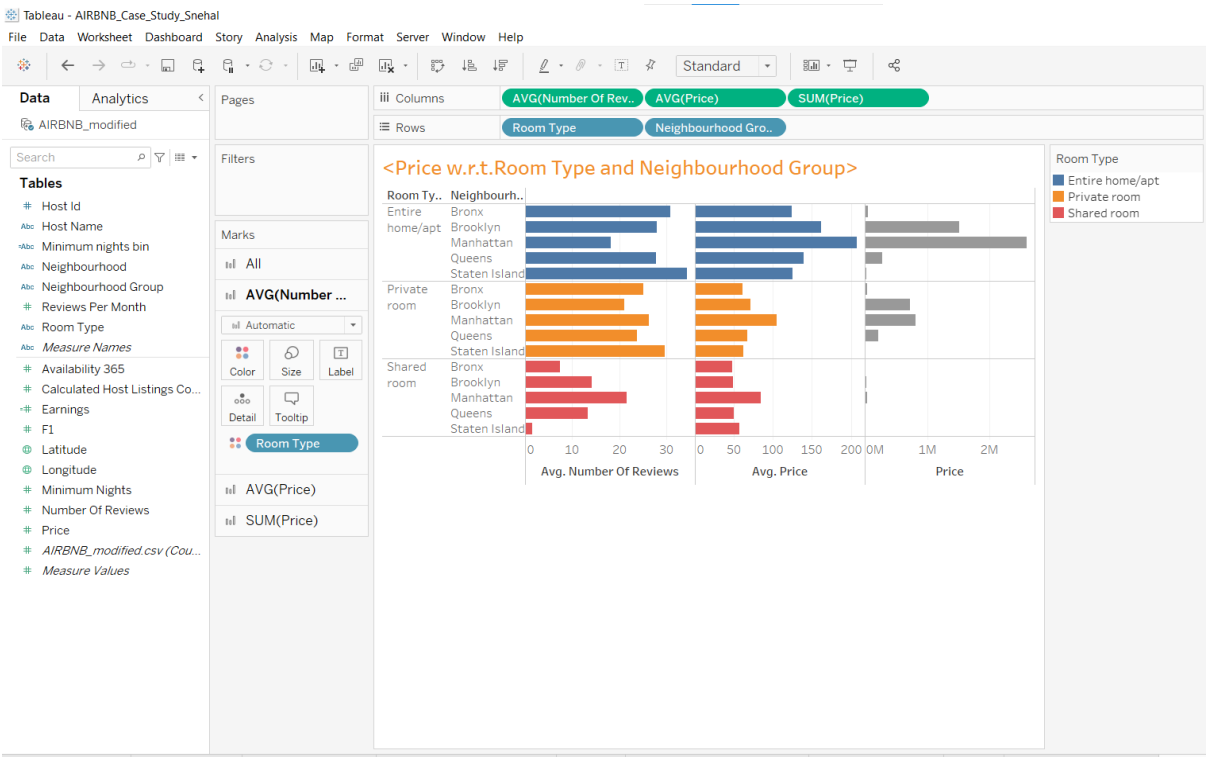
16. Host with highest listing in the Neighbourhood:

The graph shows Michael, David, Alex has more than 40% properties in New York and they earn more than others.

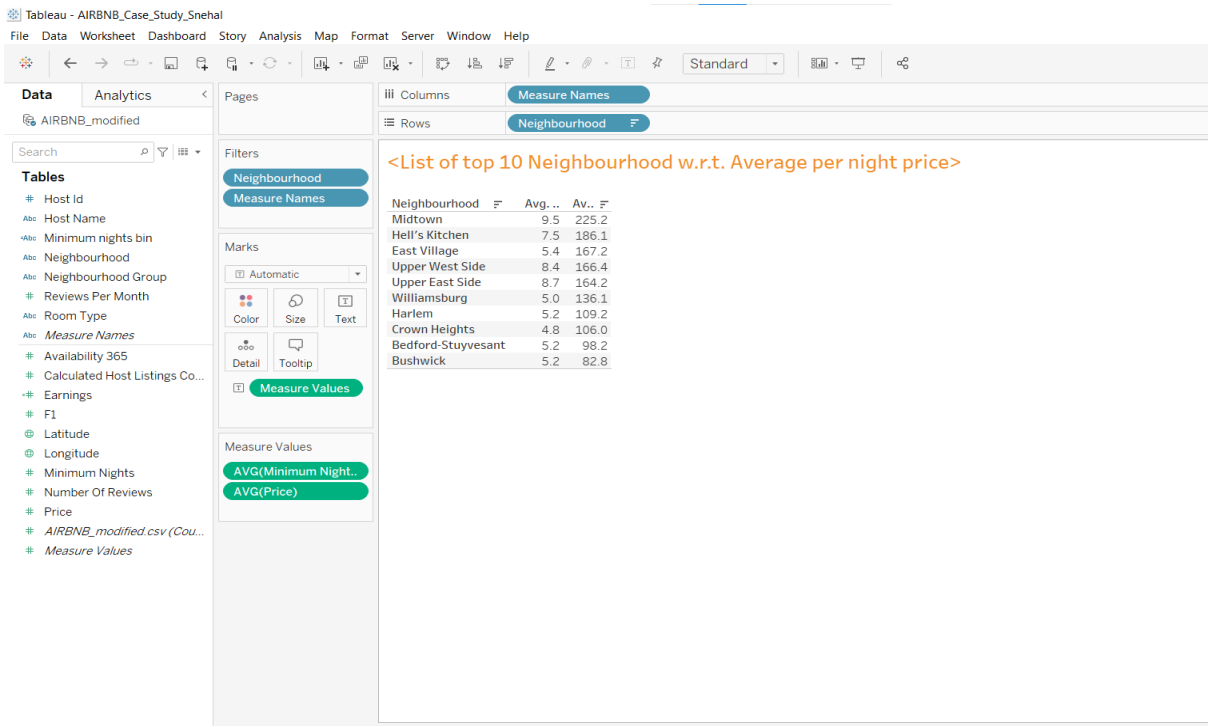


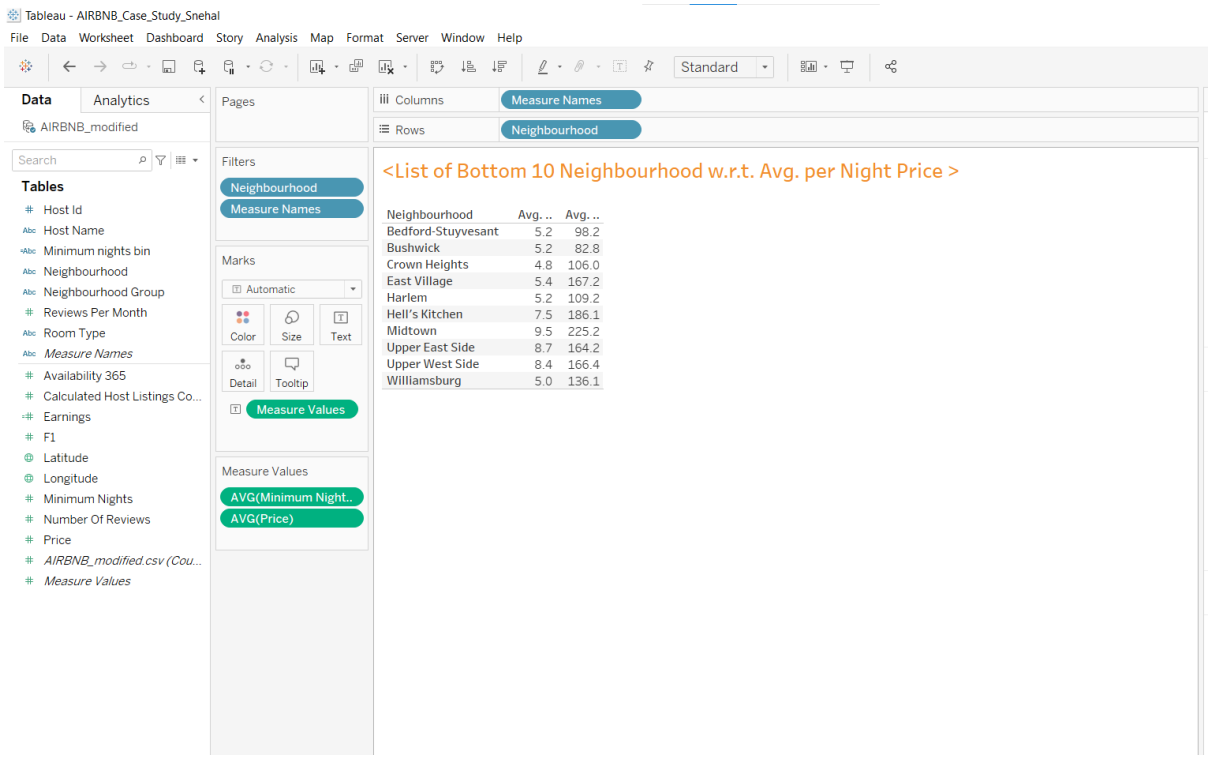
17. Price preferred by customer on basis of room type and neighbourhood:

In this graph, it shows that Manhattan and Brooklyn have upper hand in both entire home and private room.

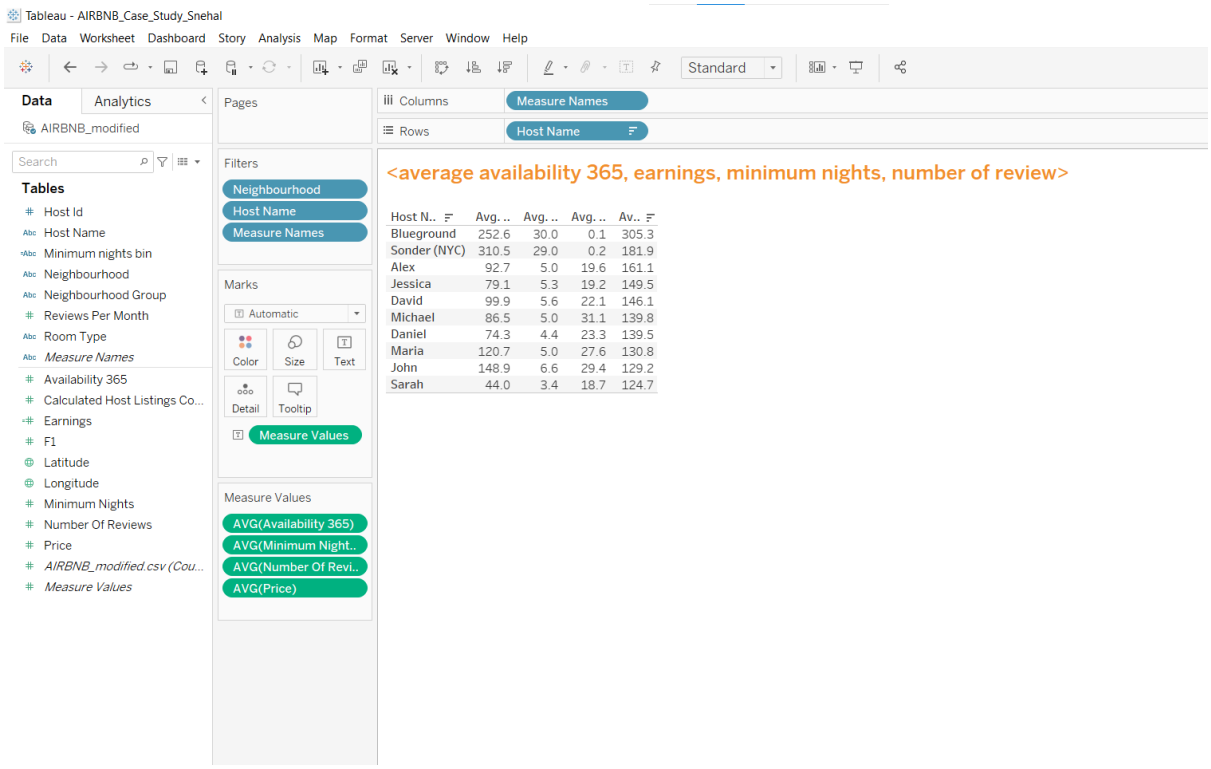


18. List of top and least 10 neighbourhood w.r.t average per night price: From below graphs, we can see Midtown is charging more price than others.





Then we have created a listing chart to show average availability 365, earnings, minimum nights, number of review and price w.r.t host name.



Then we have created dashboards and stories to get clear idea of the attributes and to give insights, suggestions to improve the business.

Then made the presentation and added recommendations.

Important Findings:

1. Michael is the top earner who is earning more and he belongs to Manhattan
2. Manhattan is the only Neighbourhood in the Borough that lies in offering the Highest Price range properties on the platform followed by others with a Medium Price range on average
3. Having a high price range, Entire home/apt types rooms are available for less than 100 days on average followed by Private rooms.
4. Manhattan has the highest number of places listed around more than 10 by a single host with an average price of 230 $ followed by Brooklyn with an average price of 108$.
5. Michael, David, Alex, John and Daniel are the Top 5 hosts that seem to have received the highest number of reviews for their listed sites and have also sites listed with High price range.
6. On an average Entire home/apt types are preferred more by the customers followed by Private rooms and then the Shared Rooms. Mostly because they are also available for a higher number of minimum night's stay window booking as compared to Private and Shared rooms.