Methodology Document

Submitted by: Cassia Rodrigues

Pieyush C Joy

August 2020 Batch for UpGrad

IMPORTING THE LIBRARIES

FOLLOWING LIBRARIES WERE IMPORTED FOR THE CASE STUDY:

- **NUMPY LIBRARY**
- **PANDAS LIBRARY**
- **MATPLOTLP AND SEASBORN LIBRARY**

```
# Importing the library
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

READING THE DATASET

```
# Reading the dataset
data=pd.read_csv('AB_NYC_2019.csv')
data.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	nu
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	

CHANGING THE SHAPE AND DATA TYPE OF THE COLUMNS

```
#checking the shape of the dataframe
data.shape
```

```
(48895, 16)
```

```
#checking type of every column in the dataset
data.dtypes
```

```
id                                 int64
name                              object
host_id                            int64
host_name                         object
neighbourhood_group               object
neighbourhood                     object
latitude                         float64
longitude                        float64
room_type                         object
price                              int64
minimum_nights                     int64
number_of_reviews                  int64
last_review                       object
reviews_per_month                float64
calculated_host_listings_count     int64
availability_365                   int64
dtype: object
```

DATA CLEANING-CHECKING THE NULL VALUES

```
#looking to find out first which columns have null values
#using 'sum' function will show us how many nulls are found in each column in dataset
data.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
```

DATA CLEANING-DROPPING COLUMNS

```
#dropping the non-significant columns
data.drop(['id','host_name','last_review'], axis=1, inplace=True)
#examing the changes
data.head(3)
```

	name	host_id	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	revie
0	Clean & quiet apt home by the park	2787	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	
1	Skylit Midtown Castle	2845	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	
2	THE VILLAGE OF HARLEM....NEW YORK !	4632	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	

DATA CLEANING-FILLING THE NULL VALUES & OUTLIER ANALYSIS

```
#replacing all NaN values in 'reviews_per_month' with 0
data.fillna({'reviews_per_month':0}, inplace=True)
#examing changes
data.reviews_per_month.isnull().sum()
```

```
0
```

```
data.describe()
```

	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	a
count	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	
mean	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.090910	7.143982	
std	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.597283	32.952519	
min	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.000000	1.000000	
25%	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.040000	1.000000	
50%	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.370000	1.000000	
75%	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	1.580000	2.000000	
max	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.000000	

EXAMINING THE UNIQUE VALUES FOR THE CATEGORICAL VARAIBALES

```
#examining the unique values of neighbourhood column
data.neighbourhood.value_counts()
```

```
Williamsburg           3920
Bedford-Stuyvesant     3714
Harlem                 2658
Bushwick               2465
Upper West Side        1971
                       ...
Richmondtown              1
Fort Wadsworth            1
New Dorp                  1
Woodrow                   1
Willowbrook               1
Name: neighbourhood, Length: 221, dtype: int64
```

```
#examining the unique values of room_type column
data.room_type.value_counts()
```

```
Entire home/apt    25409
Private room       22326
Shared room         1160
Name: room_type, dtype: int64
```

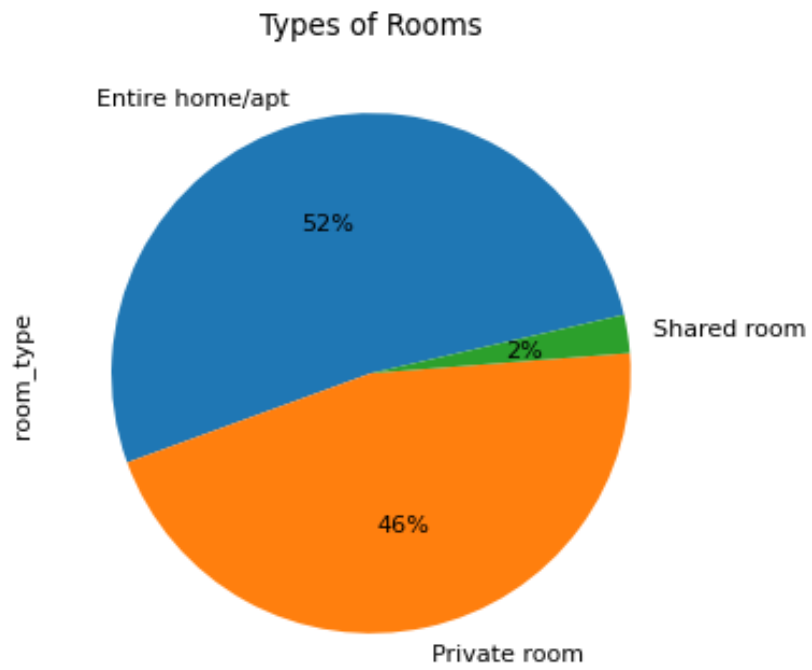
DATA VISUALIZATIONS

EXAMINING THE ROOM TYPE COLUMN

We see that around 52% of the population choose entire home/apt room type, 46% people choose private room and only 2% of the population choose shared room.

```
fig = plt.figure(figsize=(5,5), dpi=80)
data['room_type'].value_counts().plot(kind='pie', autopct='%1.0f%%', startangle=13, title='Types of Rooms')

plt.show()
```



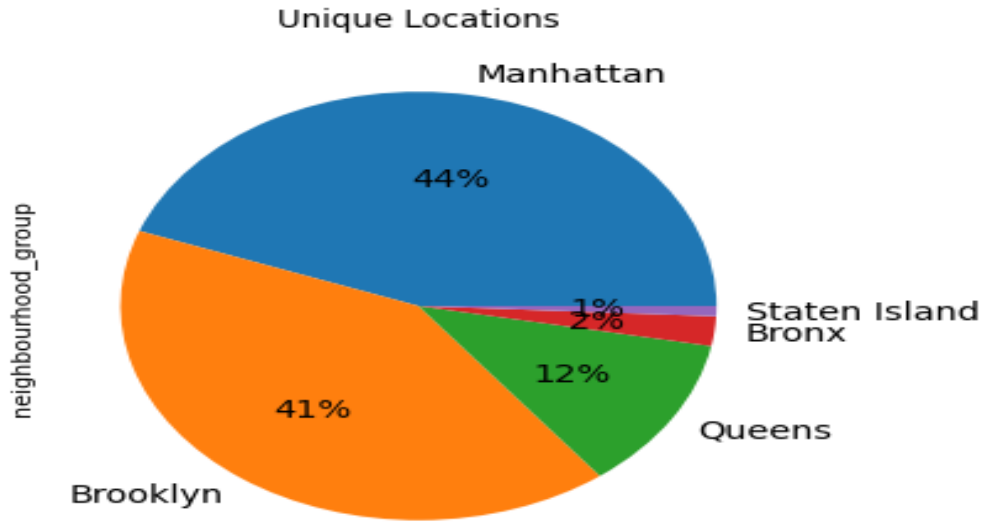
EXAMINING THE ROOM NEIGHBOURHOOD GROUP COLUMN

We see that many people are attracted towards the city of Manhattan whereas the least number of people prefer Staten Island.

```
#There are 5 particular neighborhood_group, which means 5 unique locations
data['neighbourhood_group'].value_counts()
```

```
Manhattan         21661
Brooklyn          20104
Queens             5666
Bronx              1091
Staten Island       373
Name: neighbourhood_group, dtype: int64
```

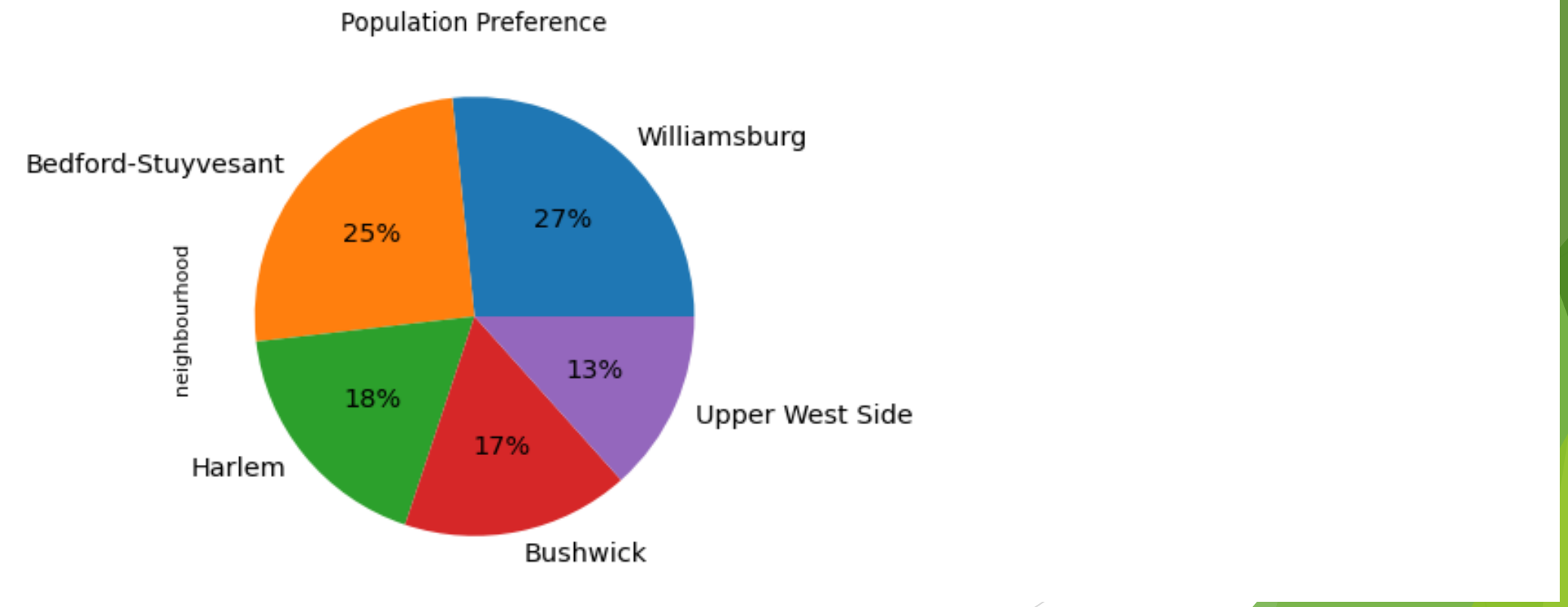
```
fig = plt.figure(figsize=(5,5), dpi=80)
data['neighbourhood_group'].value_counts().plot(kind ='pie', autopct='%1.0f%%', startangle= 360, fontsize=13, titl
plt.show()
```



EXAMINING THE ROOM NEIGHBOURHOOD COLUMN

We see that many people prefer Williamsburg neighbourhood followed by Bedford-Stuyvesant neighbourhood.

```
fig=plt.figure(figsize=(5,5), dpi=80)
data['neighbourhood'].value_counts().iloc[:5].plot(kind='pie', autopct='%1.0f%%', startangle=360, fontsize=13,
plt.show()
```



EXAMINING THE PRICE COLUMN

It is observed that the average price of Airbnb is 152$ approximately and the costliest Airbnb is 10,000$.

```
data.price.describe()
```

```
count    48895.000000
mean       152.720687
std        240.154170
min          0.000000
25%         69.000000
50%        106.000000
75%        175.000000
max      10000.000000
Name: price, dtype: float64
```

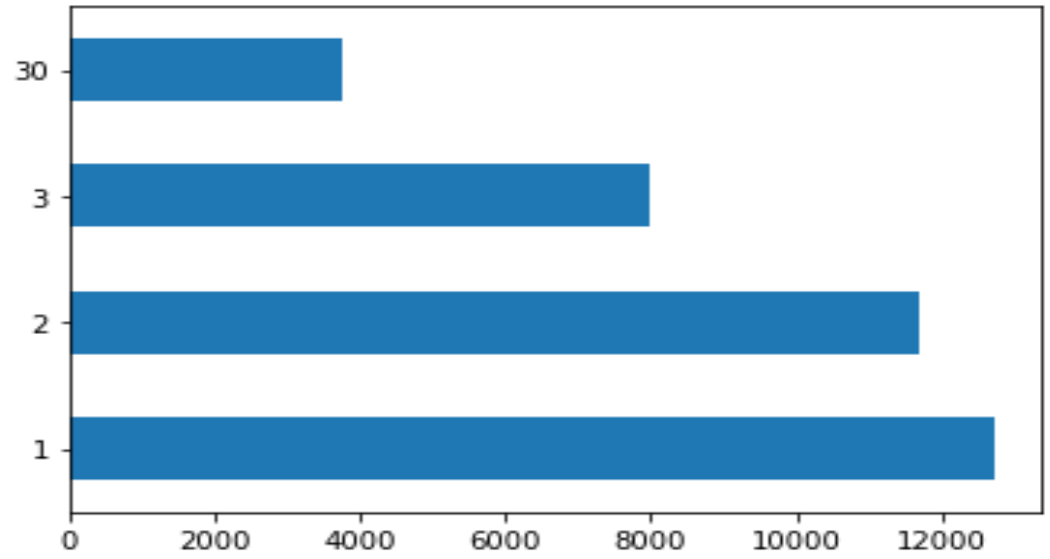
EXAMINING THE MINIMUM NIGHTS COLUMN

It is observed that around 12,000 people used 1 night stay and around 11,000 people used 2 night stay in Airbnb.

```
data['minimum_nights'].value_counts().head(4)
```

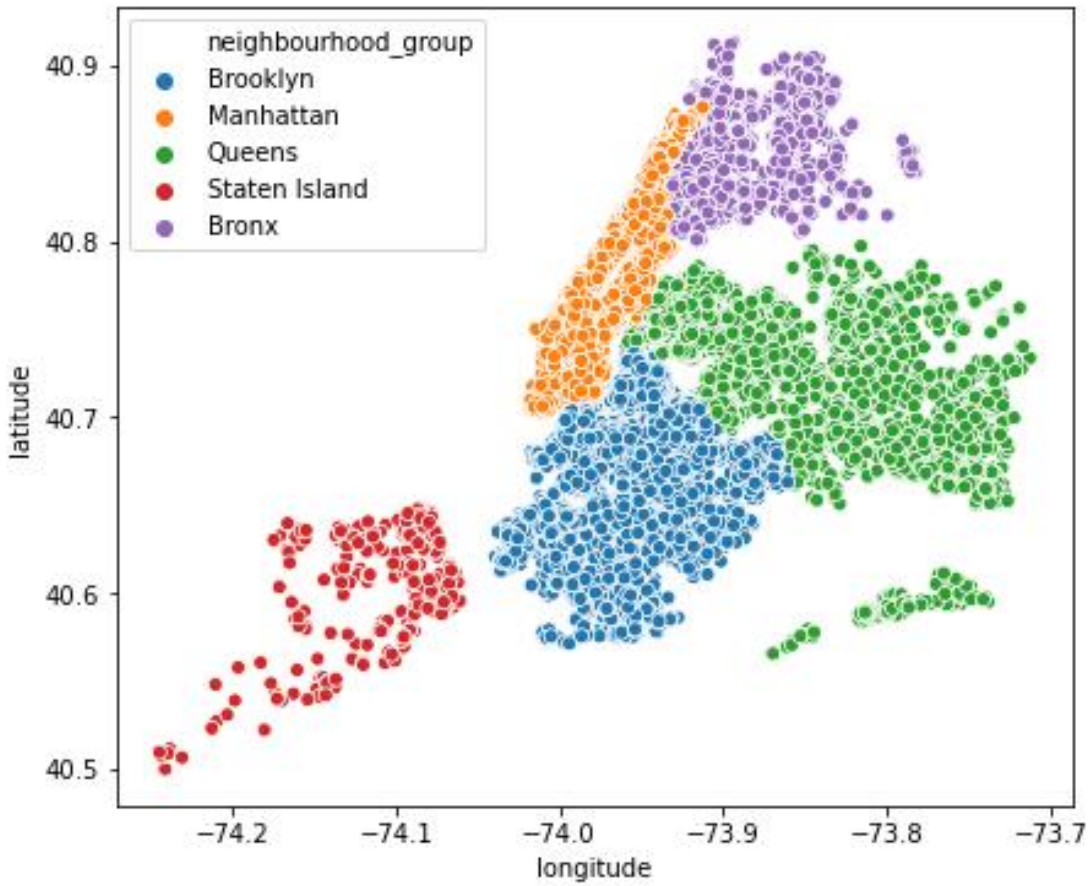
```
1     12720
2     11696
3      7999
30     3760
Name: minimum_nights, dtype: int64
```

```
data['minimum_nights'].value_counts().head(4).plot(kind='barh')
plt.show()
```



EXAMINING THE NEIGHBOURHOOD USING MAP

```
plt.figure(figsize=(7,6))
sns.scatterplot(data.longitude, data.latitude, hue=data.neighbourhood_group)
plt.show()
```

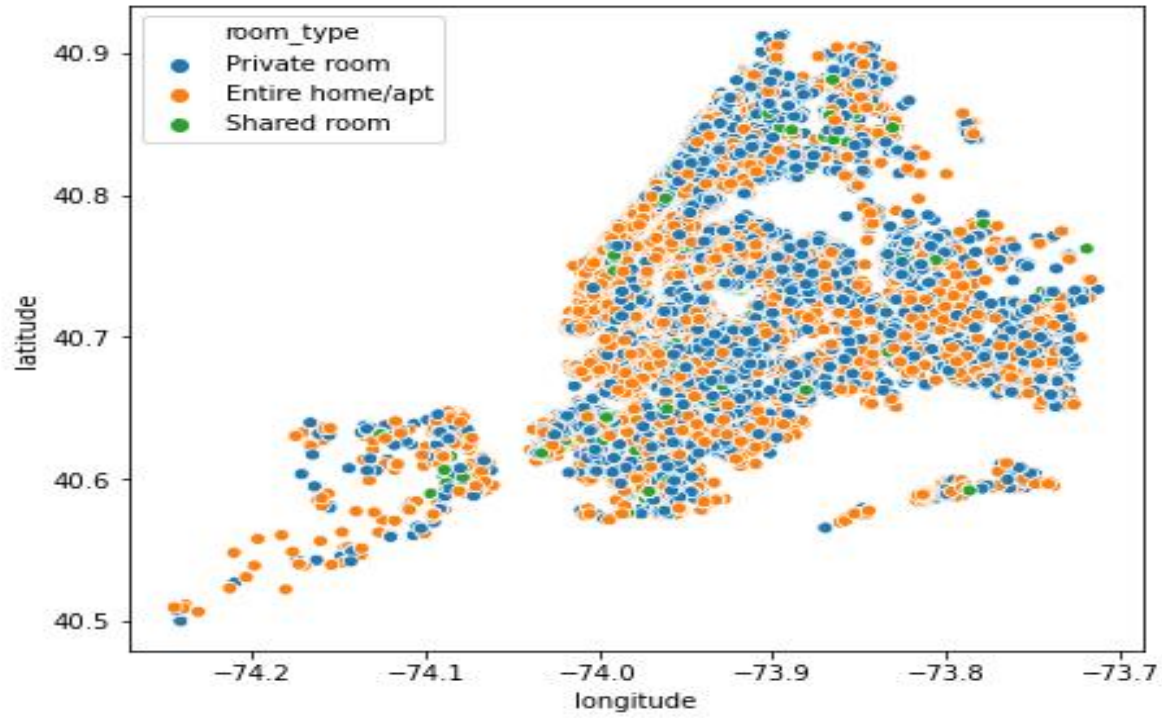


ROOM TYPES AMONG THE 5 BOROUGHS

```
data['room_type'].value_counts()
```

```
Entire home/apt    25409
Private room       22326
Shared room         1160
Name: room_type, dtype: int64
```

```
plt.figure(figsize=(7,6))
sns.scatterplot(data.longitude, data.latitude, hue=data.room_type)
plt.show()
```



THANK YOU

Submitted by:

Cassia Rodrigues

Pieyush C Joy