AIRBNB Case Study IIIT-B Harish DV

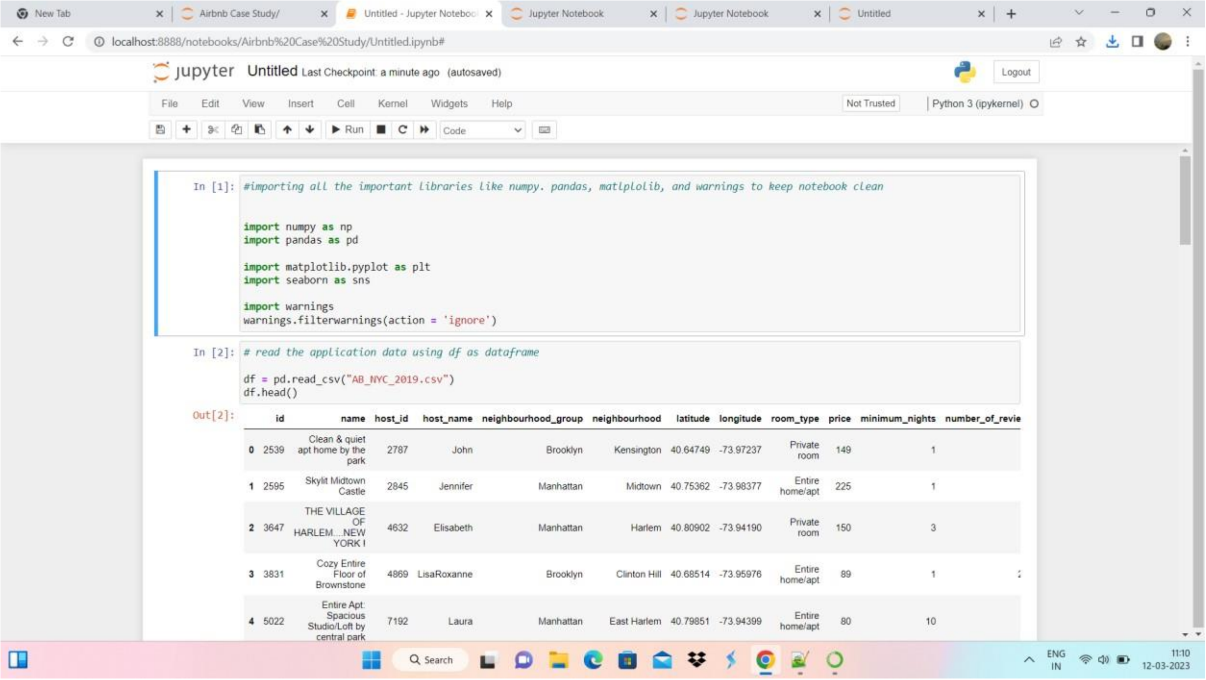
Methodology Document PPT 1:

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: AB_NYC_2019.csv

Number of Rows: 48895

Number of Columns: 16

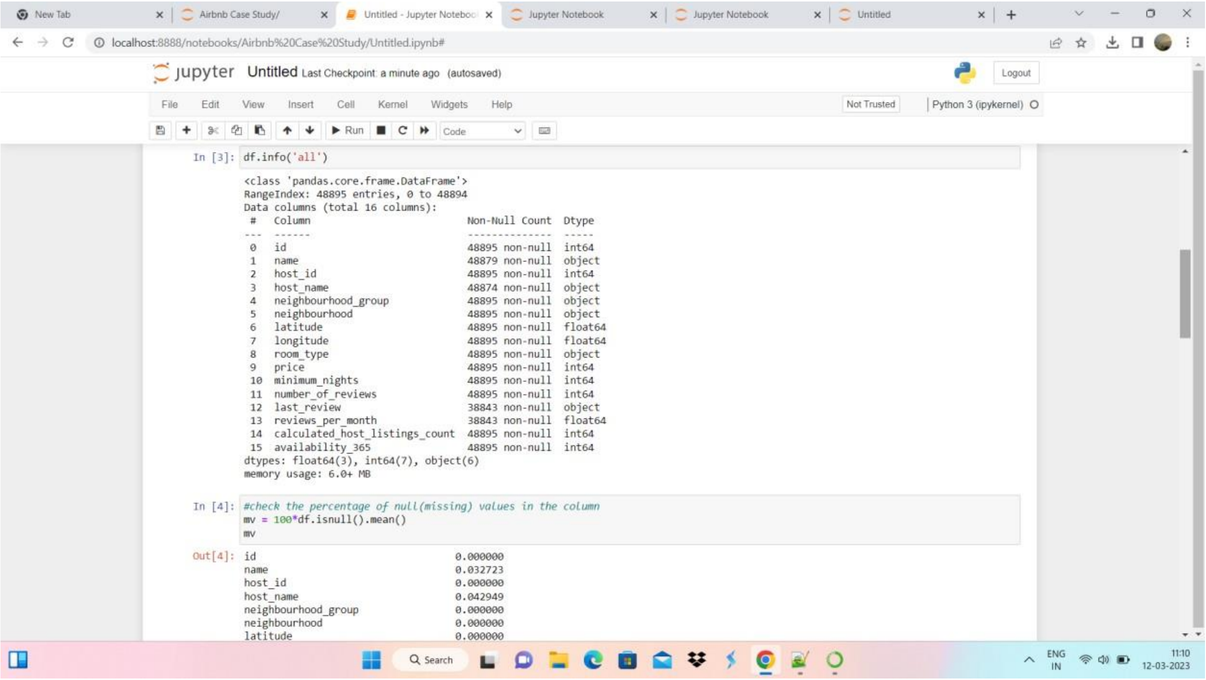

New Tab
Airbnb Case Study/
Untitled - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
Untitled
localhost:8888/notebooks/Airbnb%20Case%20Study/Untitled.ipynb#
jupyter Untitled Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
11:10
12-03-2023

```
In [1]: #importing all the important libraries like numpy. pandas, matlplolib, and warnings to keep notebook clean


        import numpy as np
        import pandas as pd

        import matplotlib.pyplot as plt
        import seaborn as sns

        import warnings
        warnings.filterwarnings(action = 'ignore')
```

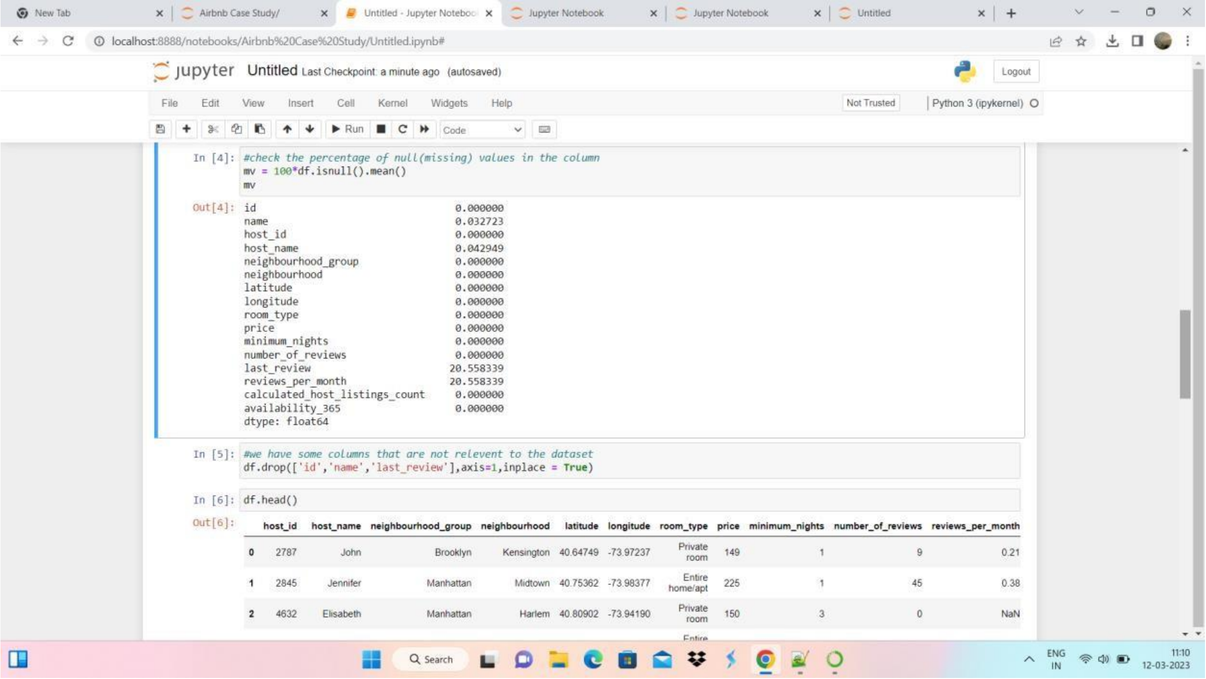
```
In [2]: # read the application data using df as dataframe

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

Out[2]:

	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	

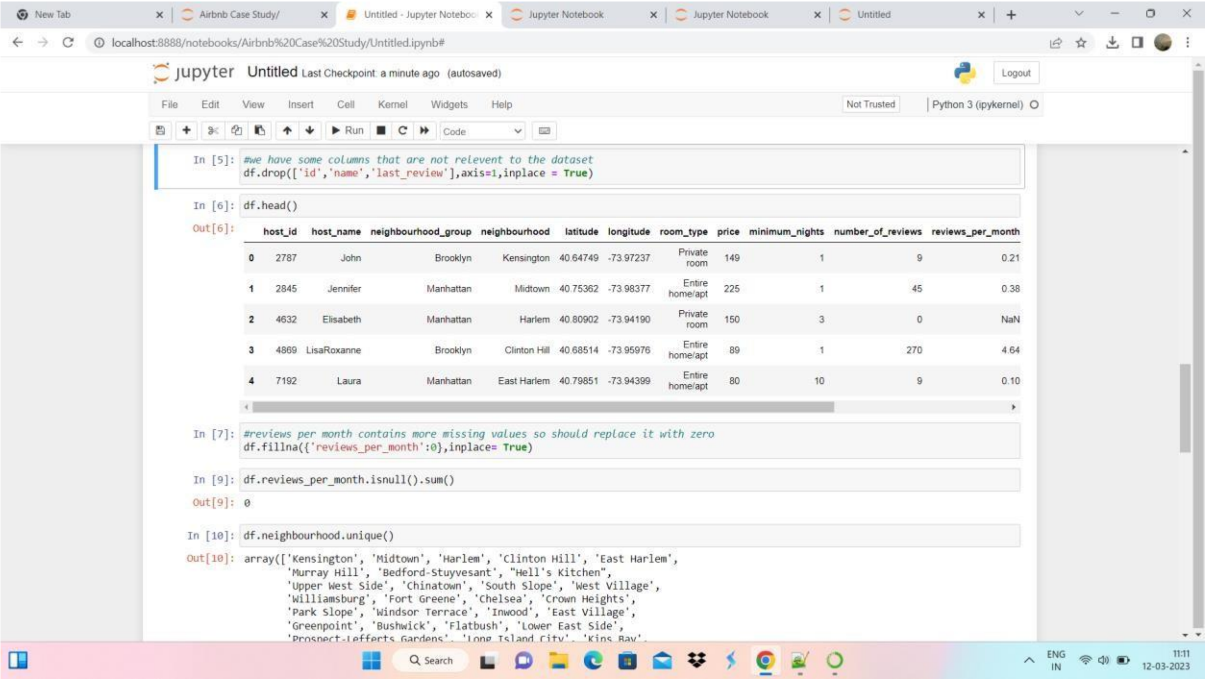

New Tab
Airbnb Case Study/
Untitled - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
Untitled
localhost:8888/notebooks/Airbnb%20Case%20Study/Untitled.ipynb#
jupyter Untitled Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
11:10
12-03-2023

```
In [3]: df.info('all')

        <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 [4]: #check the percentage of null(missing) values in the column
        mv = 100*df.isnull().mean()
        mv

Out[4]: id                                0.000000
        name                              0.032723
        host_id                           0.000000
        host_name                         0.042949
        neighbourhood_group               0.000000
        neighbourhood                     0.000000
        latitude                          0.000000
```


New Tab
Airbnb Case Study/
Untitled - Jupyter Notebook
Jupyter Notebook
Jupyter Notebook
Untitled
localhost:8888/notebooks/Airbnb%20Case%20Study/Untitled.ipynb#
jupyter Untitled Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Search
ENG IN
11:10
12-03-2023

```
In [4]: #check the percentage of null(missing) values in the column
        mv = 100*df.isnull().mean()
        mv

Out[4]: 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

In [5]: #we have some columns that are not relevent to the dataset
        df.drop(['id','name','last_review'],axis=1,inplace = True)

In [6]: df.head()

Out[6]:
```

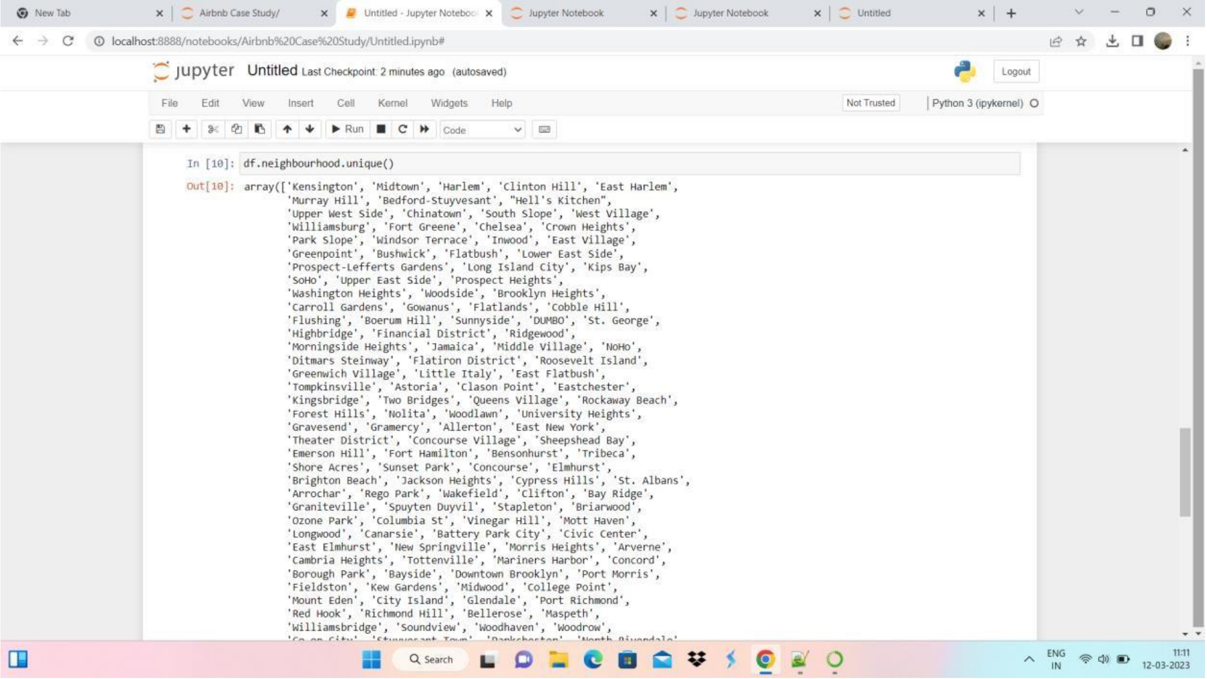
	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_month
0	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	0.21
1	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	0.38
2	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	NaN
							Entire				



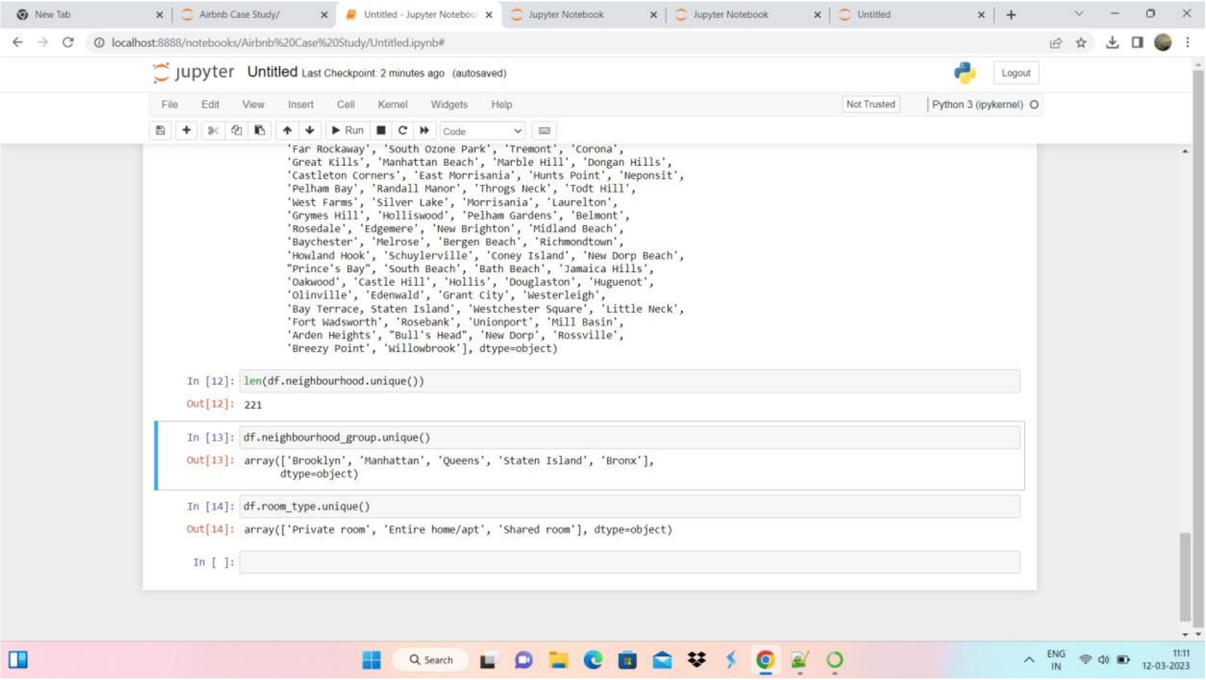
```
In [5]: #we have some columns that are not relevent to the dataset
        df.drop(['id','name','last_review'],axis=1,inplace = True)

In [6]: df.head()

Out[6]:
```

	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 [7]: #reviews per month contains more missing values so should replace it with zero
        df.fillna({'reviews_per_month':0},inplace= True)

In [9]: df.reviews_per_month.isnull().sum()

Out[9]: 0

In [10]: df.neighbourhood.unique()

Out[10]: array(['Kensington', 'Midtown', 'Harlem', 'Clinton Hill', 'East Harlem',
                'Murray Hill', 'Bedford-Stuyvesant', "Hell's Kitchen",
                'Upper West Side', 'Chinatown', 'South Slope', 'West Village',
                'Williamsburg', 'Fort Greene', 'Chelsea', 'Crown Heights',
                'Park Slope', 'Windsor Terrace', 'Inwood', 'East Village',
                'Greenpoint', 'Bushwick', 'Flatbush', 'Lower East Side',
                'Prospect-Lefferts Gardens', 'Long Island City', 'Kips Bay',
```



```
In [10]: df.neighbourhood.unique()

Out[10]: array(['Kensington', 'Midtown', 'Harlem', 'Clinton Hill', 'East Harlem',
                'Murray Hill', 'Bedford-Stuyvesant', "Hell's Kitchen",
                'Upper West Side', 'Chinatown', 'South Slope', 'West Village',
                'Williamsburg', 'Fort Greene', 'Chelsea', 'Crown Heights',
                'Park Slope', 'Windsor Terrace', 'Inwood', 'East Village',
                'Greenpoint', 'Bushwick', 'Flatbush', 'Lower East Side',
                'Prospect-Lefferts Gardens', 'Long Island City', 'Kips Bay',
                'SoHo', 'Upper East Side', 'Prospect Heights',
                'Washington Heights', 'Woodside', 'Brooklyn Heights',
                'Carroll Gardens', 'Gowanus', 'Flatlands', 'Cobble Hill',
                'Flushing', 'Boerum Hill', 'Sunnyside', 'DUMBO', 'St. George',
                'Highbridge', 'Financial District', 'Ridgewood',
                'Morningside Heights', 'Jamaica', 'Middle Village', 'NoHo',
                'Ditmars Steinway', 'Flatiron District', 'Roosevelt Island',
                'Greenwich Village', 'Little Italy', 'East Flatbush',
                'Tompkinsville', 'Astoria', 'Clason Point', 'Eastchester',
                'Kingsbridge', 'Two Bridges', 'Queens Village', 'Rockaway Beach',
                'Forest Hills', 'Nolita', 'Woodlawn', 'University Heights',
                'Gravesend', 'Gramercy', 'Allerton', 'East New York',
                'Theater District', 'Concourse Village', 'Sheepshead Bay',
                'Emerson Hill', 'Fort Hamilton', 'Bensonhurst', 'Tribeca',
                'Shore Acres', 'Sunset Park', 'Concourse', 'Elmhurst',
                'Brighton Beach', 'Jackson Heights', 'Cypress Hills', 'St. Albans',
                'Arrochar', 'Rego Park', 'Wakefield', 'Clifton', 'Bay Ridge',
                'Graniteville', 'Spuyten Duyvil', 'Stapleton', 'Briarwood',
                'Ozone Park', 'Columbia St', 'Vinegar Hill', 'Mott Haven',
                'Longwood', 'Canarsie', 'Battery Park City', 'Civic Center',
                'East Elmhurst', 'New Springville', 'Morris Heights', 'Arverne',
                'Cambria Heights', 'Tottenville', 'Mariners Harbor', 'Concord',
                'Borough Park', 'Bayside', 'Downtown Brooklyn', 'Port Morris',
                'Fieldston', 'Kew Gardens', 'Midwood', 'College Point',
                'Mount Eden', 'City Island', 'Glendale', 'Port Richmond',
                'Red Hook', 'Richmond Hill', 'Bellerose', 'Maspeth',
                'Williamsbridge', 'Soundview', 'Woodhaven', 'Woodrow',
                'Co-op City', 'Stuyvesant Town', 'Parkchester', 'North Riverdale',
```



```
       'Far Rockaway', 'South Ozone Park', 'Tremont', 'Corona',
       'Great Kills', 'Manhattan Beach', 'Marble Hill', 'Dongan Hills',
       'Castleton Corners', 'East Morrisania', 'Hunts Point', 'Neponsit',
       'Pelham Bay', 'Randall Manor', 'Throgs Neck', 'Todt Hill',
       'West Farms', 'Silver Lake', 'Morrisania', 'Laurelton',
       'Grymes Hill', 'Holliswood', 'Pelham Gardens', 'Belmont',
       'Rosedale', 'Edgemere', 'New Brighton', 'Midland Beach',
       'Baychester', 'Melrose', 'Bergen Beach', 'Richmondtown',
       'Howland Hook', 'Schuylerville', 'Coney Island', 'New Dorp Beach',
       "Prince's Bay", 'South Beach', 'Bath Beach', 'Jamaica Hills',
       'Oakwood', 'Castle Hill', 'Hollis', 'Douglaston', 'Huguenot',
       'Olinville', 'Edenwald', 'Grant City', 'Westerleigh',
       'Bay Terrace, Staten Island', 'Westchester Square', 'Little Neck',
       'Fort Wadsworth', 'Rosebank', 'Unionport', 'Mill Basin',
       'Arden Heights', "Bull's Head", 'New Dorp', 'Rossville',
       'Breezy Point', 'Willowbrook'], dtype=object)

In [12]: len(df.neighbourhood.unique())

Out[12]: 221

In [13]: df.neighbourhood_group.unique()

Out[13]: array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
               dtype=object)

In [14]: df.room_type.unique()

Out[14]: array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)

In [ ]:
```

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. Top 10 Host:

- We identified the top 10 Host Ids, Host Name with count of Host Ids using the tree map.

2. Neighbourhoods for Airbnb to Target:

- We created a pie chart to know neighbourhood for Airbnb to target using minimum nights and number of reviews

- We have added Neighbourhood in colours Marks card to highlight different minimum nights and number of reviews

3. Price Range Preferred By Customers:

- We have use packed bubbles for plot with count of id's with price(bin).
- We have create a bin for a span of $20.

4. Price of Room Type w.r.t. Neighbourhood Group:

- We have created box and whisker plot with average price in row and room type in column
- We added the Neighbourhood Groups in colours Marks card to highlight the different Neighbourhood Groups in different colours.

5. Average Room Price w.r.t. Number of Reviews:

- We have created a tree map with average price and room type.
- We added the Number of reviews in colours Marks card to highlight the different Number of reviews in different colours.

6. Popular and Unpopular Neighbourhood Groups:

- We have taken symbol maps plots with average longitude in column and average latitude in rows
- We added the Neighbourhood Groups in colours Marks card to highlight the different Neighbourhood Groups in different colours.

Methodology PPT2:

1. Top 10 Hosts:

- We identified the top 10 Host Ids, Host Name with count of Host Ids using the tree map.

2. Neighbourhoods for Airbnb to Target:

- We created a Horizontal Bar chart to know neighbourhood for Airbnb to target using minimum nights and number of reviews
- We have added Number of reviews in colours Marks card to highlight Neighbourhoods and Minimum Nights.

3. Price Range Preferred By Customers:

- We have use Horizontal Bar chart for plot with count of id's with price(bin).
- We have create a bin for a span of $20.

4. Price of Room Type w.r.t. Neighbourhood Group:

- We have created Circle views plot with Median price in row and Neighbourhood Groups in column
- We added the Room type in colours Marks card to highlight the different Neighbourhood Groups in different colours.

5. Average Room Price w.r.t. Number of Reviews:

- We have created a view circles with average price and room type.
- We added the Number of reviews in colours Marks card to highlight the different Number of reviews in different colours.

6. Popular Neighbourhood Groups:

- We have taken Horizontal Bar chart plots with count of availability and Neighbourhood.
- We added the Number of reviews in colours Marks card to highlight the different Neighbourhood in different colours.

7. Tools used:

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