Storytelling Case Study: Airbnb, NYC Methodology Document

Team Members: - Kiran Ghadigaonkar, Prince Chaturvedi, Aadesh Birhade

We have used **Jupiter Notebook** to perform **Data Preparation** and **Tableau** for **Data Analysis and Visualization** to get better insights for this case study.

Dataset used: AB_NYC_2019.csv

Tools used:

- **Data Preparation**: Jupyter Notebook – Python
- **Visualization and analysis:** Tableau
- **Data Storytelling:** Microsoft PPT

We followed the below steps for Data preparation:

- **Data Understanding**
- **Data Cleaning:** To identify and remove any missing values and duplicate values and dropped insignificant columns.
- **Outliers Treatment:** Identified outliers

In [1]:
```
#Let's import the necessary libraries
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
#Now let's understand the data
airbnb =pd.read_csv("AB_NYC_2019.csv")
airbnb.head(5)
```

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	

In [3]:
```
#Let's Check the shape of the dataset
airbnb.shape
```

Out[3]: (48895, 16)

No. of Rows: 48895

No. of Columns: 16

In [4]:
```
# Calculating the missing values in the dataset
airbnb.isnull().sum()
```

Out[4]:
```
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 [5]:
```
#Since, we have the missing values and there are some columns that are not efficient for the analysis, let's drop them.
airbnb.drop(['id','name','last_review'], axis = 1, inplace = True)
```

In [6]:
```
#Now let's check if columns are dropped
airbnb.head(5)
```

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	Lisa#roxanne	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]:
```
#Since, reviews_per_month contains maximum missing values, let's replace them with 0.
airbnb.fillna({'reviews_per_month':0},inplace=True)
```

In [8]:
```
#Now let's again check if null values are present for reviews_per_month column.
airbnb.reviews_per_month.isnull().sum()
```

Out[8]:
```
0
```

As we can see that, now there are no missing values present in reviews_per_month column.

As we can see that, now there are no missing values present in reviews_per_month column.

In [9]:
```
#Now Let's check for the unique values in room_type column.
airbnb.room_type.unique()
```

Out[9]:
```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

In [10]:
```
#Now Let's check for count of the unique values.
len(airbnb.room_type.unique())
```

Out[10]:
```
3
```

In [11]:
```
#Now Let's check for the unique values in neighbourhood_group column.
airbnb.neighbourhood_group.unique()
```

Out[11]:
```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

In [12]:
```
#Now Let's check for count of the unique values.
len(airbnb.neighbourhood_group.unique())
```

Out[12]:
```
5
```

In [13]:
```
#Now Let's check for count for the unique values in neighbourhood column.
len(airbnb.neighbourhood.unique())
```

Out[13]:
```
221
```

In [14]:
```
#Let's check for Value counts for the host_id column
airbnb.host_id.value_counts().head(10)
```

Out[14]:
```
219517861    327
107434423    232
30283594     121
137358866    103
16098958      96
12243051      96
61391963      91
22541573      87
200380610     65
7503643       52
Name: host_id, dtype: int64
```

In [15]:
```
#Now Let's sort values basis on calculated_host_listings_count
airbnb2 = airbnb.sort_values(by="calculated_host_listings_count",ascending=False)
airbnb2.head()
```

Out[15]:

	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_
39773	219517861	Sonder (NYC)	Manhattan	Hell's Kitchen	40.76037	-73.99744	Entire home/apt	185	29	1	
41463	219517861	Sonder (NYC)	Manhattan	Financial District	40.70782	-74.01227	Entire home/apt	396	2	8	
41469	219517861	Sonder (NYC)	Manhattan	Financial District	40.70620	-74.01192	Entire home/apt	498	2	8	
38294	219517861	Sonder (NYC)	Manhattan	Financial District	40.70771	-74.00641	Entire home/apt	229	29	1	
41468	219517861	Sonder (NYC)	Manhattan	Financial District	40.70726	-74.01080	Entire home/apt	229	2	2	

Data Analysis and Visualizations using Tableau:

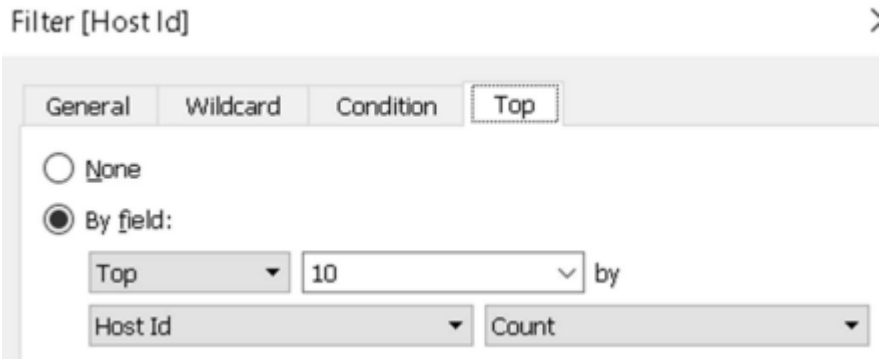
We have used **Tableau** to visualize the data.

Methodology Document PPT 1:

Please find below the steps performed: -

1. Top 10 Host:

We have identified the top 10 Host IDs, Host Names with Count of Host IDs using the Bar Chart.



2. Room type w.r.t. Neighbourhood Group:

- We have created a Pie Chart for understanding the percentage-wise bifurcation of each room type w.r.t. Neighbourhood Group.
- Then added Room Type to the Colours Marks card and count of Host Id to the Size card to highlight the different types of Rooms in different colours.

3. Price of Neighbourhood Groups:

- Created a Bubble Chart with by plotting Neighbourhood Groups in Columns and Prices in Rows.
- Then, we added the Neighbourhood Groups to the colours Marks card to highlight the different neighbourhood Groups in different colours.
- Then, we added the Average Price to Label Marks Card.

4. Price w.r.t. Neighborhood Groups

- For visualizing Price w.r.t. Neighborhood Groups, we used a Box and Whisker plot with Neighbourhood Groups in Columns and Prices in Rows.
- Then, we changed the Price from a Sum Measure to a median measure to get accurate insights.

5. Neighbourhood vs Availability w.r.t. Prices:

- We have created a bar chart using Availability 365 and the price for identifying the top 10 Neighbourhood groups which are sorted by price.

6. **Neighbourhood Popularity:**
    - We have added Neighbourhood in Rows and Sum of reviews in Column.
    - Then, we added neighbourhood groups to the colour marks card.
    - After that, we used a filter to identify neighbours as per the sum of reviews greater than 10000.

7. **Booking w.r.t. Minimum Nights:**
    - First, we created the bin for Minimum Nights using the calculated field.
    - Then, we used these bins to display the Distribution of Minimum Nights based on the count of IDs booked for each Neighbourhood Group.

Methodology Document PPT 2:

1. Room type w.r.t. Neighbourhood Group:

- We have created a Pie Chart for understanding the percentage-wise bifurcation of each room type w.r.t. Neighbourhood Group.
- Then added Room Type to the Colours Marks card and count of Host Id to the Size card to highlight the different types of Rooms in different colours.

2. Neighbourhood vs Availability w.r.t. Prices:

- We have created a bar chart using Availability 365 and the price for identifying the top 10 Neighbourhood groups which are sorted by price.

3. Price Range Analysis:

- We have identified the Customer's Pricing Range Preference based on the volume of bookings done in a price range and the Count of IDs to create a Bar Chart. We have created bins for the Price column with an interval of $20.
- Then, we created Minimum nights bin.
- We used these bins to display the Distribution of Minimum Nights based on the Count of IDs booked for each Neighbourhood Group.

4. Price Variation w.r.t. Geography:

- We used the Maps chart to plot Neighbourhood, and Neighbourhood Groups in the map to visualize the Variation of Prices w.r.t. Geography.

5. Price Variation w.r.t. Room Type and Neighbourhood:

- We created a Table chart by adding Room Type in Rows & Neighbourhood Groups in Columns.
- After that, we have added the Average Price in colour Marks card to highlight the different Room Type in different colours.

6. Bookings w.r.t. Minimum Nights:

- First, we created the bin for Minimum Nights using the calculated field.
- Then, we used these bins to display the Distribution of Minimum Nights based on the count of IDs booked for each Neighbourhood Group.

7. Neighbourhood Popularity:

- We have added Neighbourhood in Rows and Sum of reviews in Column.
- Then, we added neighbourhood groups to the colour marks card.
- After that, we used a filter to identify neighbours as per the sum of reviews greater than 10000.