AIRBNB CASE STUDY: METHODOLOGY DOCUMENT

Presented By: Swapnil Srivastava

Methodology Document PPT 1:

In the case study we have used Jupiter notebook to perform initial analysis of data and Tableau for data analysis and visualisation.

Initial Analysis:

Data Set Used: AB_NYC_2019.csv
Number of Rows: 48895
Number of Columns: 16

Importing libraries and reading the data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [65]:
```
airbnb = pd.read_csv('AB_NYC_2019.csv')
airbnb.head(10)
```

Out[65]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_revi
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	
5	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray Hill	40.74767	-73.97500	Entire home/apt	200	3	
6	5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	Private room	60	45	
7	5178	Large Furnished Room Near B'way	8967	Shunichi	Manhattan	Hell's Kitchen	40.76489	-73.98493	Private room	79	2	
8	5203	Cozy Clean Guest Room - Family Apt	7490	MaryEllen	Manhattan	Upper West Side	40.80178	-73.96723	Private room	79	2	

```
airbnb.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
```

```
airbnb.shape
```

```
(48895, 16)
```

```
airbnb.describe()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listing
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

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.
- Dropped the column last_review as it won't have any significant impact on analysis.
- Checked the formatting in our dataset.
- Identified and review outliers.

```python
airbnb.shape
```

```
(48895, 16)
```

```python
airbnb.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
```

Certain columns that are not efficient to the dataset can be removed

```python
airbnb.drop(['last_review'], axis = 1, inplace = True)
```

```python
# Now reviews per month contains lot of missing values which should be replaced with 0 respectively
airbnb.fillna({'reviews_per_month':0},inplace=True)
```

```python
airbnb.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
reviews_per_month                  0
calculated_host_listings_count     0
availability_365                   0
dtype: int64
```

Verify unique rows

```
airbnb.neighbourhood_group.unique()
```

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

```
airbnb.room_type.unique()
```

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

Creating Features:

categorizing the "availability_365" column into 5 categories

```
def availability_365_categories_function(row):
    """
    Categorizes the "availability_365" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 100:
        return 'Low'
    elif row <= 200 :
        return 'Medium'
    elif (row <= 300):
        return 'High'
    else:
        return 'very High'
```

categorizing the "number_of_reviews" column into 5 categories

```
def number_of_reviews_categories_function(row):
    """
    Categorizes the "number_of_reviews" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 5:
        return 'Low'
    elif row <= 10 :
        return 'Medium'
    elif (row <= 30):
        return 'High'
    else:
        return 'very High'
```

categorizing the "minimum_nights" column into 5 categories

```python
def minimum_nights_function(row):
    """
    Categorizes the "minimum_nights" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 3:
        return 'Low'
    elif row <= 5 :
        return 'Medium'
    elif (row <= 7):
        return 'High'
    else:
        return 'very High'
```

categorizing the "price" column into 5 categories

```python
def price_categories_function(row):
    """
    Categorizes the "price" column into 5 categories
    """
    if row <= 50:
        return 'very Low'
    elif row <= 125:
        return 'Low'
    elif row <= 250 :
        return 'Medium'
    elif (row <= 500):
        return 'High'
    else:
        return 'very High'
```

Categorise columns as categorical and numeric:

4.1 Categorical

```
inp0.columns
```

```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'availability_365_categories',
       'minimum_night_categories', 'number_of_reviews_categories',
       'price_categories'],
      dtype='object')
```

```
# Categorical nominal
categorical_columns = inp0.columns[[0,1,3,4,5,8,16,17,18,19]]
categorical_columns
```

```
Index(['id', 'name', 'host_name', 'neighbourhood_group', 'neighbourhood',
       'room_type', 'availability_365_categories', 'minimum_night_categories',
       'number_of_reviews_categories', 'price_categories'],
      dtype='object')
```

4.2 Numerical

```
numerical_columns = inp0.columns[[9,10,11,13,14,15]]
numerical_columns
```

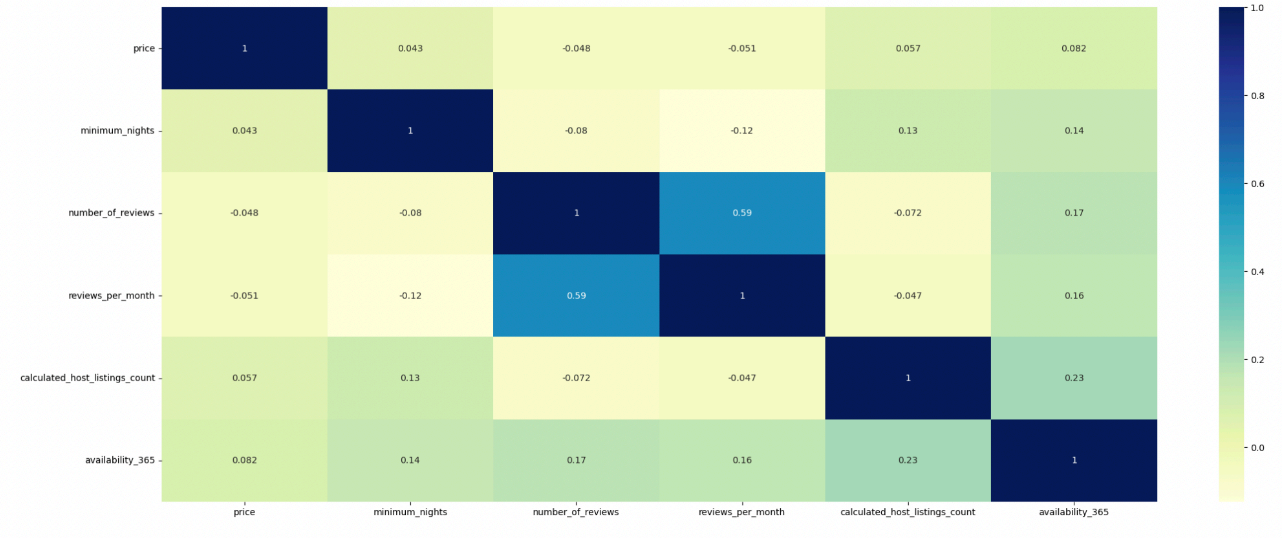
```
Index(['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

```
inp0[numerical_columns].describe()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	48895.000000	48895.000000	48895.000000	38843.000000	48895.000000	48895.000000
mean	152.720687	7.029962	23.274466	1.373221	7.143982	112.781327
std	240.154170	20.510550	44.550582	1.680442	32.952519	131.622289
min	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	69.000000	1.000000	1.000000	0.190000	1.000000	0.000000
50%	106.000000	3.000000	5.000000	0.720000	1.000000	45.000000
75%	175.000000	5.000000	24.000000	2.020000	2.000000	227.000000
max	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

Created heatmap in Jupyter notebook for understanding correlation between numerical variables

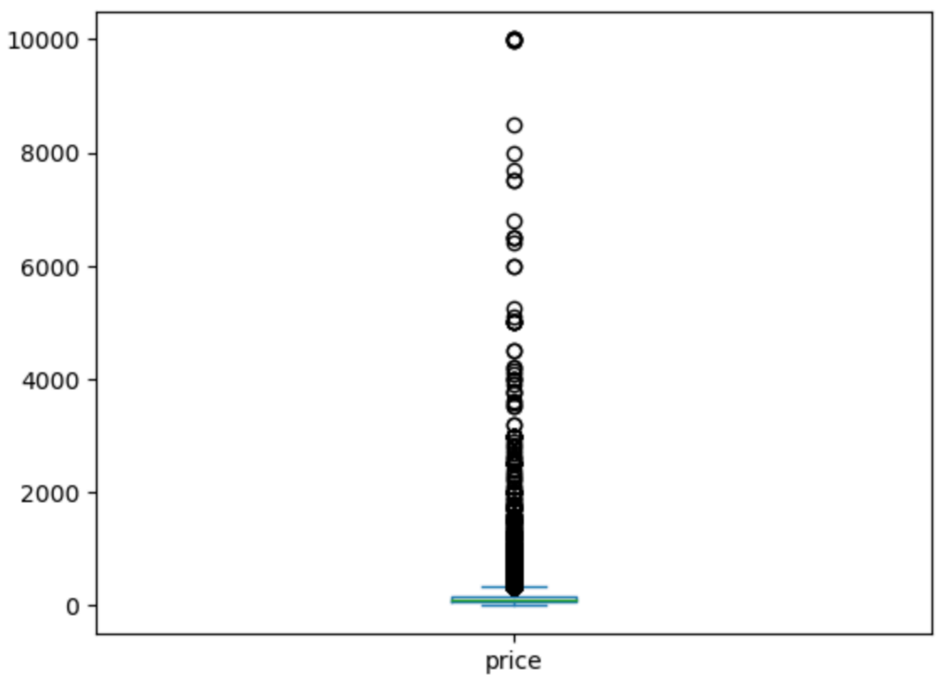
```
cor=airbnb[['price','minimum_nights','number_of_reviews','reviews_per_month','calculated_host_listings_count','avail

plt.figure(figsize=(25,10))
sns.heatmap(cor, cmap="YlGnBu", annot = True)
plt.show()
```



Created a boxplot for Price column to determine outliers. We see majority of Listings have Price below 5000

In [90]:
```
airbnb.price.plot.box()
```

Out[90]: <AxesSubplot:>



Created some visualisations using Tableau:

• Neighbourhood groups participation chart

Identified neighbourhood groups with most listings using a pie chart. We used the count of Ids to determine the size and angle of the chart and we differentiated neighbourhood groups by colours.

• Price Variation in different neighbourhoods

Used map feature in tableau using latitude and longitude columns and identified different price variations across all neighbourhoods. Based on the size density of the prices we were able to identify Manhattan and Brooklyn as neighbourhoods with most Airbnb prices.

• Popular neighbourhoods

We used number of reviews in each neighbourhood to create a bar graph of 5 most popular neighbourhoods in every borough of NYC.

We create a rank_measure using RANK(SUM([Number Of Reviews])) to get ranks of neighbourhoods based on number_of_reviews. We filtered the neighbourhoods using rank_measure condition from filters pane with Range between 1 to 5 for the Neighbourhoods. The higher number of customer reviews imply higher satisfaction in these neighbourhoods.

Methodology Document PPT 2:

• Average Airbnb Prices

We created a bar graph to depict average prices of Airbnb across neighbourhoods, calculated by averaging the prices across each neighbourhood groups.

• Effect Of Minimum Nights On Customer Reviews

Created a tree map using average(minimum nights) as the size of tree map and review category ranging from very high to very low. We see that customers are more likely to leave reviews for lower number of minimum nights.

• Top 10 Hosts

We identified the top 10 Host Names using calculated number of host listings and visualised through the bubble chart.

- **Price Vs Availability In Different Neighbourhoods**

We were able to represent price vs availability of Airbnbs by creating a dual axis bar graph to denote availability_365 and trend line to denote Prices.

- **Room Type Preferences**

Created a bar graph by taking room type on X-axis and count of room type on Y-axis. Shared rooms only account for 2% of the total types of rooms.

- **Tools Used:**

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