DATA METHODOLOGY

In this Airbnb case study, we have used Jupyter Notebook as tool with which we performed extensive EDA, Data analysis and Visualization.

Dataset: AB_NYC_2019.csv

Imported below python library for executing the task:

Imported Library:

```
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib.pyplot import figure
import matplotlib.pyplot as plt
```

Imported dataset:

```
df = pd.read_csv('AB_NYC_2019.csv')
```

Number of Rows: 48895

Number of Columns: 16

1. EDA: This stage includes understanding the rows, column, data type, null values, outliers and manipulating the dataset.

```
df.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)
```

- Two columns last_review , reviews_per_month has around 20.56% missing values.

- The "last_review" column represents latest review received from customer, manipulated this column from object to datetime.

```
df['last_review'] = pd.to_datetime(df['last_review'], errors='coerce', format='%d-%m-%Y')
```

- The reviews_per_month column is float data type.

  Converting the float to integer and replace empty cells with 0.

```
df['reviews_per_month'] = (df['reviews_per_month'].fillna(0) * 100).astype('int32')
```

- “name” and “host_name” column found missing value which were replaced as “Note specified”.

```
df['name'] = df['name'].fillna('No_Specified')
```

```
df.host_name = df.host_name.fillna('Not_Specified')
```

- The avarege minimum_nights=7 days and max=1250 days. The max minimum days in a year is 365 day. This could be outliers hence imputed minimum_nights above 365 days with 365 days.

```
#Replace minimum nights.
df.loc[df['minimum_nights'] > 365, 'minimum_nights'] = 365
```

- Identified Outliers in numerical columns. These outliers may be true and may impact the statistical calculation.

We are Analysing the data set, and each column is important feature hence we are not dropping any rows and columns.

2. Adding features:

Categorised the "price" column into 7 categories.

```
def price_category_function(row):

    if row <= 30:
        return 'very Low'
    elif row <=50:
        return 'Low'
    elif row <= 70:
        return 'Medium'
    elif (row <= 110):
        return 'High'
    elif (row<=150):
        return 'very High'
    elif (row<=200):
        return 'extreme'
    else:
        return 'Very extreme'
```

Categorised the "minimum_nights" column into 5 categories.

```
def minimum_night_categories_function(row):

    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'
```

Categorised the number_of_reviews column in 5 categories.

```
def number_of_reviews_categories_function(row):
    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'
```

Categorizes the "availability_365" column into 5 categories

```
def availability_365_categories_function(row):
    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'
```

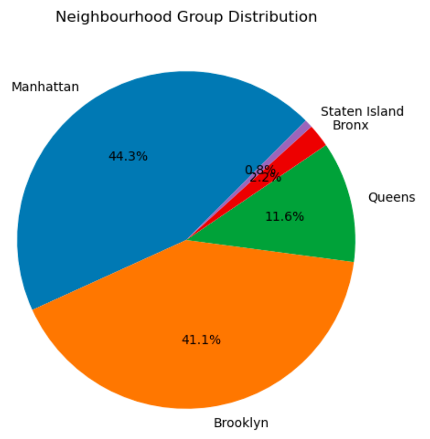
3. Data Analysis and Visualization:

3.1 Univariate Analysis and Visualization:

- Analysed columns “name”, “host_id”, “host_name” that has highest number of unique counts.
- Analysed the column “neighbourhood group” distribution. Manhattan and Brooklyn contributes highest distribution.

```
plt.figure(figsize=(6, 6))
plt.pie(x=df.neighbourhood_group.value_counts(normalize=True) * 100,
        labels=df.neighbourhood_group.value_counts(normalize=True).index,
        autopct='%1.1f%%',
        startangle=45)

plt.title('Neighbourhood Group Distribution')
plt.show()
```

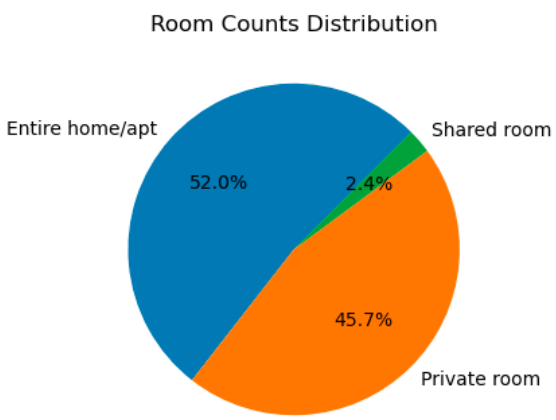


- Analysed the “neighbourhood” for highest number of unique counts.

```
df.neighbourhood.value_counts()
```

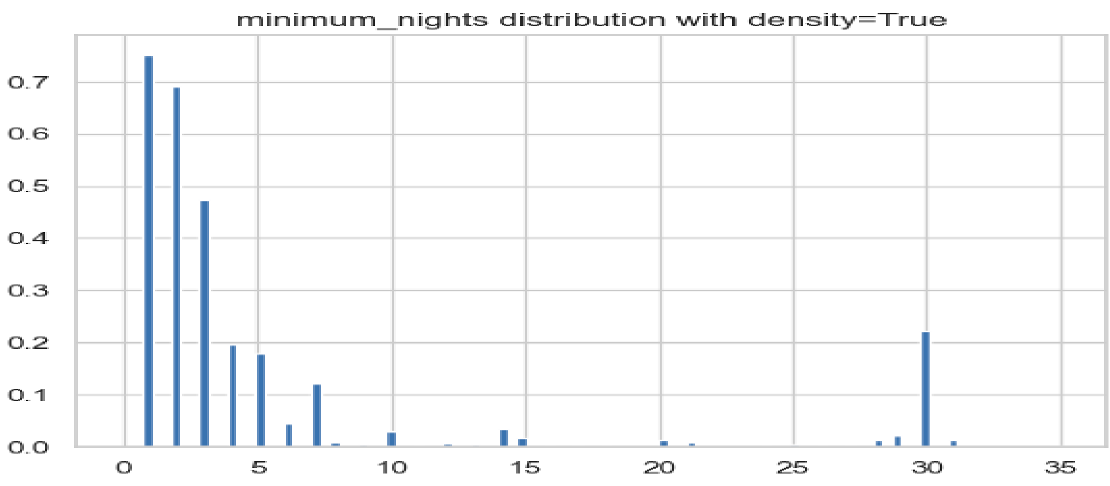
```
neighbourhood
Williamsburg          3920
Bedford-Stuyvesant    3714
Harlem                2658
Bushwick              2465
Upper West Side       1971
                      ...
Fort Wadsworth           1
Richmondtown             1
New Dorp                 1
Rossville                1
Willowbrook              1
```

- Analysed column “room_type” distribution. Entire home/apt and Private room contributes 98% of distribution.



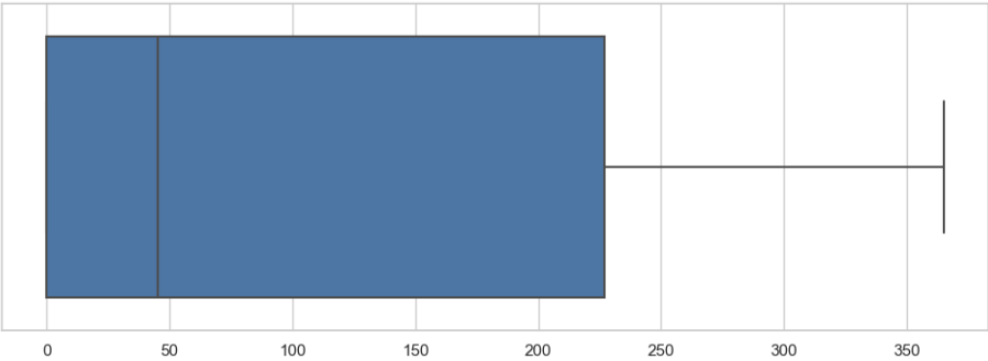
- Analysed column “minimum_nights” distribution. 0-2 days night counts highest distribution.

```
plt.hist(data = df, x = 'minimum_nights',bins=100,range=(0,35),density=True)
plt.title("minimum_nights distribution with density=True")
plt.show()
```

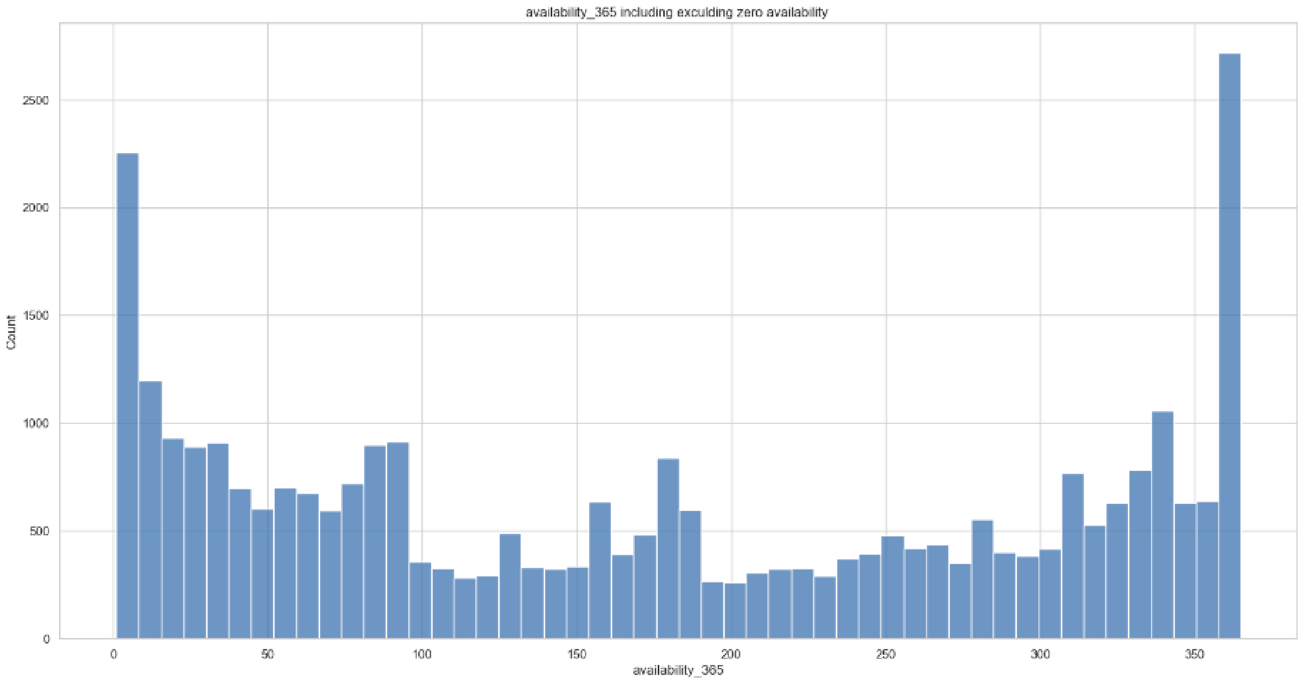


- Analysed column “availability_365”.

```
plt.figure(figsize = (12,4))
sns.boxplot(data = df , x = 'availability_365')
plt.show()
```




```
plt.figure(figsize = (20,10))
sns.histplot(data = df, x = 'availability_365',bins=50,binrange=(1,365))
plt.title("availability_365 including exculding zero availability ")
plt.show()
```

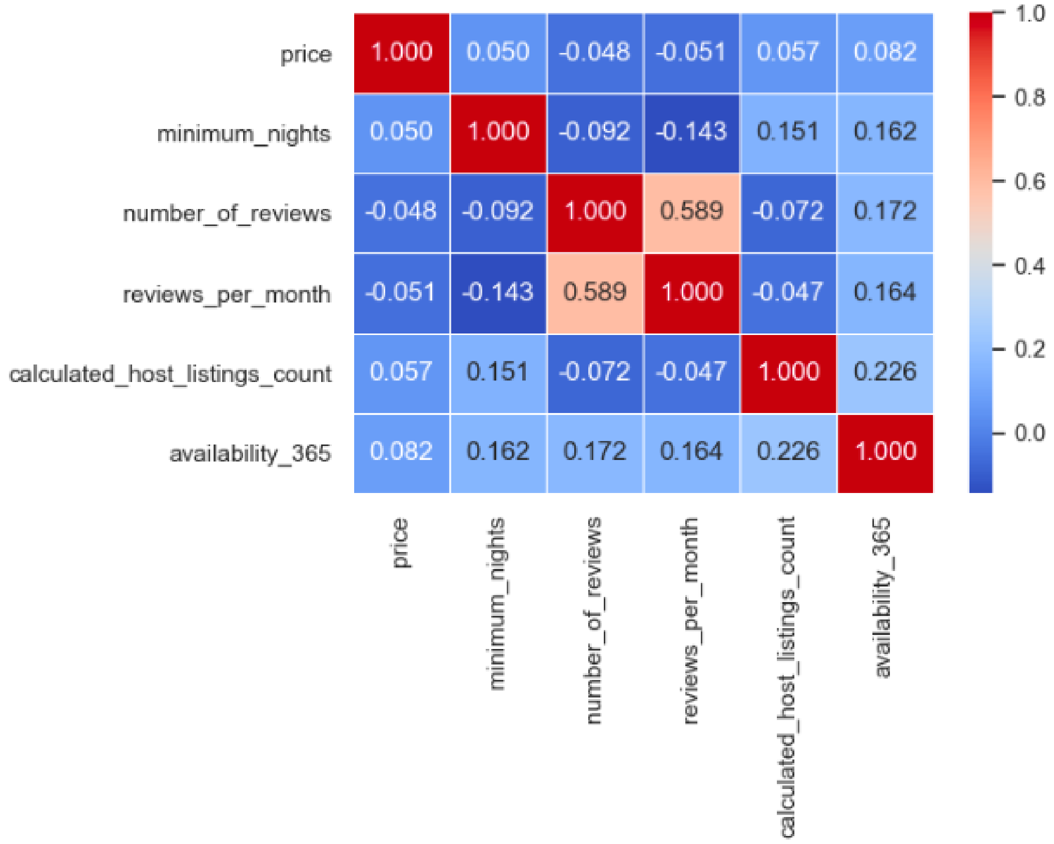


3.1 Bivariate Analysis:

- Numerical columns correlation:

```
numerical_columns = df[['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365']]
numerical_columns.head()
```

```
plt.figure(figsize=(6,4))
sns.heatmap(data=numerical_columns.corr(), annot=True, cmap='coolwarm', fmt='.3f', linewidths=.5)
plt.show()
```



- The

number_of_reviews vs reviews_per_month show high correlation 58%

- Analysed number_of_reviews_category vs prices.

```
group0 = df.groupby('number_of_reviews_category')['price'].sum/mean/median().sort_values(ascending = False)
group0
```

SUM

```
number_of_reviews_category
Low          4002323
very Low     1806531
High          971346
Medium        508647
very High     178431
Name: price, dtype: int64
```

Mean

```
number_of_reviews_category
very High     238.863454
High          164.830477
Low           153.746274
Medium        145.203254
very Low      142.022877
Name: price, dtype: float64
```

Median

```
: number_of_reviews_category
  very High     238.863454
  High          164.830477
  Low           153.746274
  Medium        145.203254
  very Low      142.022877
  Name: price, dtype: float64
```

- Analysed neighbourhood vs prices to find the highest revenue contributing neighbourhood.

```
group3 = df.groupby('neighbourhood')['price'].sum().sort_values(ascending = False).head(5)
group3
```

```
neighbourhood
Williamsburg              563707
Midtown                   436801
Upper West Side           415720
Hell's Kitchen            400987
Bedford-Stuyvesant        399917
East Village              344812
```

- Analysed neighbourhood_group vs price to find the highest revenue contributing neighbourhood_group.

- room_type vs number_of_reviews_categories

```
pd.crosstab(df['room_type'], df['number_of_reviews_category'])
```

number_of_reviews_category room_type	High	Low	Medium	very High	very Low
Entire home/apt	3809	14909	1960	504	4227
Private room	1950	10769	1494	226	7887
Shared room	134	354	49	17	606

- 'room_type' vs 'price_categories'

```
pd.crosstab(df['room_type'], df['price_category'])
```

price_category room_type	High	Low	Medium	Very extreme	extreme	very High	very Low
Entire home/apt	4669	163	661	7637	5815	6437	27
Private room	7053	5327	6131	707	711	2016	381
Shared room	196	412	188	40	28	45	251

3.1 Multi-Variate Analysis:

- We performed multivariate analysis to check mean of “reviews_per_month” vs “availability_365_categories” and “price_category”.