# Airbnb Case Study Methodology

Atulya Pattamatta
Vyshnav E
Hemant Kokane                                                    (DS C44)

## 1. What is Airbnb?

- Airbnb is an online community marketplace, started in 2008, which allows property owners to rent out their properties.
- These spaces could include private rooms, entire home/apartment or a shared room.
- Major reason for the success of Airbnb could be the fact that most people prefer cheaper, pocket friendly stays.
- Over a 100 million people have rented properties.

## 2. Business Understanding

- With property listings in 190+ countries, guests could find a place to rent on the basis of where they want to go, when they want to check-in and how many guests would be staying.
- As for the hosts, it is free to post their properties on Airbnb with all the details like location, amenities etc, which helps Airbnb match with the perfect guests.
- This gives a win-win for property owners as well as guests, looking for a not-so-expensive accommodation while they enjoy their vacations.
- Airbnb also has an online community where guests and hosts can review each other, which helps improve services and overall customer experience.

## 3. Problem Statement

For the past few months, Airbnb has seen a major decline in revenue. Now that the Restrictions have started lifting and people have started to travel more, Airbnbwants to make sure that it is fully prepared for this change. The different leadersatAirbnb want to understand some important insights based on various attributes in the dataset so as to increase the revenue

## 4. Reading the Dataset

```
In [2]: airbnb_data = pd.read_csv('AB_NYC_2019.csv')
        airbnb_data.head()
```

Out[2]:

| | 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 | |

Here, we have a dataset of Airbnb listings of New York, with around 49,000 rows and 16 columns. It is absolutely essential to analyze the data using appropriate parameters/variables like reviews, location, price, services etc., in order to get insights on the next best steps that Airbnb needs to take as a business.

```
In [3]: # Checking rows and columns
        s = airbnb_data.shape
        print('Rows:',s[0])
        print('Columns:',s[1])
```

```
Rows: 48895
Columns: 16
```

## 5. Cleaning the Data

- Checking data types

```
In [4]:  # Checking datatypes
         airbnb_data.dtypes

Out[4]:  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
```

The column "last_review" is a date but the dtype is "object". Hence converting into "datetime"

```
In [5]:  airbnb_data['last_review'] = pd.to_datetime(airbnb_data['last_review'], format='%d-%m-%Y')

In [6]:  airbnb_data.dtypes

Out[6]:  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                  datetime64[ns]
         reviews_per_month                   float64
         calculated_host_listings_count        int64
         availability_365                      int64
         dtype: object
```

- Checking for Null values

```
In [7]: airbnb_data.isnull().sum()

Out[7]: 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
```

Here, we will be treating certain columns.
- **"name"** and **"host_name"** - Since most of our analysis would be based on the major demographics like location, price, etc., we do not need actual names of the hosts and the properties. However, instead of removing the column, we will impute the null values in the "name" column with $ and in "host_name" with #.
- **"reviews _per_month"** - Here, since null values depict that there were no reviews for the property, we can safely replace null values with "0.0"
- **"Last_review"** - This column has more than 10,000 null values and is not important for our analysis. Hence we can drop the column.

After treating the columns:

```
In [8]:  # Treating the columns

         airbnb_data['name'] = airbnb_data['name'].fillna("$")
         airbnb_data['host_name'] = airbnb_data['host_name'].fillna("#")
         airbnb_data['reviews_per_month'] = airbnb_data['reviews_per_month'].fillna(0.0)
         airbnb_data = airbnb_data.drop('last_review', axis=1)
```

```
In [9]:  airbnb_data.isnull().sum()

Out[9]:  id                               0
         name                             0
         host_id                          0
         host_name                        0
         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
```

## 6. Using Cleaned Data for Visualization

File   Data   Server   Window   Help

⊖· Airbnb_NYC

**Connections**                          Add

Airbnb_NYC
Text file

Airbnb_NYC.csv

**Files**

☐ Use Data Interpreter

Data Interpreter might be able to
clean your Text file workbook.

⊞ Airbnb_NYC.csv