# Course Title:

# Data Science (数据科学)

**(Semester: Fall 2021)**

**Dr. Oluwarotimi W. SAMUEL**

**Research Center for Neural Engineering**
**Shenzhen Institutes of Advanced Technology**
**Chinese Academy of Sciences**

**Contact:** (Email: samuel@siat.ac.cn & timitex92@gmail.com)

Phone: +86-15814491870

**(2021.11.04)**

# Exploratory Data Analysis (EDA)

## ❑ Outline for today's lecture

- ✓ Explorative Data Analysis
- ✓ Case Study I: Using Online Dataset
- ✓ Case Study II: Using Online Dataset

# Exploratory Data Analysis (EDA)

- **Objective:** This lecture will focus on Exploratory Data Analysis with emphases on Case studies.

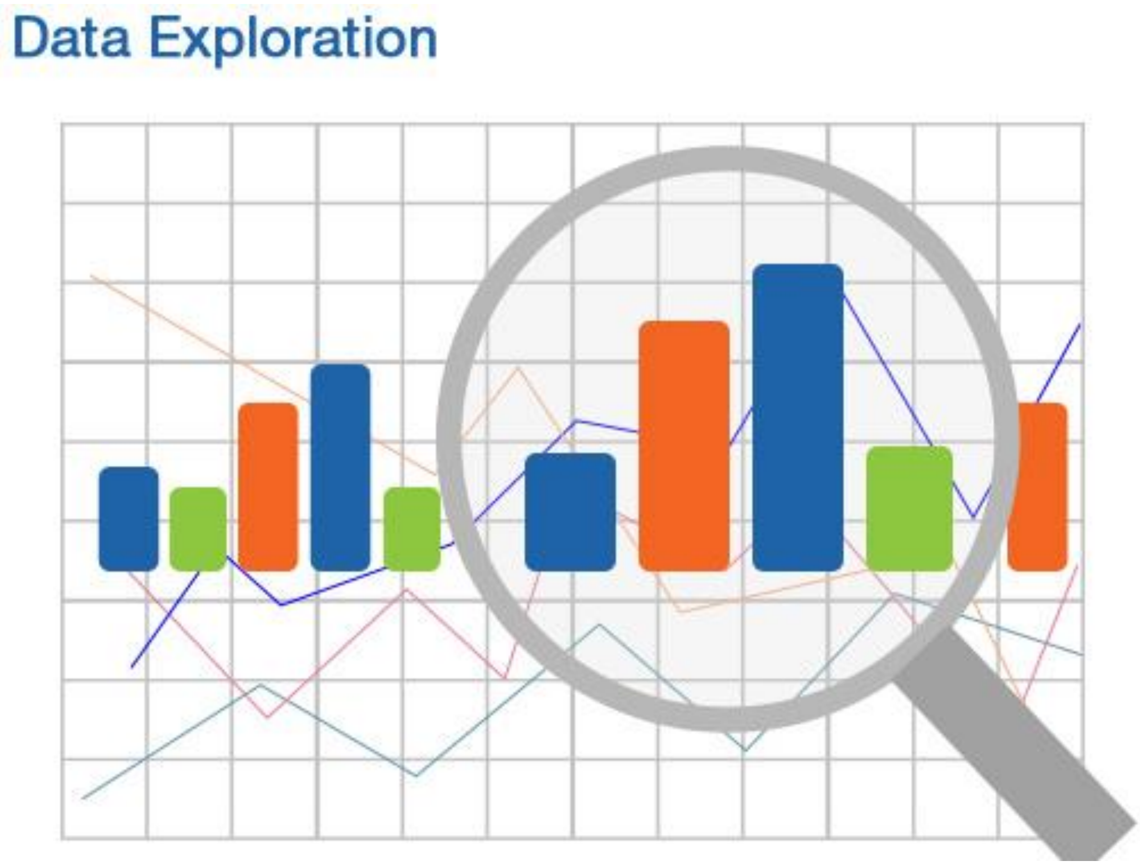
- **Expectation:** At the end of this lecture, students are expected to understand the procedure for Exploring and Analyzing data when carrying out a Data Science projects.

## ❑ Data Exploration:

- ✓ Data exploration and analysis is an approach similar to whereby visual tools are employed to understand the characteristics of the data.

- ✓ Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics.

## ❑ Data Exploration:



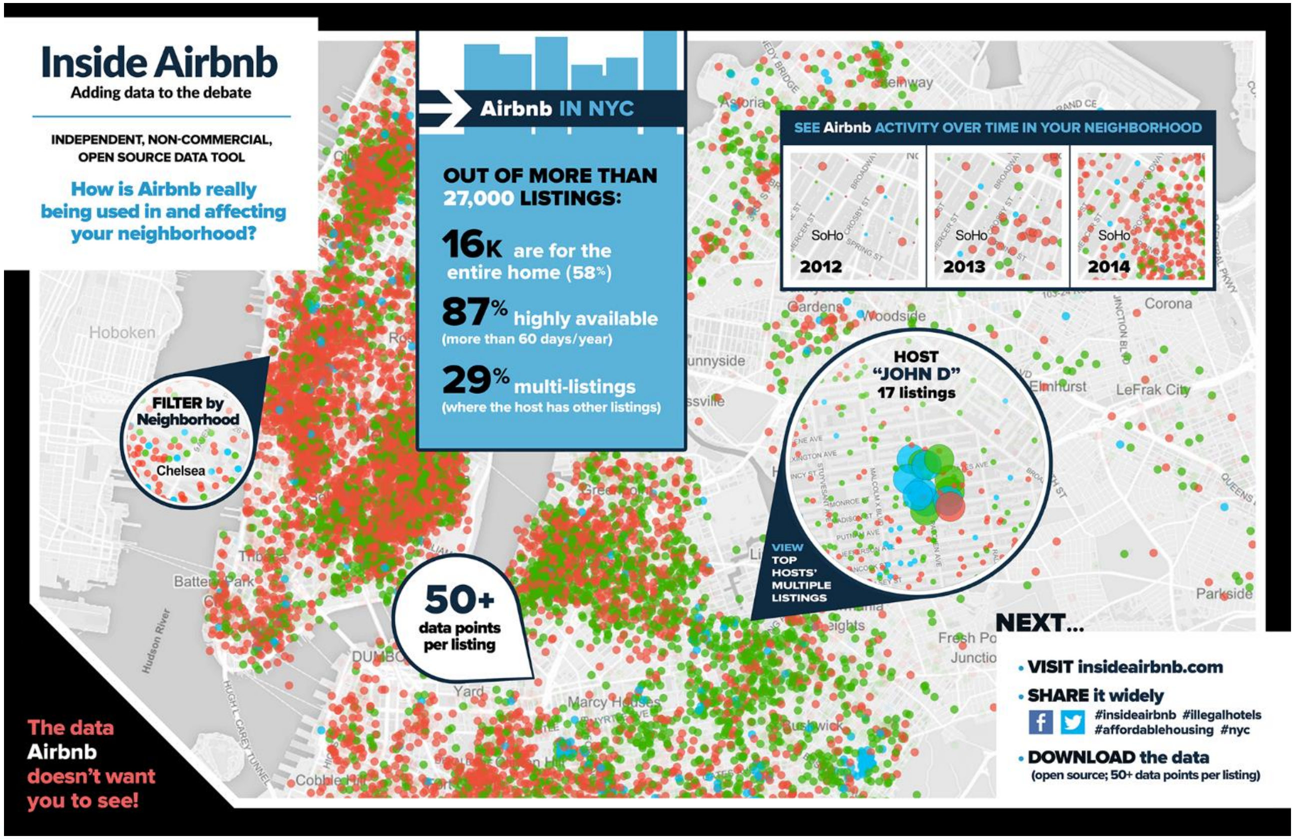
## ❑ Importance of Data Exploration:

- ✓ Helps to determine the cleaning processes that should be applied
- ✓ Helps us to determine the right tool for analysis
- ✓ **May provide Data Scientists with pre-knowledge of inherent trend/s in the dataset**
- ✓ **Helps us to select the appropriate machine learning model/s**

# ❑ Case Study:

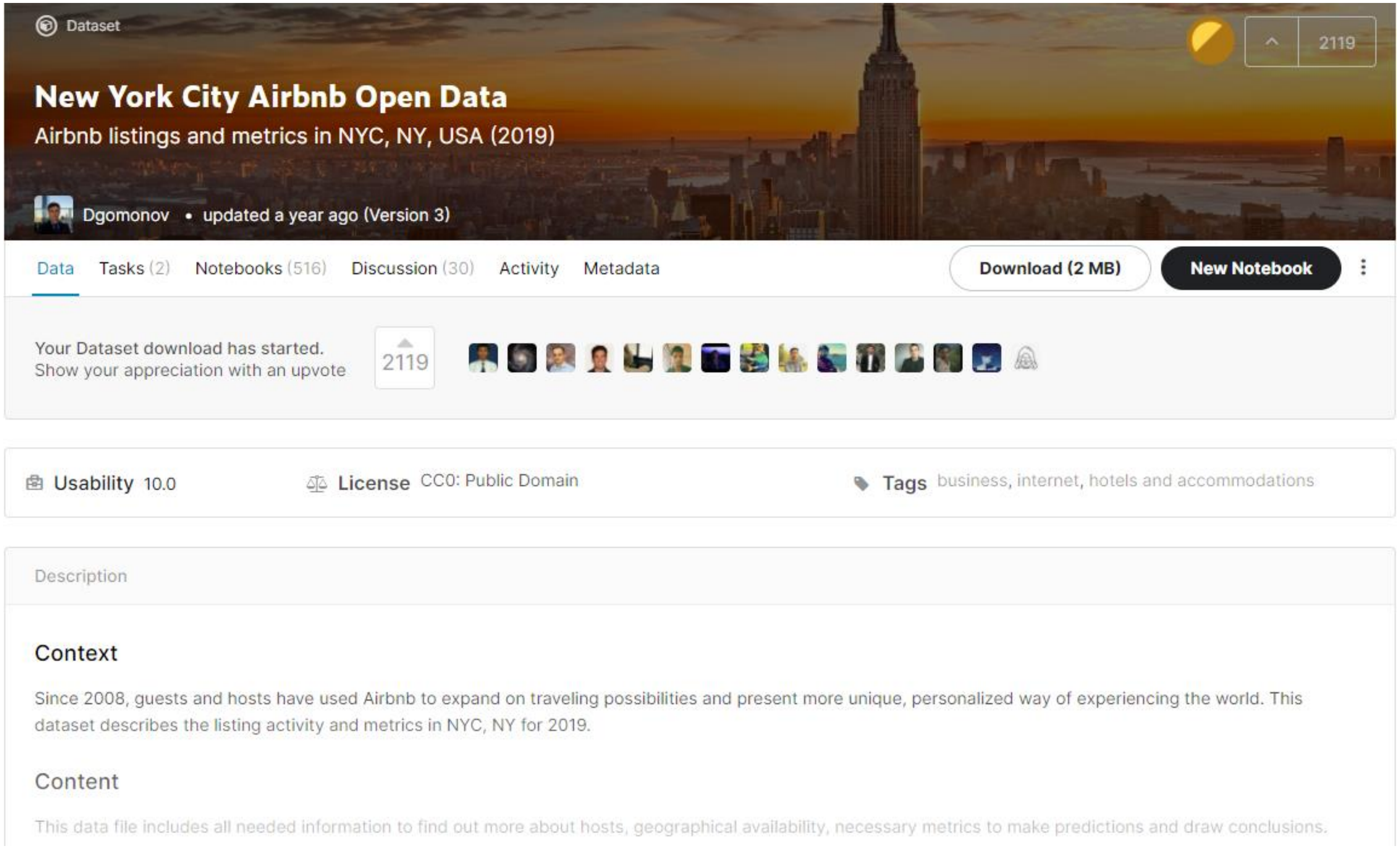
**Airbnb--**Provide data that quantifies the impact of short-term rentals on housing and residential communities; and also provides a platform to support advocacy for policies to protect our cities from the impacts of short-term rentals.

http://insideairbnb.com/about.html



# ❑ Exploring Data Visualization (Case Study)

## Dataset




https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

# Data Exploration

## ❑ Case Study:

First, relevant libraries are imported (numpy, pandas, matplolib, seaborn, etc.) to work with the Airbnb data.

```
###... Import all relevant libraries...###
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sn
```

## ❑ Case Study:

Next, we read the csv file that contained the data using read_csv function provided by the Pandas library. And then display the first 5 records in the Dataframe(df)

```
###... Loading the Dataset ...###
# Read the CSV file that contains the dataset using read_csv function
# Display the first 5 records in the pandas dataframe (df)
df = pd.read_csv('Airbnb_NYC_2019.csv', low_memory=False)
df.head()  df: {DataFrame: (48895, 16)}
```

# Data Exploration

## ❑ Case Study:

Displaying the first 5 records in the data frame:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude |
|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 |
| 3 | 3831 | Cozy Entire Floor of Brownstone | 4869 | LisaRoxanne | Brooklyn | Clinton Hill | 40.68514 |
| 4 | 5022 | Entire Apt: Spacious Studio/Loft by central park | 7192 | Laura | Manhattan | East Harlem | 40.79851 |

# Data Exploration

## ❑ Case Study:

Next, we tried to check the data types of all the columns in the data frame:

```
###...Exploring the Dataset...###
# Checking data types of all columns
df.dtypes   df: {DataFrame: (48895, 16)}
```

# Data Exploration

## ❑ Case Study:

data types in df:

```
10  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
```

# Data Exploration

## ❑ Case Study:

Next, we tried describing some basic statistics of the data in each column of the data frame using the "describe ()" method.

```
# Describing some basic statistics of the data in each column of the data frame
df.describe()  df: {DataFrame: (48895, 16)}
```

# Data Exploration

## ❑ Case Study:

Basic Statistics

```
# Describing some basic statistics of the data in each column of the data frame
df.describe()  df: {DataFrame: (48895, 16)}
```

| | id | host_id | latitude | longitude | price | minimum_nights | number_of_reviews | reviews_p |
|---|---|---|---|---|---|---|---|---|
| **count** | 4.889500e+04 | 4.889500e+04 | 48895.000000 | 48895.000000 | 48895.000000 | 48895.000000 | 48895.000000 | 38843.000 |
| **mean** | 1.901714e+07 | 6.762001e+07 | 40.728949 | -73.952170 | 152.720687 | 7.029962 | 23.274466 | 1.373221 |
| **std** | 1.098311e+07 | 7.861097e+07 | 0.054530 | 0.046157 | 240.154170 | 20.510550 | 44.550582 | 1.680442 |
| **min** | 2.539000e+03 | 2.438000e+03 | 40.499790 | -74.244420 | 0.000000 | 1.000000 | 0.000000 | 0.010000 |
| **25%** | 9.471945e+06 | 7.822033e+06 | 40.690100 | -73.983070 | 69.000000 | 1.000000 | 1.000000 | 0.190000 |
| **50%** | 1.967728e+07 | 3.079382e+07 | 40.723070 | -73.955680 | 106.000000 | 3.000000 | 5.000000 | 0.720000 |
| **75%** | 2.915218e+07 | 1.074344e+08 | 40.763115 | -73.936275 | 175.000000 | 5.000000 | 24.000000 | 2.020000 |
| **max** | 3.648724e+07 | 2.743213e+08 | 40.913060 | -73.712990 | 10000.000000 | 1250.000000 | 629.000000 | 58.500000 |

# Data Exploration

## ❑ Case Study:

Next, Checking for null values.

```
# Checking for null values
df.isnull().sum()   df: {DataFrame: (48895, 16)}
```

```
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
```

# Data Exploration

## ❑ Case Study:

Next, we need to drop some columns:

After a quick analysis, I deiced to drop less effective variables such as the:

✓ last_review, id, host_name, since some values are missing in these columns.

✓ I have also filled the NaN values of reviews_per_month with "zero" (0) and name by NoName.

# Data Exploration

## ❑ Case Study:

```
# After careful analysis, we have to drop some less effective columns (attributes)
# last_review,id, host_name
df.drop(['id','host_name','last_review'], axis = 1,inplace=True)  df: {DataFrame: (48895, 13)}
df.shape  df: {DataFrame: (48895, 13)}
```

```
# Also, I have filled the NaN values of reviews_per_month with zero and name by NoName
df.reviews_per_month.fillna(value=0,inplace=True)  df: {DataFrame: (48895, 13)}
df.name.fillna("NoName", inplace=True)  df: {DataFrame: (48895, 13)}
```

# Data Exploration

## ❑ Case Study:

Next, we then check again to see if 'null' values still exist to be sure we are on the right path…

```
# Check to be sure if 'null' values still exist
df.isnull().sum()  df: {DataFrame: (48895, 13)}
```

```
name                              0
host_id                           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
```

## ❑ Case Study:

Graphical explorations:

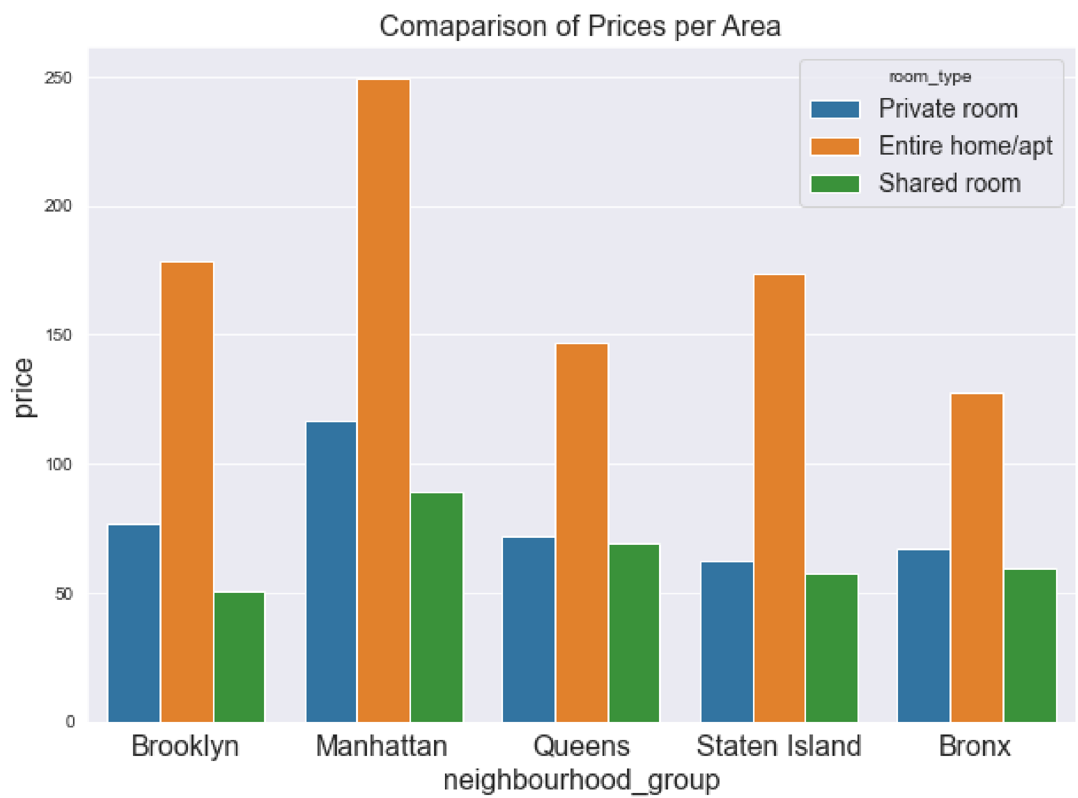
Exploring the prices of apartments/accommodation in the 5 neighborhood areas of NYC

```
# Graphical Exploration
#plt.figure(figsize=(16, 6))
sns.set_style('darkgrid')
plt.figure(figsize=(10,7))
plt.rc('xtick',labelsize=16)
plt.rc('axes',labelsize=16)
plt.rc('axes',titlesize=16)
plt.rc('legend',fontsize=14)
plt.title('Comaparison of Prices per Area')
sns.barplot(df.neighbourhood_group,df.price,hue=df.room_type,ci=None)  df: {DataFrame: (48895, 13)}
```

# Data Exploration

## ❑ Case Study:

Graphical explorations:



# Data Exploration

- **Case Study:**

**The above bar plot concludes that:**

- ✓ Manhattan is the most expensive region in neighborhood group
- ✓ The price of entire home/apt is more than any other room type.
- ✓ Bronx is the cheapest among neighborhood groups.

# Data Exploration

## ❑ Case Study:

Graphical explorations:

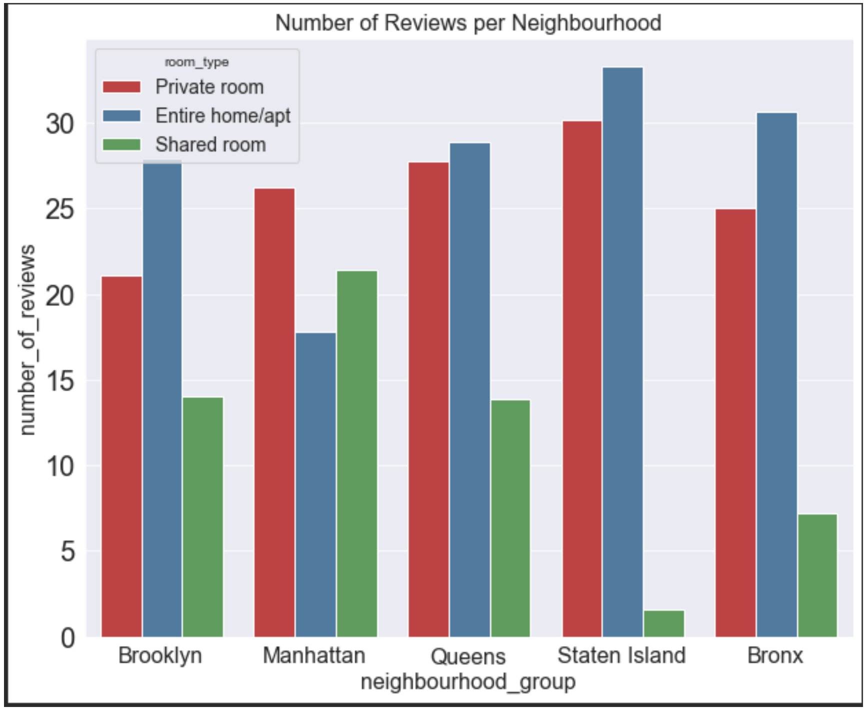
Analyze the Number of Reviews given per Neighbourhood.

```
# Number of Review per neighbourhood
sns.set_style('darkgrid')
plt.figure(figsize=(10,8))
plt.title('Number of Reviews per Neighbourhood')
plt.rc('axes',titlesize=14)
sns.barplot(data=df,x='neighbourhood_group', y='number_of_reviews',hue='room_type',ci=None,palette='Set1', saturation=0.6)
```

# Data Exploration

## ❑ Case Study:

Analyze the Number of Reviews given per Neighbourhood.



# Data Exploration

## ❑ Case Study:

Graphical explorations:

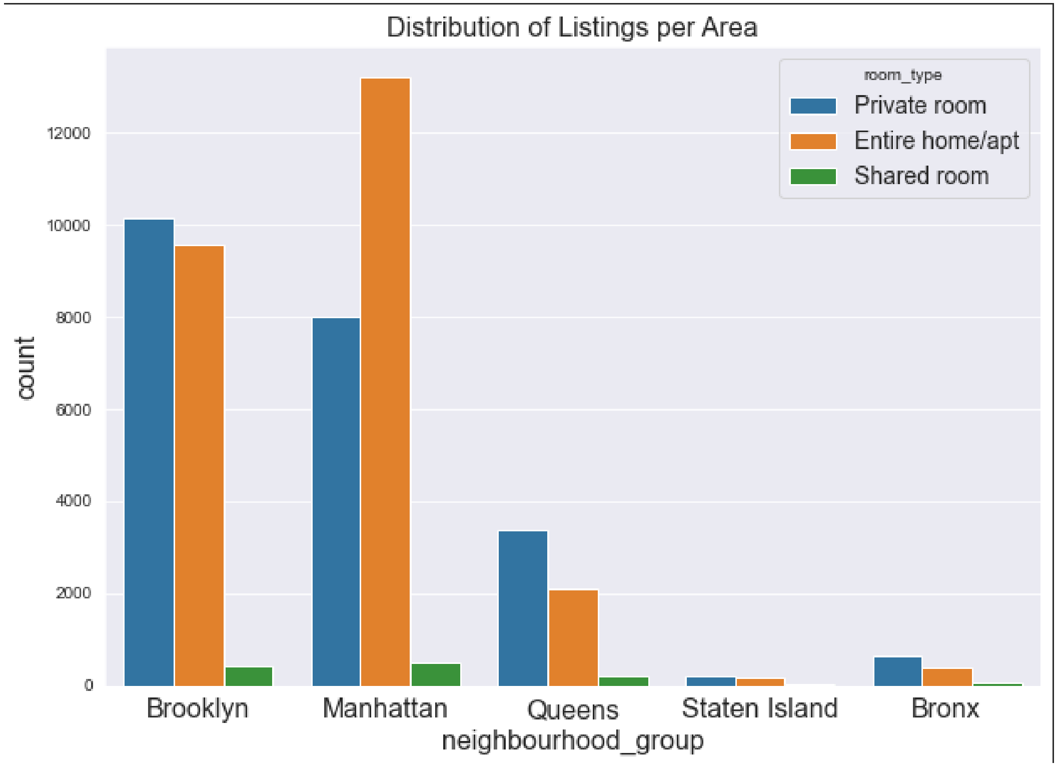
Exploring the distribution of listings across the neighborhood in NYC.

```python
# Determining the distribution of listings across the neighbourhoods
# plt.figure(figsize=(16, 6))
sns.set_style('darkgrid')
plt.figure(figsize=(10,7))
plt.rc('xtick',labelsize=16)
plt.rc('axes',labelsize=16)
plt.rc('axes',titlesize=16)
plt.rc('legend',fontsize=14)
plt.title('Distribution of Listings per Area')
sns.countplot(df.neighbourhood_group,hue=df.room_type)  df: {DataFrame:
```

# Data Exploration

## ❑ Case Study:

Graphical explorations: Exploring the distribution of listings across NYC



# Data Exploration

## ❑ Case Study:

**The above count plot concludes:**

- ✓ Staten Island and Bronx have the least number of entries in the listings.
- ✓ Shared rooms are less available in the listings.
- ✓ Manhattan and Brooklyn neighborhoods have far more entries in the listings.

# Data Exploration

## ❑ Case Study:

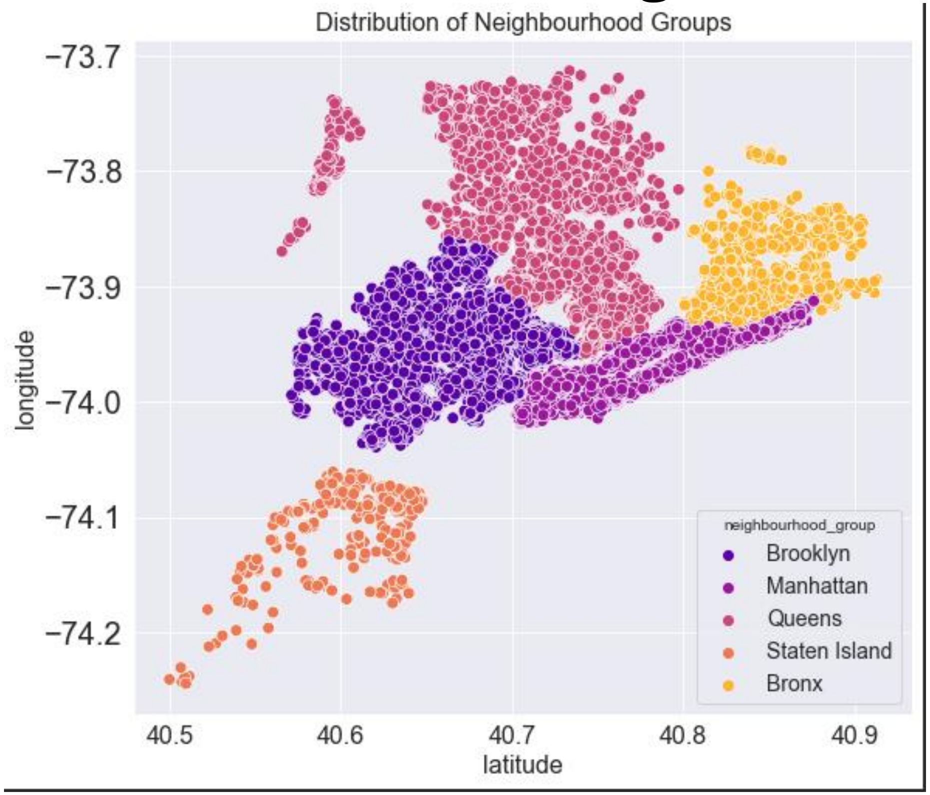
Graphical explorations: Distribution of Neighbouhood by groups…

```
# Exploring the Distribution of neighbourhood by groups
plt.figure(figsize=(9,8))
plt.title('Distribution of Neighbourhood Groups')
plt.rc('axes',titlesize=14)# Here I have compared the prices of all Neighbouhoods with
sns.scatterplot(data=df,x='latitude',y='longitude',# sns.set_style('darkgrid')
                hue='neighbourhood_group',# plt.figure(figsize=(10,7))
                palette='plasma',# plt.rc('xtick',labelsize=16)
                s=60)# plt.rc('axes',labelsize=16)
```

# Data Exploration

## ❑ Case Study:

Graphical explorations: Distribution of neighborhood by groups…



# Data Exploration

## ❑ Case Study:

Graphical explorations: Analyzing accommodation availability…

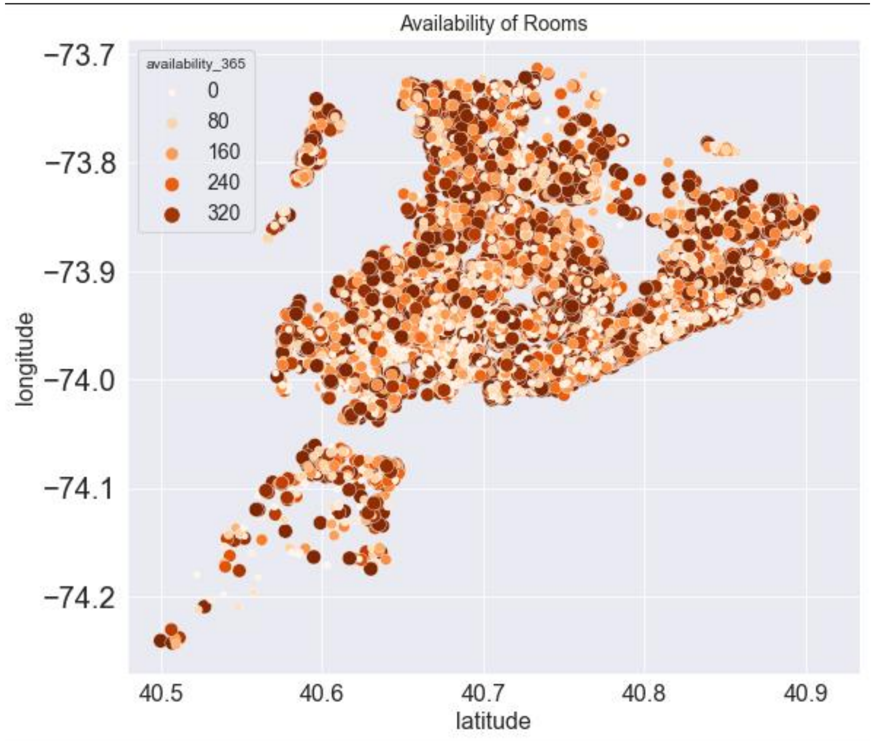
```
# Analyzing the accomodation availability
plt.figure(figsize=(9,8))
plt.title('Availability of Rooms')
plt.rc('axes',titlesize=14)
sns.scatterplot(data=df,x='latitude',y='longitude',
               size='availability_365',
               hue='availability_365',
               sizes=(20,100),
               palette='Oranges',
               s=70)
```

# Data Exploration

## ❑ Case Study:

Graphical explorations: Analyzing accommodation availability…

**Here I have shown which rooms have the most availability**



# Data Exploration

## ❑ Case Study:

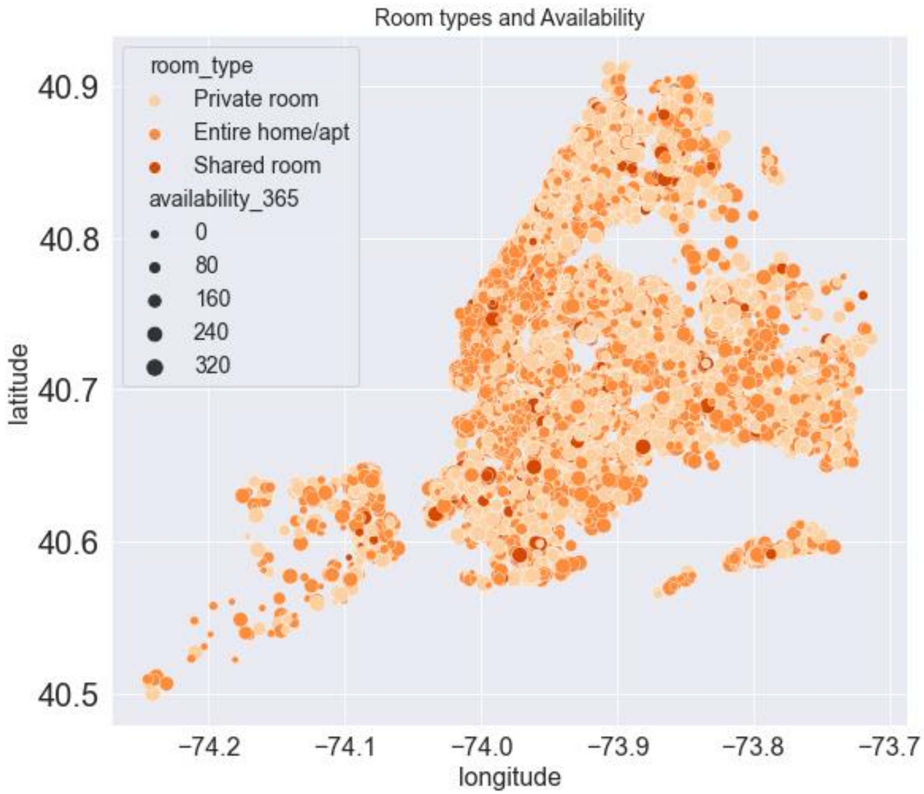
Graphical explorations: Analyzing Room types and Availability…

```python
# Analysis of the Room type and Availability
plt.figure(figsize=(9,8))
plt.title('Room types and Availability')
plt.rc('axes',titlesize=14)
sns.scatterplot(data=df,x='longitude',y='latitude',
               size='availability_365',
               hue=df.room_type,
               sizes=(20,100),
               palette='Oranges',
               s=70)
```

# Data Exploration

## ❑ Case Study:

Graphical explorations: Analyzing Room types and Availability…



# Data Exploration

- **Case Study:**

Graphical explorations:

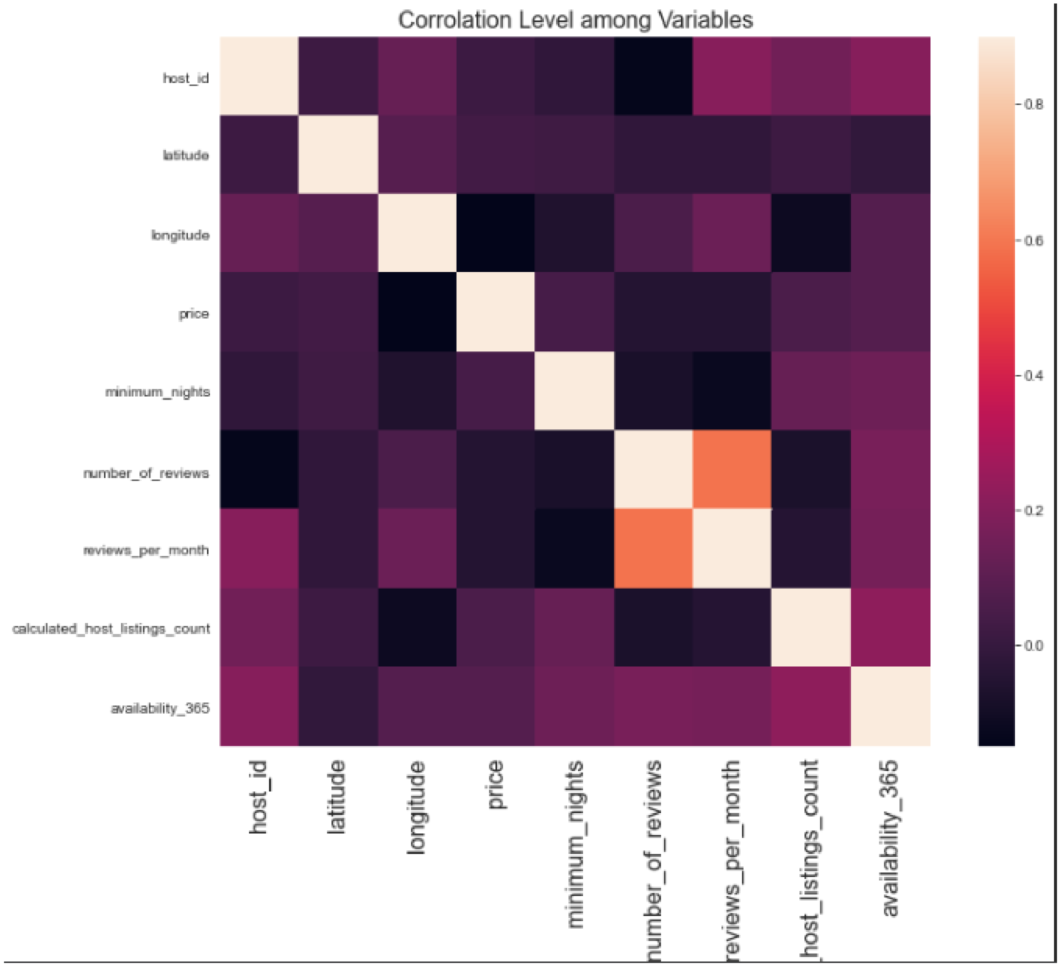
✓ We like to explore the correlation level among the data variables.

```
# Exploring the correlation level among the data variables.
corrmat = df.corr()  df: {DataFrame: (48895, 13)}
plt.subplots(figsize=(12,9))
sns.heatmap(corrmat, vmax=0.9, square=True)  corrmat: {DataFrame: (9, 9)}
plt.title('Corrolation Level among Variables')
```

# Data Exploration

## ❑ Case Study:

Graphical explorations:



# Data Exploration

## ❑ Case Study:

Graphical explorations:

**The above plot concludes:** The above graph presents the correlation level among data variables. We can observe that none of the variables are strictly correlated between each other.

# ❑ Data Exploration :

- ✓ In summary, exploring your data helps you observe hidden patterns that enables you determine the kind of analysis and model to adopt.

# Questions and Comments!

Thank You

中国科学院深圳先进技术研究院
SHENZHEN INSTITUTES OF ADVANCED TECHNOLOGY
CHINESE ACADEMY OF SCIENCES