1. Setup and Initial Inspection

[1]

```
## Load the Dataset
import pandas as pd
import numpy as np
```

[2]

```
## Read the csv file
df = pd.read_csv('Airbnb_Open_Data.csv')
```

```
<ipython-input-2-a7a2039eefd2>:2: DtypeWarning: Columns (25) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('Airbnb_Open_Data.csv')
```

[3]

```
## Display the first 5 rows
df.head()
```

[46]

```
## Get summary of Datasets
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 22 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   name                            102599 non-null  object 
 1   host_identity_verified          102599 non-null  object 
 2   host_name                       102599 non-null  object 
 3   neighbourhood_group             102599 non-null  object 
 4   neighbourhood                   102599 non-null  object 
 5   lat                             102599 non-null  float64
 6   long                            102599 non-null  float64
 7   instant_bookable                102599 non-null  bool   
 8   cancellation_policy             102599 non-null  object 
 9   room_type                       102599 non-null  object 
 10  construction_year               102599 non-null  float64
 11  price                           102599 non-null  object 
 12  service_fee                     102599 non-null  object 
 13  minimum_nights                  102599 non-null  float64
 14  number_of_reviews               102599 non-null  float64
 15  last_review                     102599 non-null  object 
 16  reviews_per_month               102599 non-null  float64
```

```
 17  review_rate_number              102599 non-null  float64
 18  calculated_host_listings_count  102599 non-null  float64
 19  days_booked                     102599 non-null  float64
 20  house_rules                     102599 non-null  object
 21  license                         102599 non-null  object
dtypes: bool(1), float64(9), object(12)
memory usage: 16.5+ MB
```

[4]

```
## Display the data type
df.dtypes
```

[47]

```
## Getting basic statistics for numerical columns
df.describe()
```

Task 2a: Data Cleaning

1. Drop some of the unwanted columns. These include host id, id, country and country code from the dataset. Please include the code in the cells below.

[7]

```
# Drop unwanted columns (host_id,id, country, country_code)
df.drop(columns=["id","host id","country code","country"], inplace=True)
```

[8]

```
df
```

Task 2b: Data Cleaning

1. Check for missing values in the dataframe and display the count in ascending order. If the values are missing, impute the values as per the datatype of the columns.
2. Check whether there are any duplicate values in the dataframe and, if present, remove them.
3. Display the total number of records in the dataframe before and after removing the duplicates.

[9]

```
## Check for missing values in the dataframe and display the count in ascending order.
df.isnull().sum().sort_values()
```

[10]

```
for col in df.columns:
  if df[col].dtype == 'O':
    print(col)
    df[str(col)].fillna(value=df[str(col)].mode()[0], inplace=True)
```

```
    else:
        df[str(col)].fillna(value=df[str(col)].median(), inplace=True)
```

```
NAME
host_identity_verified
host name
neighbourhood group
neighbourhood
instant_bookable
cancellation_policy
room type
price
service fee
last review
house_rules
license
```

[48]

```
## Check whether there are any duplicate values in the dataframe and if present remove them.
df.duplicated().sum()
```

3461

Task 3: Data Transformation

1. Rename the column availability 365 to days_booked
2. Convert all column names to lowercase and replace the spaces in the column names with an underscore "_". Please include the code in the cells below.

[13]

```
## Rename the column.
df.rename(columns={'availability 365':'Days_booked'},inplace=True)
```

[14]

```
## Convert all column names to lowercase and replace the spaces with an underscore "_"
df.columns=[col.lower().replace(" ","_") for col in df.columns]
df.columns
```

```
Index(['name', 'host_identity_verified', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'lat', 'long', 'instant_bookable',
       'cancellation_policy', 'room_type', 'construction_year', 'price',
       'service_fee', 'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'review_rate_number',
       'calculated_host_listings_count', 'days_booked', 'house_rules',
       'license'],
      dtype='object')
```

[15]

Display the total number of records in the dataframe after removing the duplicates.
df.shape

(102599, 22)

Task 4: Exploratory Data Analysis

1. List the count of various room types avaliable in the dataset.
2. Which room type has the most strict cancellation policy?
3.

a. Summary Statistics: Calculate basic statistics for numerical columns b. Visualizations: Create plots to explore data c. Correlation Analysis: Examine correlations between features

[16]

List the count of various room types avaliable with Airbnb
df[**'room_type'**].value_counts()

[20]

Which room type adheres to more strict cancellation policy
df_group_two= df[df[**'cancellation_policy'**]==**'strict'**]
df_group_two[**'room_type'**].value_counts()

[21]

a. Summary Statistics: Calculate basic statistics for numerical columns
print(df.describe())

```
              lat          long  construction_year  minimum_nights  \
count  102599.000000  102599.000000      102599.000000   102599.000000
mean       40.728093     -73.949644        2012.486447        8.115371
std         0.055854       0.049519           5.759583       30.494537
min        40.499790     -74.249840        2003.000000    -1223.000000
25%        40.688740     -73.982580        2008.000000        2.000000
50%        40.722290     -73.954440        2012.000000        3.000000
75%        40.762760     -73.932350        2017.000000        5.000000
max        40.916970     -73.705220        2022.000000     5645.000000

       number_of_reviews  reviews_per_month  review_rate_number  \
count      102599.000000      102599.000000       102599.000000
mean           27.447207           1.275896            3.278219
std            49.472332           1.622073            1.282711
min             0.000000           0.010000            1.000000
25%             1.000000           0.280000            2.000000
50%             7.000000           0.740000            3.000000
```

```
75%          30.000000          1.710000          4.000000
max        1024.000000         90.000000          5.000000

      calculated_host_listings_count    days_booked
count                102599.000000  102599.000000
mean                      7.915038     140.936179
std                      32.170972     135.171770
min                       1.000000     -10.000000
25%                       1.000000       3.000000
50%                       1.000000      96.000000
75%                       2.000000     268.000000
max                     332.000000    3677.000000
```

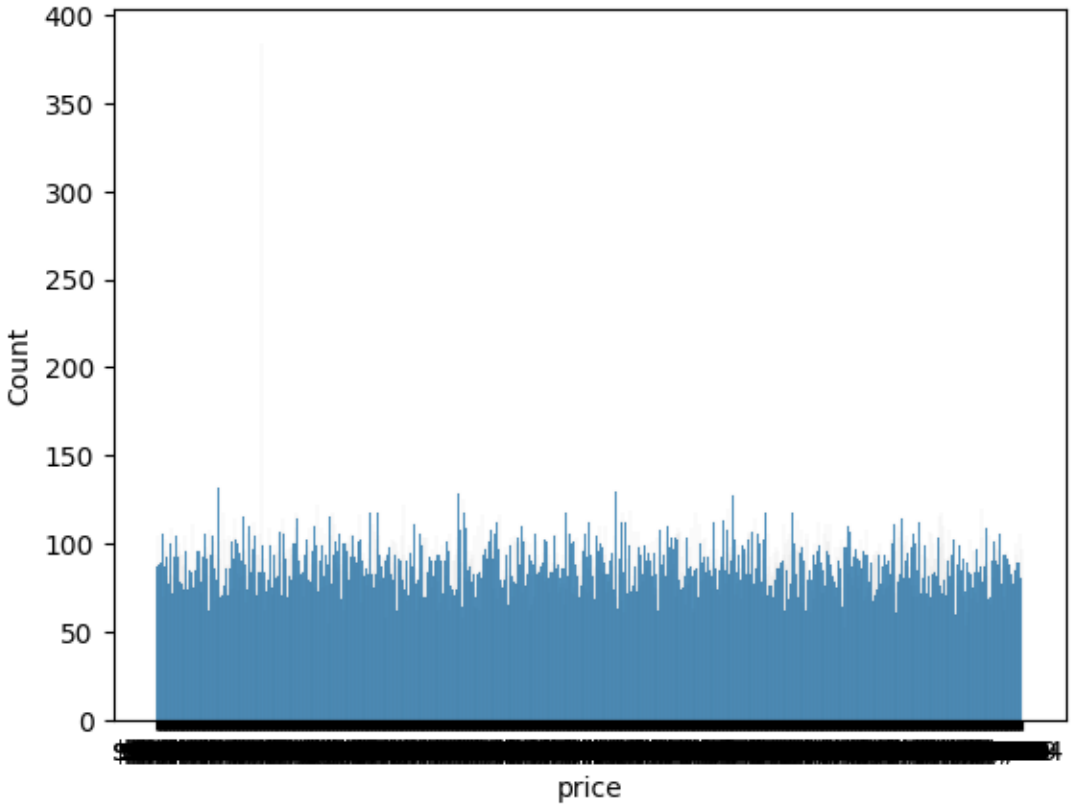
[30]

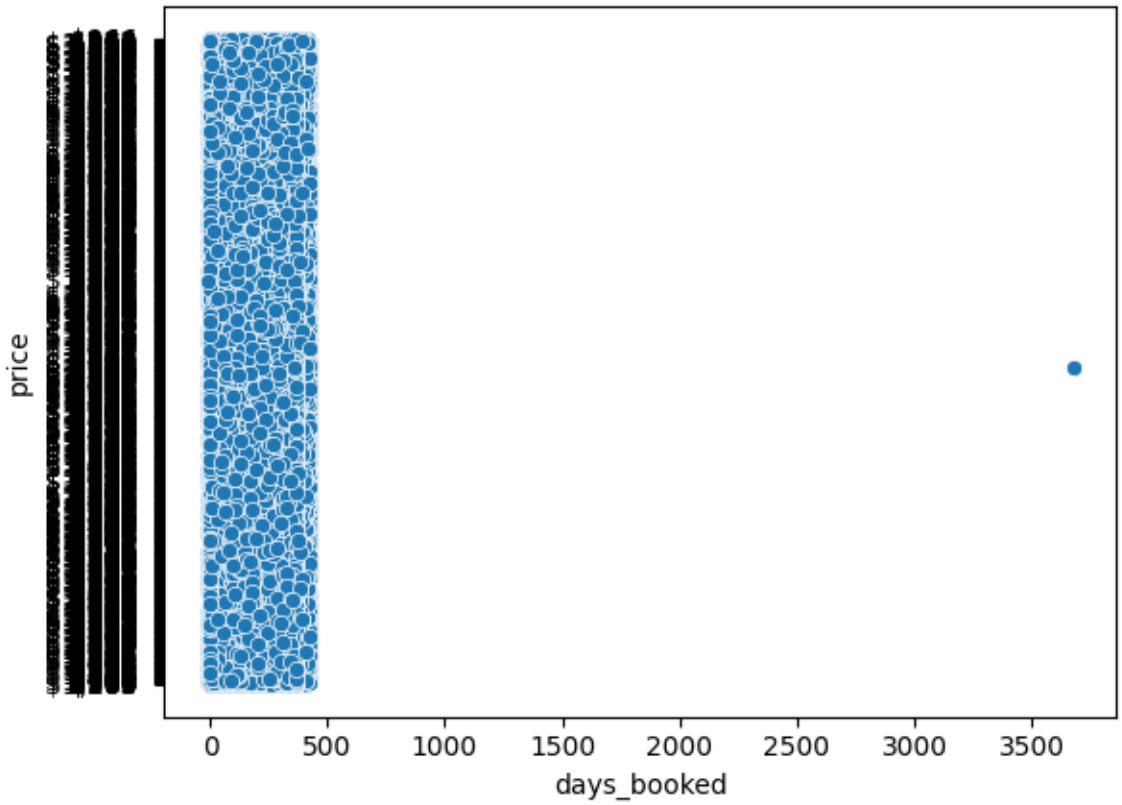
```
## b. Visualizations: Create plots to explore data
import seaborn as sns
import matplotlib.pyplot as plt

# Distribution of prices
sns.histplot(df['price'])
plt.show()

# Price vs. availability_365
print(df.columns)
sns.scatterplot(data=df, x='days_booked', y='price') # Fix column name to 'availability_365'
plt.show()
```

```
Index(['name', 'host_identity_verified', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'lat', 'long', 'instant_bookable',
       'cancellation_policy', 'room_type', 'construction_year', 'price',
       'service_fee', 'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'review_rate_number',
       'calculated_host_listings_count', 'days_booked', 'house_rules',
       'license'],
      dtype='object')
```


400
350
300
250
200
150
100
50
0
Count
price


price
0
500
1000
1500
2000
2500
3000
3500
days_booked

[33]

```
## c. Correlation Analysis: Examine correlations between features
## Select only numerical columns before calculating correlations

numerical_df = df.select_dtypes(include=['number'])

print(numerical_df.corr())
sns.heatmap(numerical_df.corr(), annot=True)
plt.show()
```

```
                                  lat      long  construction_year  \
lat                          1.000000  0.074348           0.005692
long                         0.074348  1.000000           0.000880
construction_year            0.005692  0.000880           1.000000
minimum_nights               0.014842 -0.039502          -0.000498
number_of_reviews           -0.025245  0.068999           0.001835
reviews_per_month           -0.021800  0.116834           0.003721
review_rate_number          -0.003982  0.015265           0.004792
calculated_host_listings_count  0.032357 -0.104034         -0.002699
days_booked                 -0.004960  0.058294          -0.008388

                                minimum_nights  number_of_reviews  \
lat                               0.014842          -0.025245
long                             -0.039502           0.068999
construction_year                -0.000498           0.001835
minimum_nights                    1.000000          -0.049860
number_of_reviews                -0.049860           1.000000
reviews_per_month                -0.087013           0.601314
review_rate_number               -0.002093          -0.018608
calculated_host_listings_count    0.084622          -0.080699
days_booked                       0.058783           0.098399

                                reviews_per_month  review_rate_number  \
lat                                 -0.021800          -0.003982
long                                 0.116834           0.015265
construction_year                    0.003721           0.004792
minimum_nights                      -0.087013          -0.002093
number_of_reviews                    0.601314          -0.018608
reviews_per_month                    1.000000           0.033897
review_rate_number                   0.033897           1.000000
calculated_host_listings_count      -0.030541           0.024365
days_booked                          0.071948          -0.006581

                                calculated_host_listings_count  days_booked
```

lat	0.032357	-0.004960
long	-0.104034	0.058294
construction_year	-0.002699	-0.008388
minimum_nights	0.084622	0.058783
number_of_reviews	-0.080699	0.098399
reviews_per_month	-0.030541	0.071948
review_rate_number	0.024365	-0.006581
calculated_host_listings_count	1.000000	0.158876
days_booked	0.158876	1.000000