```
import pandas as pd

file_path = '/content/drive/My Drive/AirBnB_data.csv'
df = pd.read_csv(file_path)
```

```
/tmp/ipython-input-4-397545261.py:4: DtypeWarning: Columns (25) have mixed types. Specify dtype option on import or s
  df = pd.read_csv(file_path)
```

```
#Import libraries

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

```
df.head()
```

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harlem	40.80902	-73.94190
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399

5 rows × 26 columns

```
df.columns
```

```
Index(['id', 'NAME', 'host id', 'host_identity_verified', 'host name',
       'neighbourhood group', 'neighbourhood', 'lat', 'long', 'country',
       'country code', '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',
       'availability 365', 'house_rules', 'license'],
      dtype='object')
```

1. Checking missing values

```
#Checking missing values
print(df.isnull().sum())
```

```
id                                    0
NAME                                250
host id                               0
host_identity_verified              289
host name                           406
```

```
neighbourhood group                  29
neighbourhood                        16
lat                                   8
long                                  8
country                             532
country code                        131
instant_bookable                    105
cancellation_policy                  76
room type                             0
Construction year                   214
price                               247
service fee                         273
minimum nights                      409
number of reviews                   183
last review                       15893
reviews per month                 15879
review rate number                  326
calculated host listings count      319
availability 365                    448
house_rules                       52131
license                          102597
dtype: int64
```

```
#Checking the data type of each column
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   object
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: float64(9), int64(2), object(15)
memory usage: 20.4+ MB
```

```
#Format data type of last review: object (string) --> date
df['last review'] = pd.to_datetime(df['last review'], errors = 'coerce')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102599 entries, 0 to 102598
Data columns (total 26 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   id                              102599 non-null  int64
 1   NAME                            102349 non-null  object
 2   host id                         102599 non-null  int64
 3   host_identity_verified          102310 non-null  object
 4   host name                       102193 non-null  object
 5   neighbourhood group             102570 non-null  object
 6   neighbourhood                   102583 non-null  object
 7   lat                             102591 non-null  float64
```

```
 8   long                            102591 non-null  float64
 9   country                         102067 non-null  object
 10  country code                    102468 non-null  object
 11  instant_bookable                102494 non-null  object
 12  cancellation_policy             102523 non-null  object
 13  room type                       102599 non-null  object
 14  Construction year               102385 non-null  float64
 15  price                           102352 non-null  object
 16  service fee                     102326 non-null  object
 17  minimum nights                  102190 non-null  float64
 18  number of reviews               102416 non-null  float64
 19  last review                     86706 non-null   datetime64[ns]
 20  reviews per month               86720 non-null   float64
 21  review rate number              102273 non-null  float64
 22  calculated host listings count  102280 non-null  float64
 23  availability 365                102151 non-null  float64
 24  house_rules                     50468 non-null   object
 25  license                         2 non-null       object
dtypes: datetime64[ns](1), float64(9), int64(2), object(14)
memory usage: 20.4+ MB
```

2. Handling missing values (from the most to the least)

```
#1. 'last review'  -->  Replace NaN in last review wich minimum date

df.fillna({'last review' : df['last review'].min()}, inplace = True)

print(df.isnull().sum())
```

```
id                                  0
NAME                              250
host id                             0
host_identity_verified            289
host name                         406
neighbourhood group                29
neighbourhood                      16
lat                                 8
long                                8
country                           532
country code                      131
instant_bookable                  105
cancellation_policy                76
room type                           0
Construction year                 214
price                             247
service fee                       273
minimum nights                    409
number of reviews                 183
last review                         0
reviews per month               15879
review rate number                326
calculated host listings count    319
availability 365                  448
house_rules                     52131
license                        102597
dtype: int64
```

```
#2. 'reviews per month'  -->  Replace NaN in reviews per month with 0
df.fillna({'reviews per month' : 0}, inplace = True)

print(df.isnull().sum())
```

```
id                                  0
NAME                              250
host id                             0
host_identity_verified            289
host name                         406
neighbourhood group                29
neighbourhood                      16
lat                                 8
long                                8
country                           532
country code                      131
instant_bookable                  105
cancellation_policy                76
```

```
room type                              0
Construction year                    214
price                                247
service fee                          273
minimum nights                       409
number of reviews                    183
last review                            0
reviews per month                      0
review rate number                   326
calculated host listings count       319
availability 365                     448
house_rules                        52131
license                           102597
dtype: int64
```

```
#3. 'NAME' + 'host name'   -->  Drop NaN
df.dropna(subset = ['NAME', 'host name'], inplace = True)

print(df.isnull().sum())
```

```
id                                     0
NAME                                   0
host id                                0
host_identity_verified               276
host name                              0
neighbourhood group                   26
neighbourhood                         16
lat                                    8
long                                   8
country                              526
country code                         122
instant_bookable                      96
cancellation_policy                   70
room type                              0
Construction year                    200
price                                239
service fee                          268
minimum nights                       403
number of reviews                    182
last review                            0
reviews per month                      0
review rate number                   314
calculated host listings count       318
availability 365                     420
house_rules                        51867
license                           101947
dtype: int64
```

```
#4. 'house_rules' + 'license'  --> Drop
df.drop(columns = ['house_rules', 'license'], inplace = True, errors='ignore')

print(df.isnull().sum())
```

```
id                                 0
NAME                               0
host id                            0
host_identity_verified           276
host name                          0
neighbourhood group               26
neighbourhood                     16
lat                                8
long                               8
country                          526
country code                     122
instant_bookable                  96
cancellation_policy               70
room type                          0
Construction year                200
price                            239
service fee                      268
minimum nights                   403
number of reviews                182
last review                        0
reviews per month                  0
review rate number               314
calculated host listings count   318
availability 365                 420
```

```
dtype: int64
```

```
#5. 'price' + 'service fee'   -->   Remove the '$' to avoid the errors + Reformat data type:  object (string) --> float
df['price'] = df['price'].replace('[$,]', '',regex = True).astype(float)

df['service fee'] = df['service fee'].replace('[$,]', '',regex = True).astype(float)

print(df.isnull().sum())
```

```
id                                0
NAME                              0
host id                           0
host_identity_verified          276
host name                         0
neighbourhood group              26
neighbourhood                    16
lat                               8
long                              8
country                         526
country code                    122
instant_bookable                 96
cancellation_policy              70
room type                         0
Construction year               200
price                           239
service fee                     268
minimum nights                  403
number of reviews               182
last review                       0
reviews per month                 0
review rate number              314
calculated host listings count  318
availability 365                420
dtype: int64
```

```
df.head()
```

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harlem	40.80902	-73.94190
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399
5	1004098	Large Cozy 1 BR Apartment In Midtown East	45498551794	verified	Michelle	Manhattan	Murray Hill	40.74767	-73.97500

5 rows × 24 columns

3. Remove duplicates

```
#Cheking for duplicates

print(len(df))
print(df.shape)
print(df.duplicated().sum())
print(df[df.duplicated()])
```

Hiện kết quả đã ẩn

```
#Remove duplicates

df.drop_duplicates(inplace = True)

print(df.duplicated().sum())
```

```
0
```

4. Cleaning typo data

```
#Checking for typo data:

df['neighbourhood group'].unique()
df['neighbourhood group'].value_counts()

#Replace correct data:

df['neighbourhood group'] = df['neighbourhood group'].replace({
    'brookln': 'Brooklyn',
    'manhatan': 'Manhattan'
})

df['neighbourhood group'].value_counts()
```

neighbourhood group	count
Manhattan	43279
Brooklyn	41364
Queens	13120
Bronx	2678
Staten Island	943

dtype: int64

5. Descrptive Statistics

```
df.describe()
```

	id	host id	lat	long	Construction year	price	service fee	minimum nights	num r
count	1.014100e+05	1.014100e+05	101402.000000	101402.000000	101210.000000	101171.000000	101142.000000	101016.000000	101228
mean	2.920959e+07	4.926155e+10	40.728082	-73.949663	2012.486908	625.381008	125.043998	8.113744	27
min	1.001254e+06	1.236005e+08	40.499790	-74.249840	2003.000000	50.000000	10.000000	-1223.000000	0
25%	1.507574e+07	2.459183e+10	40.688730	-73.982570	2007.000000	340.000000	68.000000	2.000000	1
50%	2.922911e+07	4.912069e+10	40.722300	-73.954440	2012.000000	625.000000	125.000000	3.000000	7
75%	4.328308e+07	7.399747e+10	40.762750	-73.932340	2017.000000	913.000000	183.000000	5.000000	31
max	5.736742e+07	9.876313e+10	40.916970	-73.705220	2022.000000	1200.000000	240.000000	5645.000000	1024
std	1.626820e+07	2.853703e+10	0.055850	0.049474	5.765130	331.609111	66.313374	30.378014	49

6. Visualization

a. What is the distribution of listing prices?

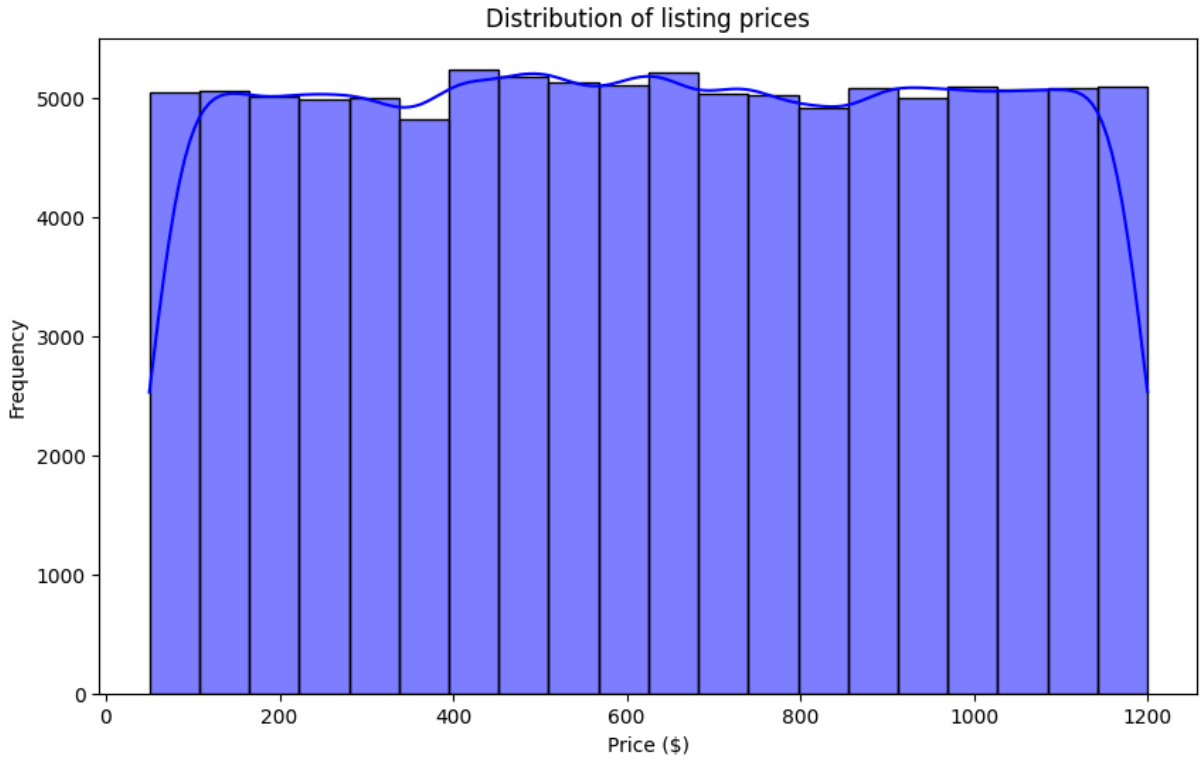
Answer:

- The histogram shows a fairly even distribution of listing prices across different price ranges, indication no particular concentration of listings in any specifc range.
- The KDE line helps visualize this even spread more clearly, confirming that the dataset contains listings with a wide variety of prices.

```
# Histogram
plt.figure(figsize = (10,6))
sns.histplot(df['price'], bins = 20, kde = True, color = 'blue') #kde là đường cong xác suất của data

plt.title('Distribution of listing prices')
plt.xlabel('Price ($)')
plt.ylabel('Frequency')

plt.show()
```



b. How are different room types distributed?

Answer:

- Entire room/ Apt > Private room > Shared room > Hotel room

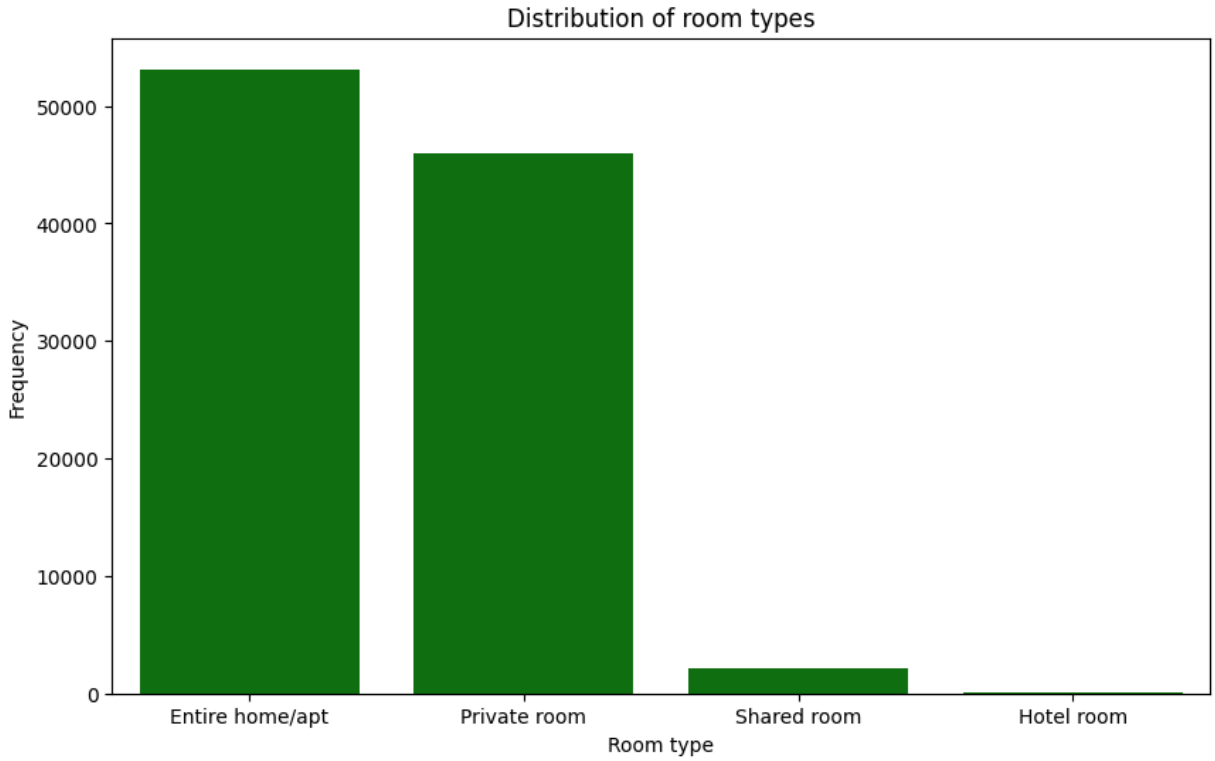
```
plt.figure(figsize = (10,6))

#Descending Order (default of value_counts())
order = df['room type'].value_counts().index
#Ascending Order: order = df['col'].value_counts().sort_values(ascending=True).index

sns.countplot  (data = df, x = 'room type', order = order, color = 'green') #Add the feature: bins in desc order
#Có thể gộp thành 1line như này: sns.countplot  (data = df, x = 'room type', order = df['room type'].value_counts().inde


plt.title('Distribution of room types')
plt.xlabel('Room type')
plt.ylabel('Frequency')

plt.show()
```



c. How are listings distributed across neighborhoods?

Answer:

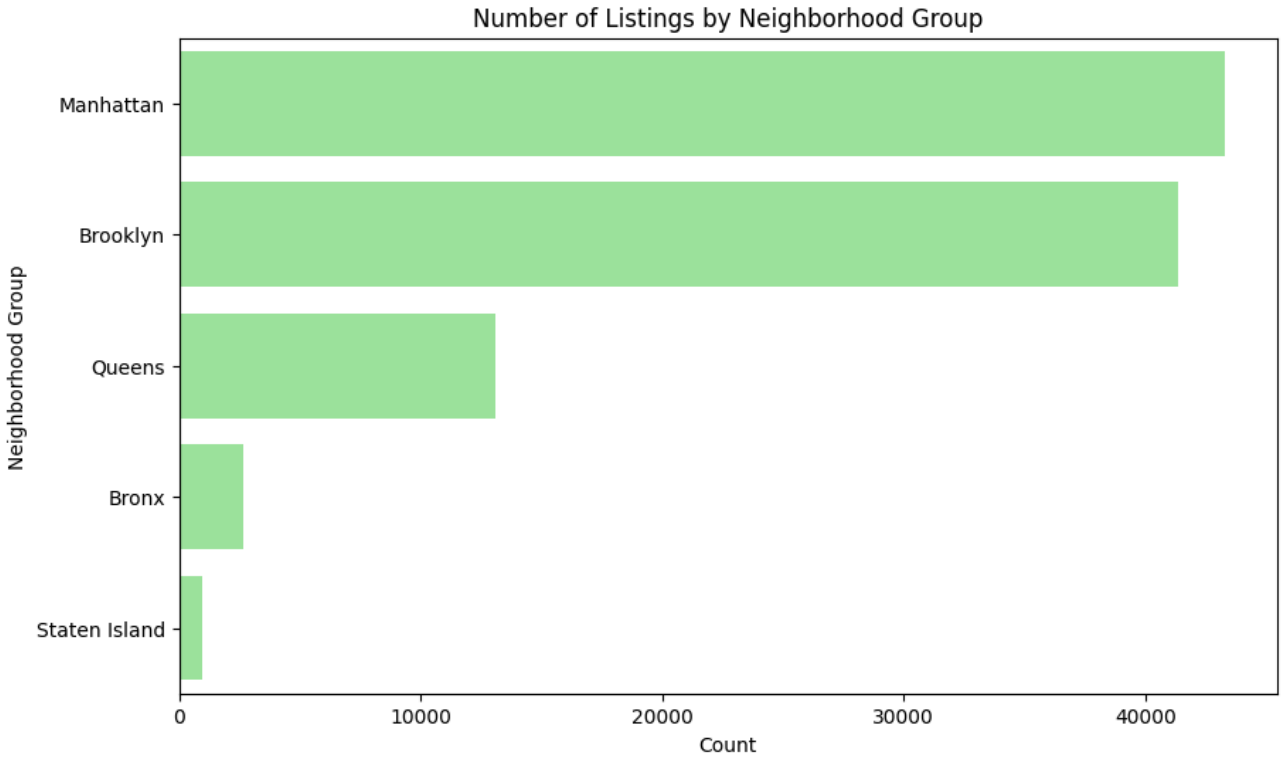
- Manhattan > Brooklyn > Queens > Bronx > Staten Island

```
plt.figure(figsize= (10, 6))

sns.countplot(y='neighbourhood group', data=df,color="lightgreen" , order=df ['neighbourhood group']. value_counts () .i

plt.title( 'Number of Listings by Neighborhood Group')
plt.xlabel('Count')
plt.ylabel('Neighborhood Group')

plt.show()
```



d. What is the relationship between price & roomtype?

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='room type', y='price', data=df, palette='Set2')

plt.title('Price Distribution by Room Type')
plt.xlabel('Room Type')
plt.ylabel('Price ($)')
plt.legend(title='Room Type')

plt.show()
```

```
/tmp/ipython-input-24-2347991949.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h

  sns.boxplot(x='room type', y='price', data=df, palette='Set2')
```