

## airbnb-project-1

```
[10]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[11]: df = pd.read_csv("Airbnb_Open_Data.csv")
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16660\3424017332.py:1: DtypeWarning:
Columns (25) have mixed types. Specify dtype option on import or set
low_memory=False.
df = pd.read_csv("Airbnb_Open_Data.csv")
```

```
[12]: df.head()
```

```
[12]:      id      NAME      host id \
0  1001254      Clean & quiet apt home by the park  80014485718
1  1002102      Skylit Midtown Castle  52335172823
2  1002403      THE VILLAGE OF HARLEM...NEW YORK !  78829239556
3  1002755      NaN  85098326012
4  1003689  Entire Apt: Spacious Studio/Loft by central park  92037596077
```

```
      host_identity_verified host name neighbourhood group neighbourhood \
0      unconfirmed      Madaline      Brooklyn      Kensington
1      verified      Jenna      Manhattan      Midtown
2      NaN      Elise      Manhattan      Harlem
3      unconfirmed      Garry      Brooklyn      Clinton Hill
4      verified      Lyndon      Manhattan      East Harlem
```

```
      lat      long      country ... service fee minimum nights \
0  40.64749 -73.97237 United States ...      $193      10.0
1  40.75362 -73.98377 United States ...      $28      30.0
2  40.80902 -73.94190 United States ...      $124      3.0
3  40.68514 -73.95976 United States ...      $74      30.0
4  40.79851 -73.94399 United States ...      $41      10.0
```

```
      number of reviews last review      reviews per month review rate number \
0      9.0  10/19/2021      0.21      4.0
1      45.0  5/21/2022      0.38      4.0
```

2	0.0	NaN	NaN	5.0
3	270.0	7/5/2019	4.64	4.0
4	9.0	11/19/2018	0.10	3.0

	calculated host listings count	availability 365 \
0	6.0	286.0
1	2.0	228.0
2	1.0	352.0
3	1.0	322.0
4	1.0	289.0

	house_rules	license
0	Clean up and treat the home the way you'd like...	NaN
1	Pet friendly but please confirm with me if the...	NaN
2	I encourage you to use my kitchen, cooking and...	NaN
3		NaN NaN
4	Please no smoking in the house, porch or on th...	NaN

[5 rows x 26 columns]

```
[13]: df.columns
```

```
[13]: 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')
```

### 0.0.1 Checking Missing Values

```
[14]: df.isnull().sum()
```

```
[14]: 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

```

```
[15]: 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

```

## 0.0.2 Handling Missing Values

```
[16]: df["last review"] = pd.to_datetime(df["last review"], errors="coerce")
```

```
[17]: df.fillna({"reviews per month": 0, "last review": df["last review"].min()},
             ↪inplace = True)
```

```
[18]: df.dropna(subset={"NAME", "host name"}, inplace = True)
```

```
[19]: df.isnull().sum()
```

```

[19]: 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

```

```
[20]: df = df.drop(columns={"house_rules", "license"}, errors="ignore")
```

```
[21]: df.head()
```

```
[21]:
```

	id	NAME	host id	\
0	1001254	Clean & quiet apt home by the park	80014485718	
1	1002102	Skylit Midtown Castle	52335172823	
2	1002403	THE VILLAGE OF HARLEM...NEW YORK !	78829239556	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	
5	1004098	Large Cozy 1 BR Apartment In Midtown East	45498551794	

	host_identity_verified	host name	neighbourhood	group	neighbourhood	\
0	unconfirmed	Madaline	Brooklyn		Kensington	
1	verified	Jenna	Manhattan		Midtown	
2	NaN	Elise	Manhattan		Harlem	
4	verified	Lyndon	Manhattan		East Harlem	
5	verified	Michelle	Manhattan		Murray Hill	

	lat	long	country	...	Construction year	price	\
0	40.64749	-73.97237	United States	...	2020.0	\$966	
1	40.75362	-73.98377	United States	...	2007.0	\$142	
2	40.80902	-73.94190	United States	...	2005.0	\$620	
4	40.79851	-73.94399	United States	...	2009.0	\$204	
5	40.74767	-73.97500	United States	...	2013.0	\$577	

	service fee	minimum nights	number of reviews	last review	reviews per month	\
0	\$193	10.0	9.0	2021-10-19	0.21	
1	\$28	30.0	45.0	2022-05-21	0.38	
2	\$124	3.0	0.0	2012-07-11	0.00	
4	\$41	10.0	9.0	2018-11-19	0.10	
5	\$115	3.0	74.0	2019-06-22	0.59	

	review rate	number	calculated host listings	count	availability	365
0		4.0		6.0		286.0
1		4.0		2.0		228.0
2		5.0		1.0		352.0
4		3.0		1.0		289.0
5		3.0		1.0		374.0

[5 rows x 24 columns]

```
[22]: # Remove dollar sign and convert to float

df["price"] = df["price"].replace(r"[\$,]", "", regex=True).astype(float)
df["service fee"] = df["service fee"].replace(r"[\$,]", "", regex=True).
    ↪astype(float)
```

### 0.0.3 Remove Duplicates

```
[23]: df.drop_duplicates(inplace=True)
```

```
[24]: df.describe()
```

```
[24]:
```

	id	host id	lat	long \
count	1.014100e+05	1.014100e+05	101402.000000	101402.000000
mean	2.920959e+07	4.926155e+10	40.728082	-73.949663
min	1.001254e+06	1.236005e+08	40.499790	-74.249840
25%	1.507574e+07	2.459183e+10	40.688730	-73.982570
50%	2.922911e+07	4.912069e+10	40.722300	-73.954440
75%	4.328308e+07	7.399747e+10	40.762750	-73.932340
max	5.736742e+07	9.876313e+10	40.916970	-73.705220
std	1.626820e+07	2.853703e+10	0.055850	0.049474

	Construction year	price	service fee	minimum nights \
count	101210.000000	101171.000000	101142.000000	101016.000000
mean	2012.486908	625.381008	125.043998	8.113744
min	2003.000000	50.000000	10.000000	-1223.000000
25%	2007.000000	340.000000	68.000000	2.000000
50%	2012.000000	625.000000	125.000000	3.000000
75%	2017.000000	913.000000	183.000000	5.000000
max	2022.000000	1200.000000	240.000000	5645.000000
std	5.765130	331.609111	66.313374	30.378014

	number of reviews	last review	reviews per month \
count	101228.000000	101410	101410.000000
mean	27.511854	2018-05-15 21:26:08.721033728	1.163207
min	0.000000	2012-07-11 00:00:00	0.000000
25%	1.000000	2017-07-30 00:00:00	0.090000
50%	7.000000	2019-05-23 00:00:00	0.480000
75%	31.000000	2019-07-01 00:00:00	1.710000
max	1024.000000	2058-06-16 00:00:00	90.000000
std	49.549258	NaN	1.683708

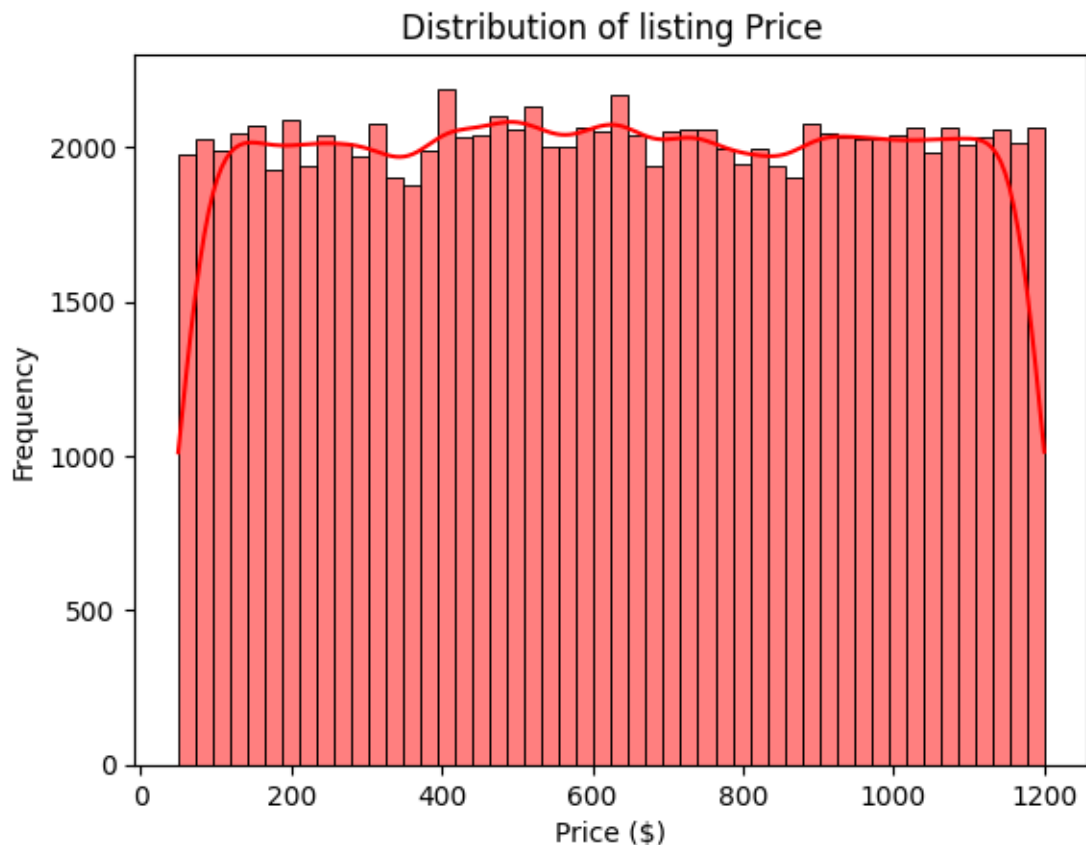
  

	review rate number	calculated host listings count	availability 365
count	101103.000000	101092.000000	100990.000000
mean	3.278558	7.948463	141.164660
min	1.000000	1.000000	-10.000000
25%	2.000000	1.000000	3.000000
50%	3.000000	1.000000	96.000000
75%	4.000000	2.000000	269.000000
max	5.000000	332.000000	3677.000000
std	1.285369	32.328974	135.419199

# 1 Visualization

## 1.1 What is the Distribution of listing Pricing?

```
[25]: sns.histplot(df["price"], bins=50, kde=True, color="red")
plt.xlabel("Price ($)")
plt.ylabel("Frequency")
plt.title("Distribution of listing Price")
plt.show()
```

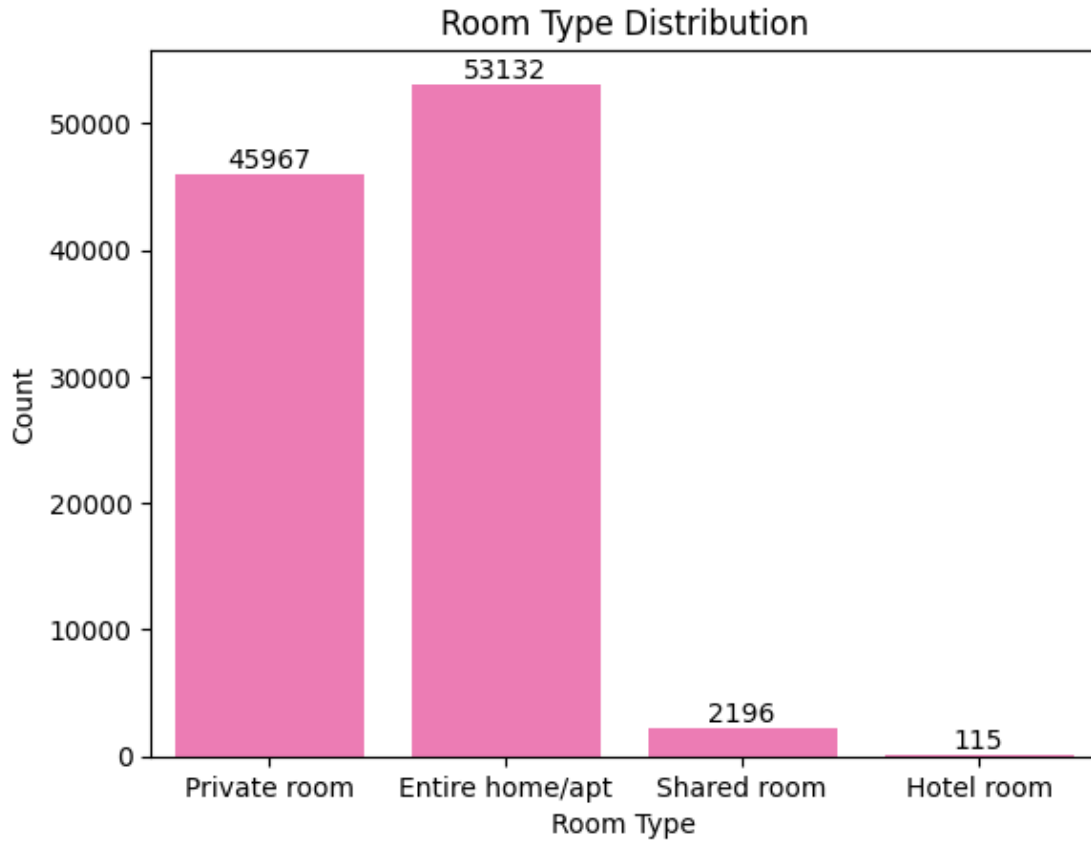


#The histogram shows a fairly even distribution of listing prices across different price ranges, indicating no particular concentration of listings in any specific price range. The KDE line helps visualize this even spread more clearly, confirming that the dataset contains listings with a wide variety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

## 1.2 How are different room types distributed?

```
[26]: ax = sns.countplot(x="room type", data=df, color="hotpink")
ax.bar_label(ax.containers[0])
```

```
plt.xlabel("Room Type")
plt.ylabel("Count")
plt.title("Room Type Distribution")
plt.show()
```



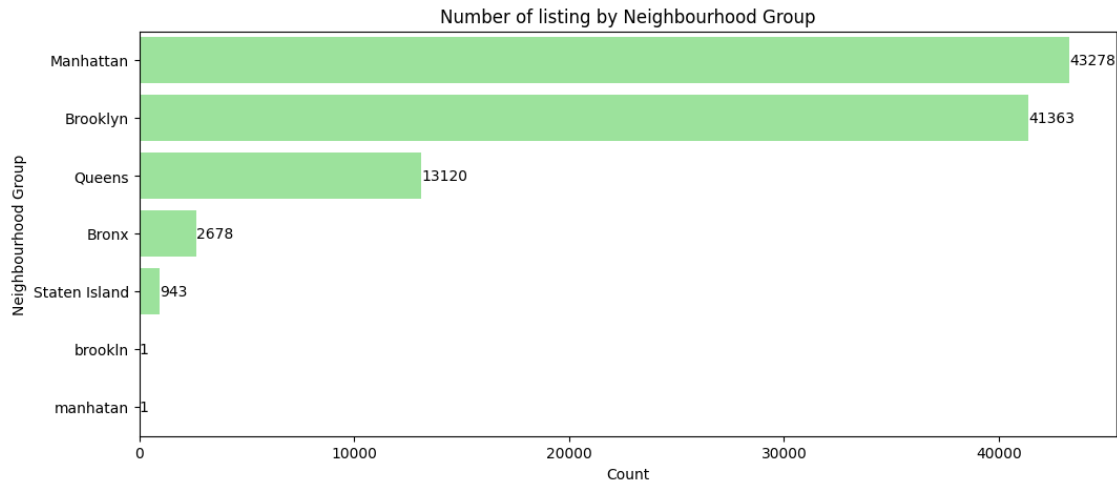
### 1.3 How are listings distributed accross differnet neighborhoods?

```
[27]: plt.figure(figsize=(12,5))

ax = sns.countplot(y = "neighbourhood group", data = df, color="lightgreen",
                  order=df["neighbourhood group"].value_counts().index)
ax.bar_label(ax.containers[0])

plt.title("Number of listing by Neighbourhood Group")
plt.xlabel("Count")
plt.ylabel("Neighbourhood Group")
plt.show()
```



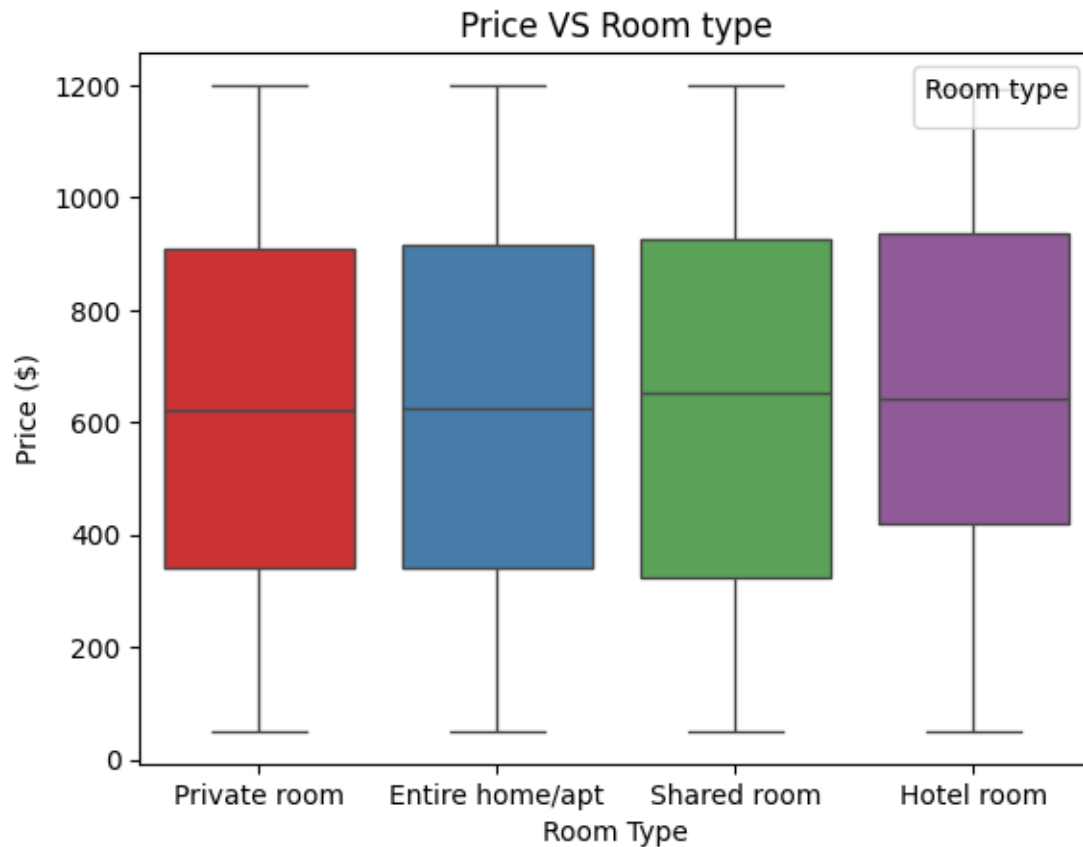


#### 1.4 What is the relationship between Price and Room Type?

```
[28]: sns.boxplot(y= "price",x="room type", data=df, palette="Set1", hue="room type")

plt.xlabel("Room Type")
plt.ylabel("Price ($)")
plt.title("Price VS Room type")
plt.legend(title="Room type", loc="upper right")
plt.show()
```

C:\Users\DELL\AppData\Local\Temp\ipykernel\_16660\1908325673.py:6: UserWarning:  
No artists with labels found to put in legend. Note that artists whose label  
start with an underscore are ignored when legend() is called with no argument.  
plt.legend(title="Room type", loc="upper right")



### 1.5 How has the numbers of reviews change over with time ?

```
[42]: df["last reivew"] = pd.to_datetime(df["last review"])
reviews_over_time = df.groupby(df["last review"].dt.to_period("M")).size()

reviews_over_time.plot(kind="line", color="red")
plt.title("Number of Review Overtime")
plt.xlabel("Date")
plt.ylabel("Number of Reviews")
plt.show()
```

