

# ***THE SECRETS OF AIRBNB IN NYC: DATA METHODOLOGY***

***By:  
Shashwat bhansali***

## Importing libraries and reading the data

```
In [5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [6]: df = pd.read_csv('AB_NYC_2019.csv')
df.head()
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_review
--	----	------	---------	-----------	---------------------	---------------	----------	-----------	-----------	-------	----------------	------------------

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	

# Data wrangling

```
In [7]: df.shape
```

```
Out[7]: (48895, 16)
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 48895 entries, 0 to 48894  
Data columns (total 16 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   id                    48895 non-null  int64    
1   name                  48879 non-null  object   
2   host_id               48895 non-null  int64    
3   host_name             48874 non-null  object   
4   neighbourhood_group    48895 non-null  object   
5   neighbourhood          48895 non-null  object   
6   latitude              48895 non-null  float64  
7   longitude              48895 non-null  float64  
8   room_type             48895 non-null  object   
9   price                 48895 non-null  int64    
10  minimum_nights        48895 non-null  int64    
11  number_of_reviews     48895 non-null  int64    
12  last_review            38843 non-null  object   
13  reviews_per_month     38843 non-null  float64  
14  calculated_host_listings_count  48895 non-null  int64    
15  availability_365       48895 non-null  int64    
dtypes: float64(3), int64(7), object(6)  
memory usage: 6.0+ MB
```

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

```
Out[9]: 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
```

```
In [10]: df.describe()
```

```
Out[10]:
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.

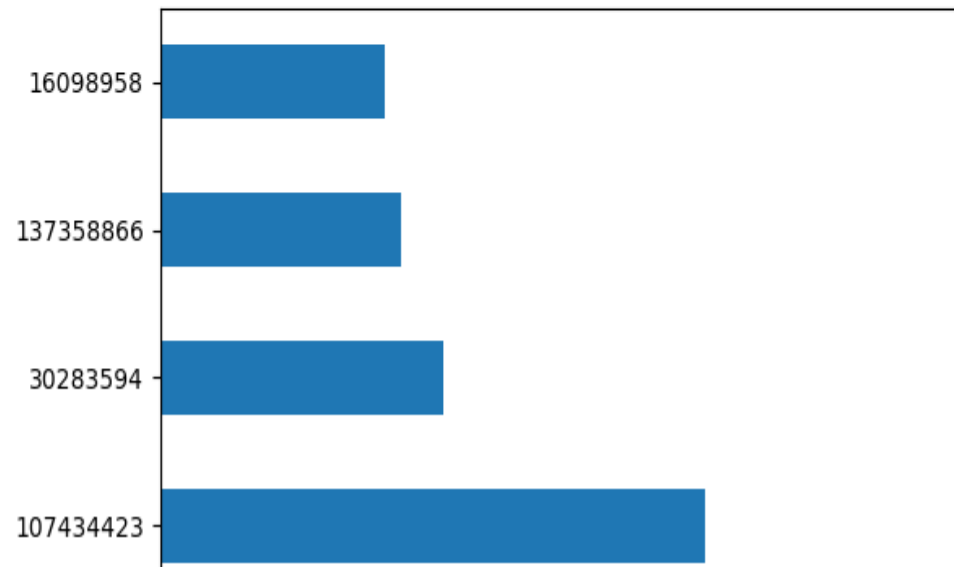
# HOST ID

```
In [13]: df.host_id.value_counts().iloc[:5]
```

```
Out[13]: 219517861    327  
         107434423    232  
         30283594    121  
         137358866    103  
         16098958     96  
         Name: host_id, dtype: int64
```

```
In [14]: df.host_id.value_counts().iloc[:5].plot(kind = 'barh')
```

```
Out[14]: <Axes: >
```



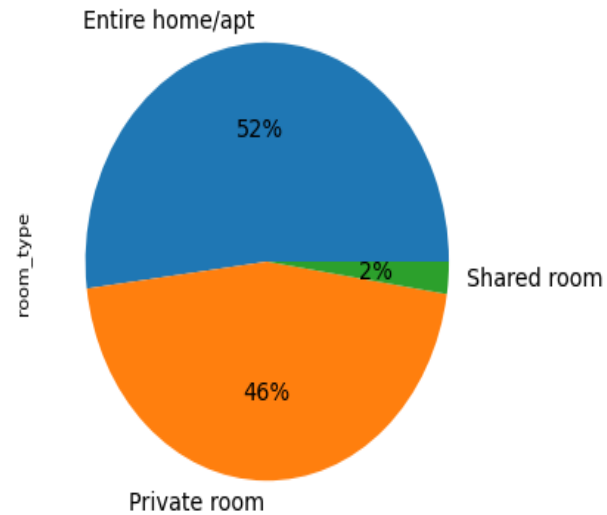
# Room Type

```
In [15]: df['room_type'].value_counts()
```

```
Out[15]: Entire home/apt    25409  
Private room    22326  
Shared room      1160  
Name: room_type, dtype: int64
```

```
In [16]: fig = plt.figure(figsize=(5,5), dpi=80)  
df['room_type'].value_counts().plot(kind='pie', autopct='%1.0f%%', startangle=360, fontsize=13)
```

```
Out[16]: <Axes: ylabel='room_type'>
```



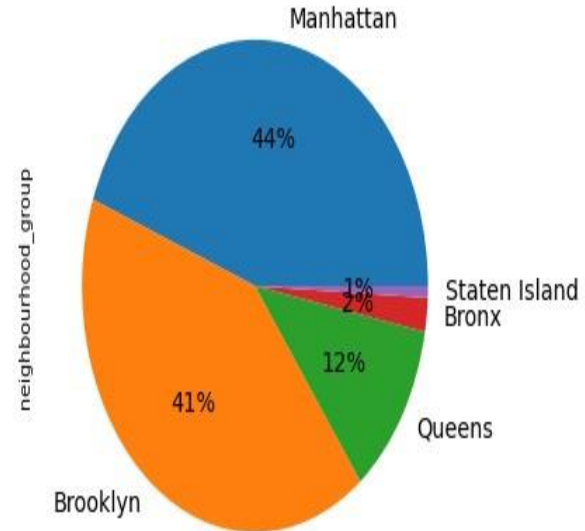
# Neighbourhood group

```
In [17]: df['neighbourhood_group'].value_counts()
```

```
Out[17]: Manhattan      21661  
         Brooklyn      20104  
         Queens         5666  
         Bronx          1091  
         Staten Island    373  
         Name: neighbourhood_group, dtype: int64
```

```
In [20]: fig = plt.figure(figsize=(5,5), dpi=80)  
         df['neighbourhood_group'].value_counts().plot(kind='pie', autopct='%1.0f%%', startangle=360, fontsize=13)
```

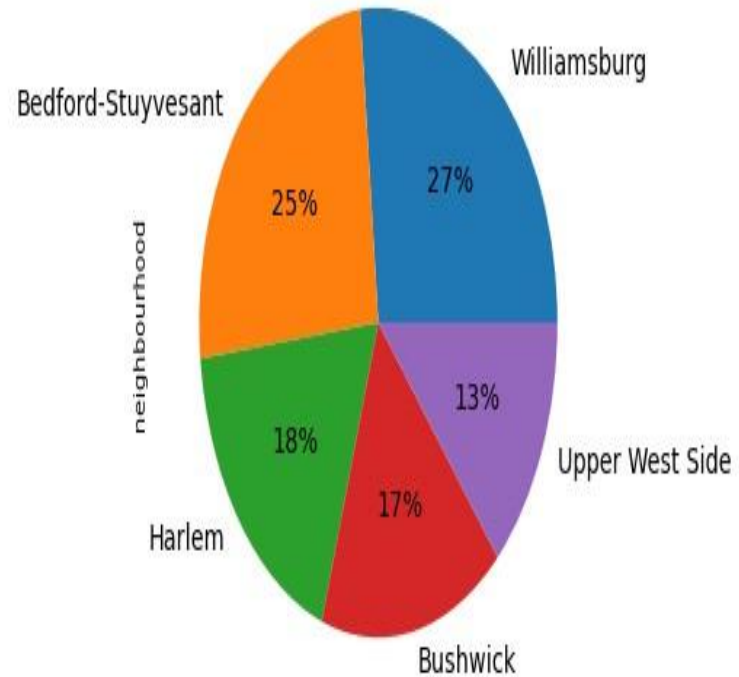
```
Out[20]: <Axes: ylabel='neighbourhood_group'>
```



# Neighbourhood

```
In [23]: fig = plt.figure(figsize=(5,5), dpi=80)  
df['neighbourhood'].value_counts().iloc[:5].plot(kind='pie', autopct='%1.0f%%', startangle=360, fontsize=13)
```

```
Out[23]: <Axes: ylabel='neighbourhood'>
```

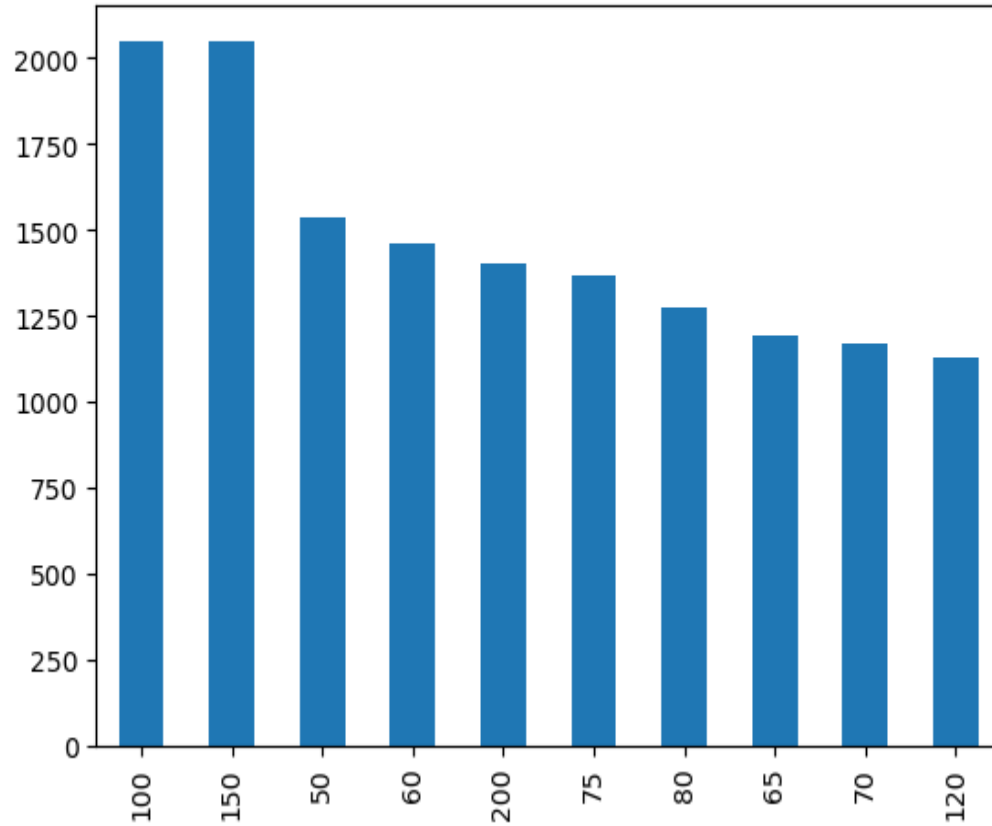




# Price

```
In [25]: df.price.value_counts().iloc[:10].plot(kind = 'bar')
```

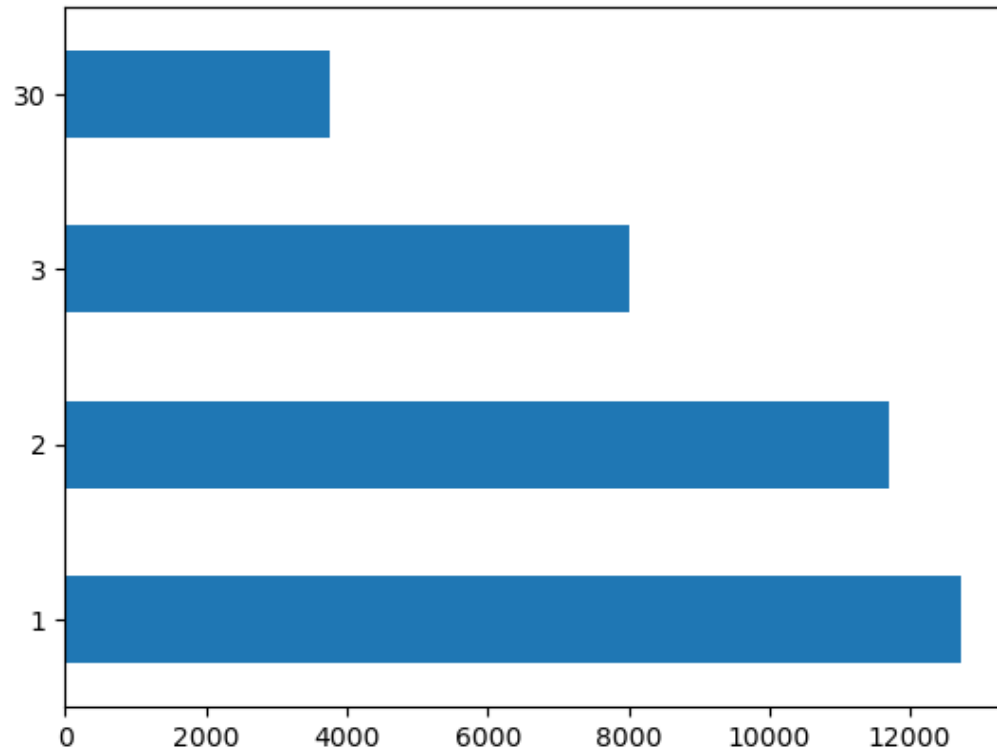
```
Out[25]: <Axes: >
```



# *Minimum Nights*

```
In [30]: df['minimum_nights'].value_counts().iloc[:4].plot(kind = 'barh')
```

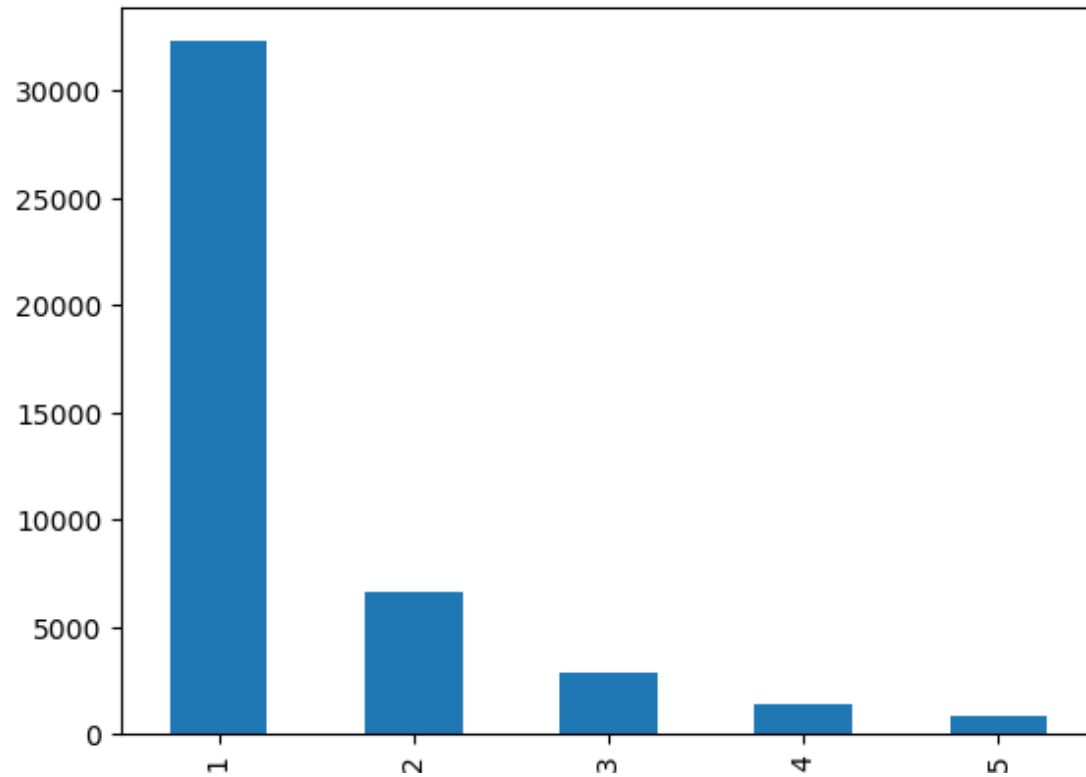
```
Out[30]: <Axes: >
```



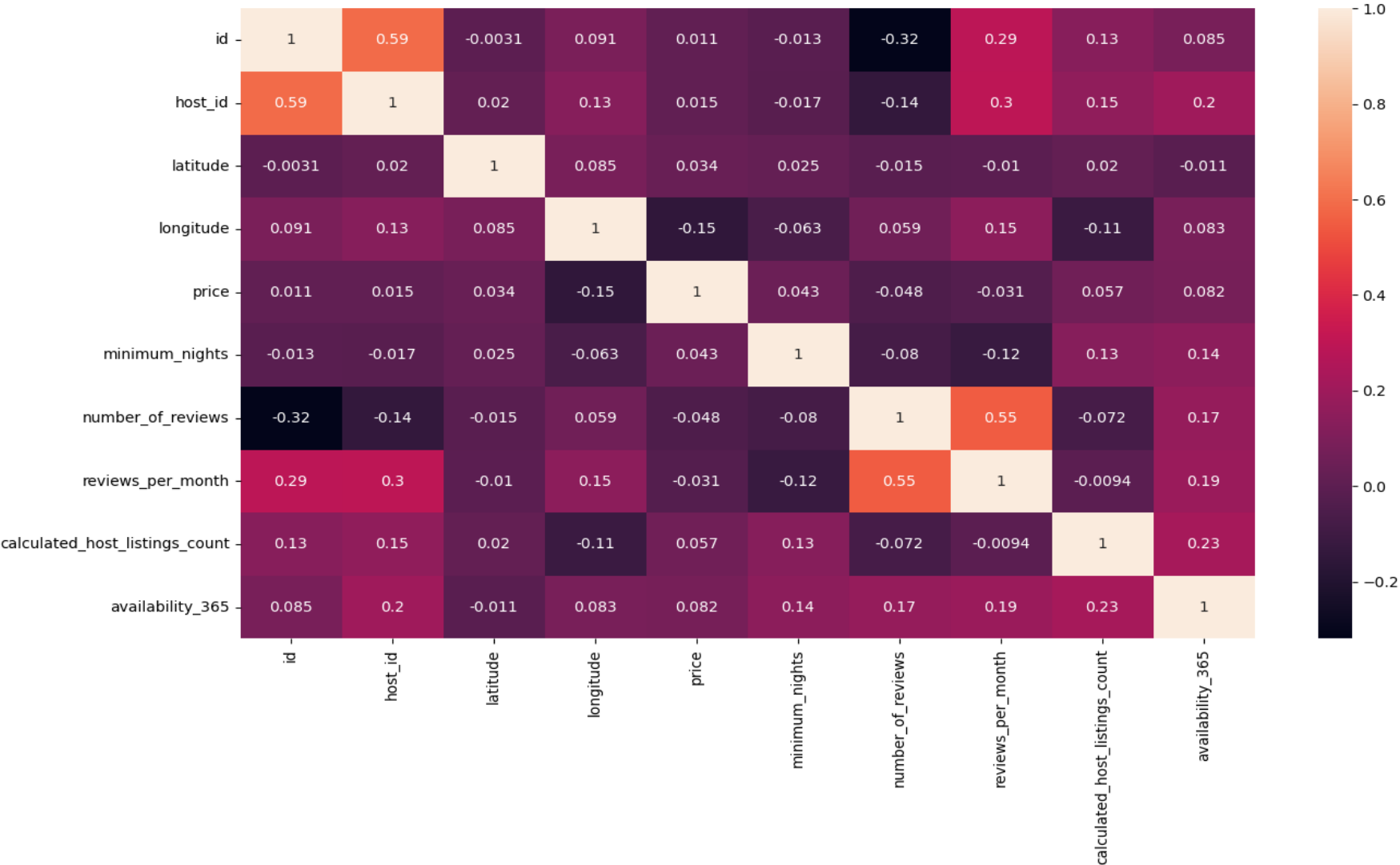
# Calculated Host Listing

```
In [42]: df.calculated_host_listings_count.value_counts().iloc[:5].plot(kind = 'bar')
```

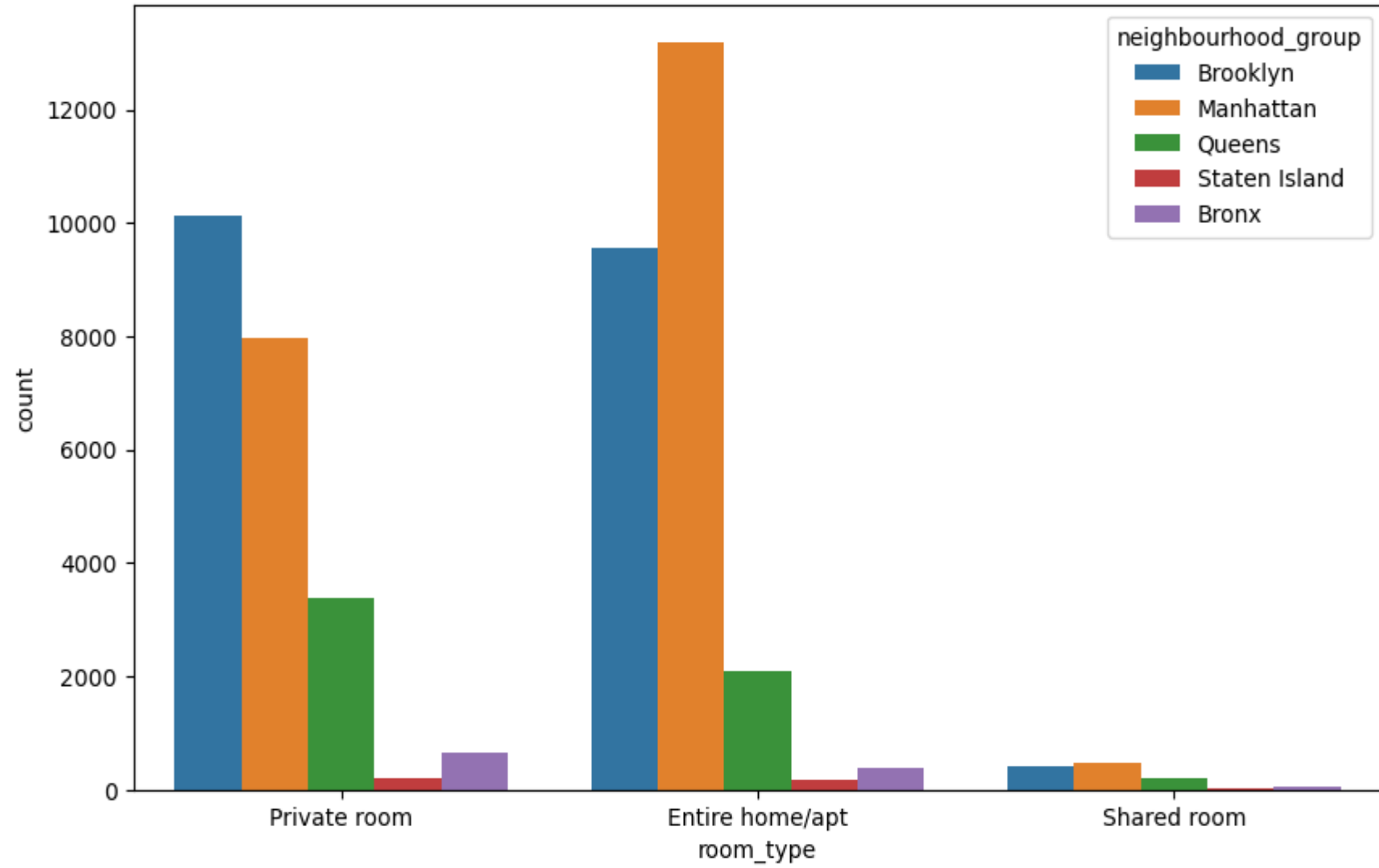
```
Out[42]: <Axes: >
```



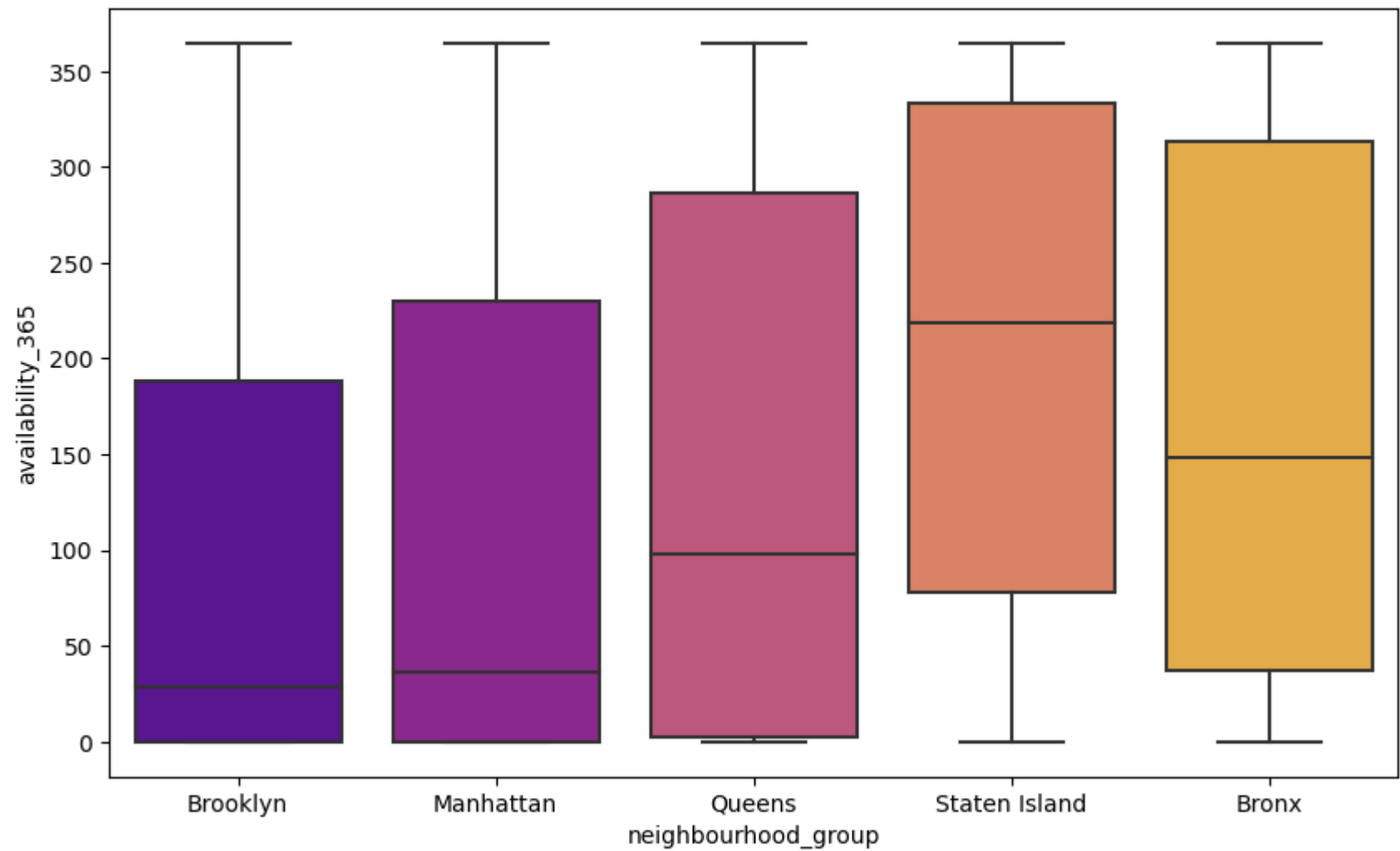
# Bivariate Analysis



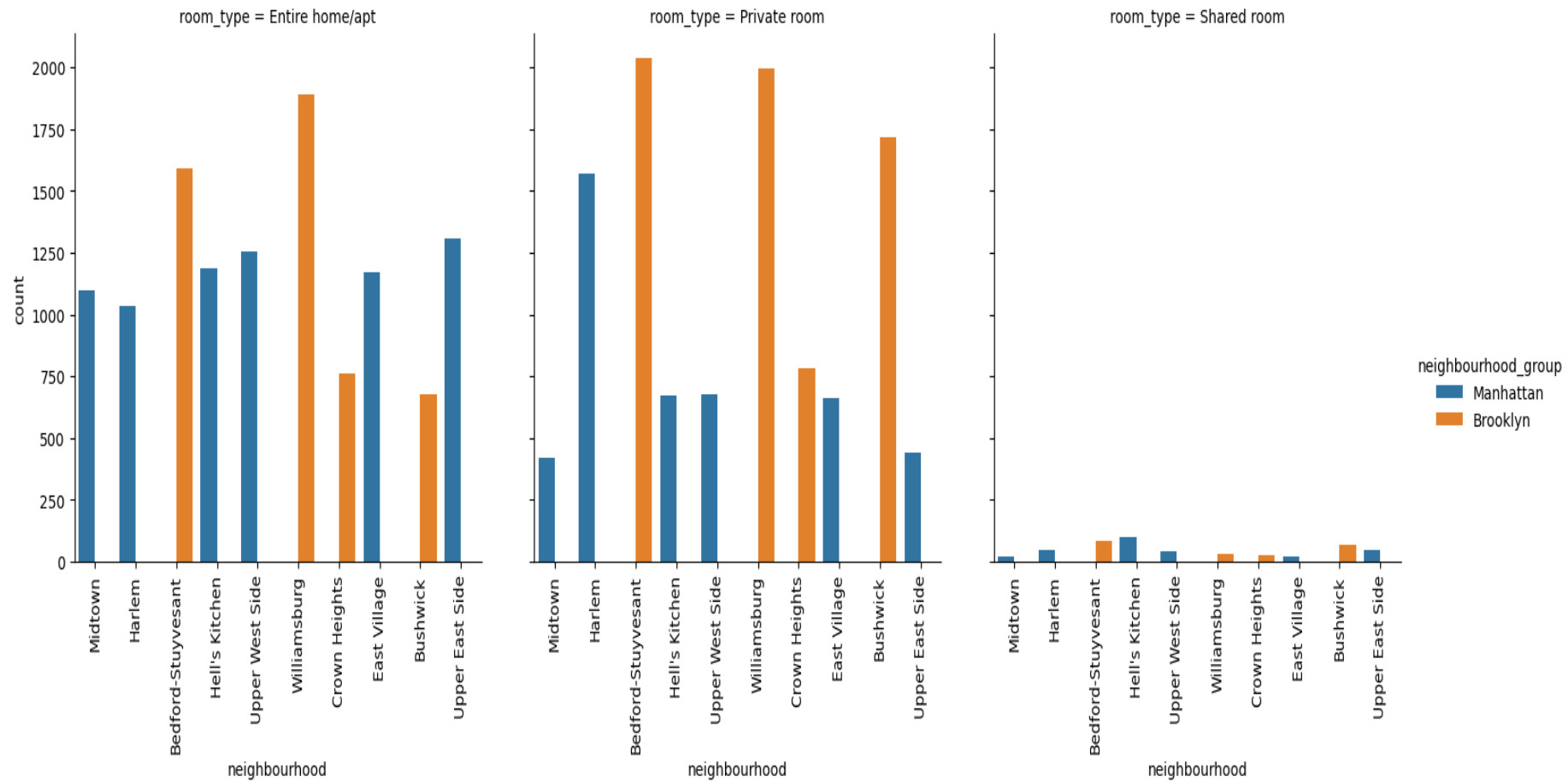
# Plots



*Availability\_365*



## Neighbourhood and Room\_type







***THANK  
YOU***