AIRBNB Case study

Methodology Document

Importing libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Reading data

```
In [2]: data = pd.read_csv("\downl\AB_NYC_2019.csv")
```

```
In [3]: data.head(5)
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_revie
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	2
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

Creating features

- categorizing the "availability_365" column into 5 categories

```
: def availability_365_categories_function(row):
    """
    Categorizes the "minimum_nights" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 100:
        return 'Low'
    elif row <= 200 :
        return 'Medium'
    elif (row <= 300):
        return 'High'
    else:
        return 'very High'
```

categorizing the "minimum_nights" column into 5 categories

```
]: def minimum_night_categories_function(row):
    """
    Categorizes the "minimum_nights" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 3:
        return 'Low'
    elif row <= 5 :
        return 'Medium'
    elif (row <= 7):
        return 'High'
    else:
        return 'very High'
```

categorizing the "number_of_reviews" column into 5 categories

```
]: def number_of_reviews_categories_function(row):
    """
    Categorizes the "number_of_reviews" column into 5 categories
    """
    if row <= 1:
        return 'very Low'
    elif row <= 5:
        return 'Low'
    elif row <= 10 :
        return 'Medium'
    elif (row <= 30):
        return 'High'
    else:
        return 'very High'
```

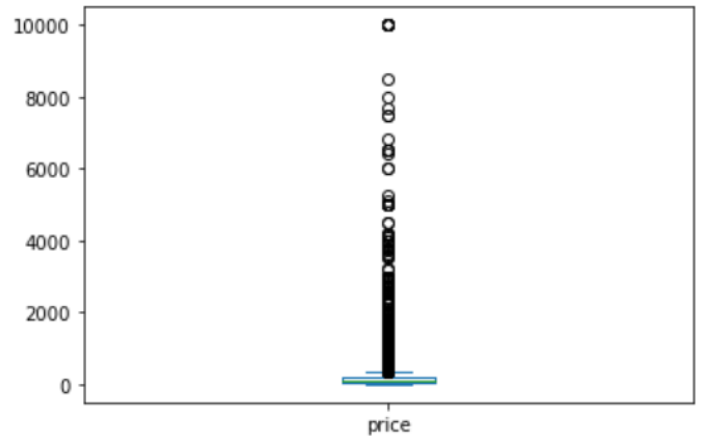
categorizing the "price" column into 5 categories

```
data.price.describe()
```

```
count    48895.000000
mean       152.720687
std        240.154170
min          0.000000
25%         69.000000
50%        106.000000
75%        175.000000
max      10000.000000
Name: price, dtype: float64
```

```
data.price.plot.box()
```

```
<AxesSubplot:>
```



Data types
Categorical

```
data.columns
```

```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'availability_365_categories',
       'minimum_night_categories', 'number_of_reviews_categories',
       'price_categories'],
      dtype='object')
```

```
# Categorical nominal
categorical_columns = data.columns[[0,1,3,4,5,8,16,17,18,19]]
categorical_columns
```

```
Index(['id', 'name', 'host_name', 'neighbourhood_group', 'neighbourhood',
       'room_type', 'availability_365_categories', 'minimum_night_categories',
       'number_of_reviews_categories', 'price_categories'],
      dtype='object')
```

Numerical

```
numerical_columns = data.columns[[9,10,11,13,14,15]]
numerical_columns
```

```
Index(['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

```
data[numerical_columns].describe()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000
mean	152.720687	7.029962	23.274466	1.090910	7.143982	112.781327
std	240.154170	20.510550	44.550582	1.597283	32.952519	131.622289
min	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
25%	69.000000	1.000000	1.000000	0.040000	1.000000	0.000000
50%	106.000000	3.000000	5.000000	0.370000	1.000000	45.000000
75%	175.000000	5.000000	24.000000	1.580000	2.000000	227.000000
max	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

Coordinates and date

```
coordinates = data.columns[[5,6,12]]
data[coordinates]
```

	neighbourhood	latitude	last_review
0	Kensington	40.64749	19-10-2018
1	Midtown	40.75362	21-05-2019
2	Harlem	40.80902	NaN
3	Clinton Hill	40.68514	05-07-2019
4	East Harlem	40.79851	19-11-2018
...	...	...	...
48890	Bedford-Stuyvesant	40.67853	NaN
48891	Bushwick	40.70184	NaN
48892	Harlem	40.81475	NaN
48893	Hell's Kitchen	40.75751	NaN
48894	Hell's Kitchen	40.76404	NaN

48895 rows × 3 columns

Univariate Analysis

- Name

```
data.name.value_counts()
```

```
Hillside Hotel                                        18
Home away from home                                   17
New york Multi-unit building                          16
Brooklyn Apartment                                    12
Loft Suite @ The Box House Hotel                      11
                                                      ..
Brownstone garden 2 bedroom duplex, Central Park       1
Bright Cozy Private Room near Columbia Univ            1
1 bdrm/large studio in a great location                1
Cozy Private Room #2 Two Beds Near JFK and J Train     1
Trendy duplex in the very heart of Hell's Kitchen      1
Name: name, Length: 47896, dtype: int64
```

Host_id

```
data.host_id.value_counts()
```

```
219517861    327
107434423    232
30283594     121
137358866    103
16098958      96
             ...
23727216       1
89211125       1
19928013       1
1017772        1
68119814       1
Name: host_id, Length: 37457, dtype: int64
```

Host_name

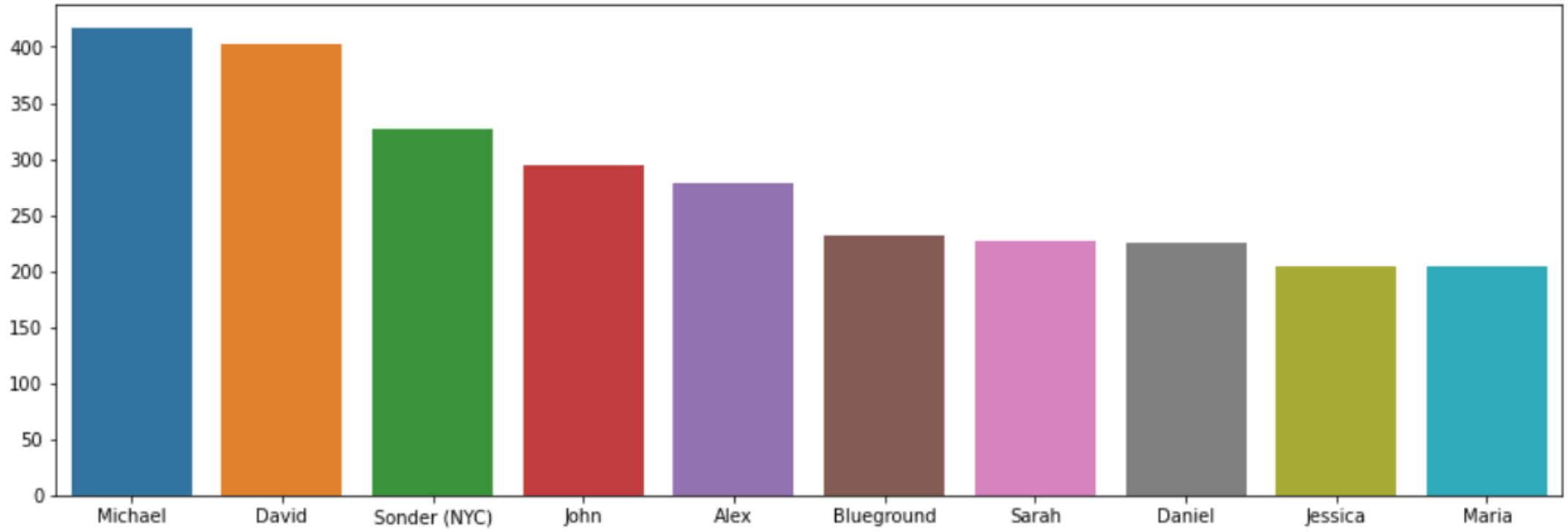
```
data.host_name.value_counts()
```

```
Michael              417
David                403
Sonder (NYC)         327
John                 294
Alex                 279
                    ...
Rhonycs                1
Brandy-Courtney        1
Shanthony              1
Aurore And Jamila      1
Ilgar & Aysel          1
Name: host_name, Length: 11452, dtype: int64
```

```
data.host_name.value_counts().index[:10]
```

```
Index(['Michael', 'David', 'Sonder (NYC)', 'John', 'Alex', 'Blueground',
       'Sarah', 'Daniel', 'Jessica', 'Maria'],
      dtype='object')
```

```
# Top 10 host's
plt.figure(figsize=(15,5))
sns.barplot(x = data.host_name.value_counts().index[:10] , y = data.host_name.value_counts().values[:10])
plt.show()
```



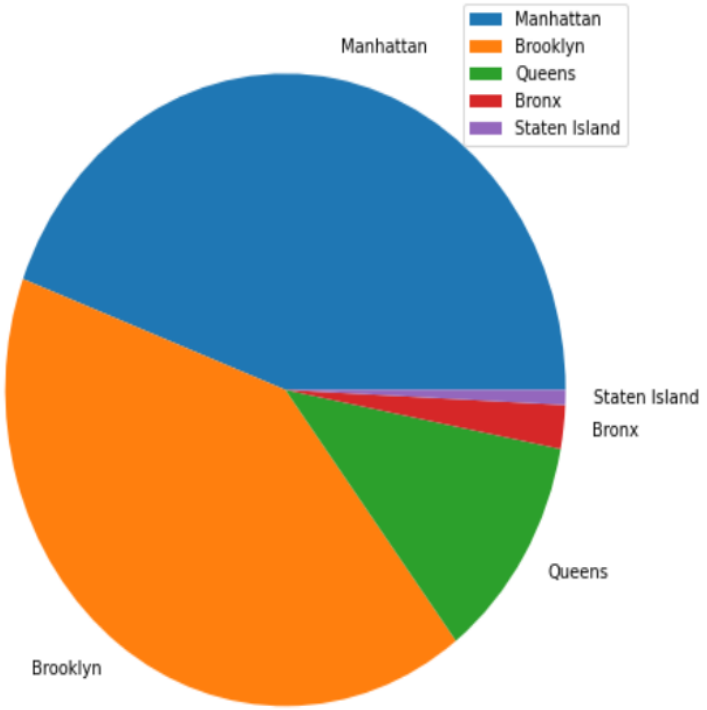
neighbourhood_group

What are the neighbourhoods they need to target? 81 % of the listing are Manhattan and Brooklyn neighbourhood_group

```
data.neighbourhood_group.value_counts()
```

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

```
plt.figure(figsize=(8,8))
plt.pie(x = data.neighbourhood_group.value_counts(normalize= True) * 100,labels = data.neighbourhood_group.value_counts(normalize
plt.legend()
plt.show()
```



neighbourhood

```
data.neighbourhood.value_counts()
```

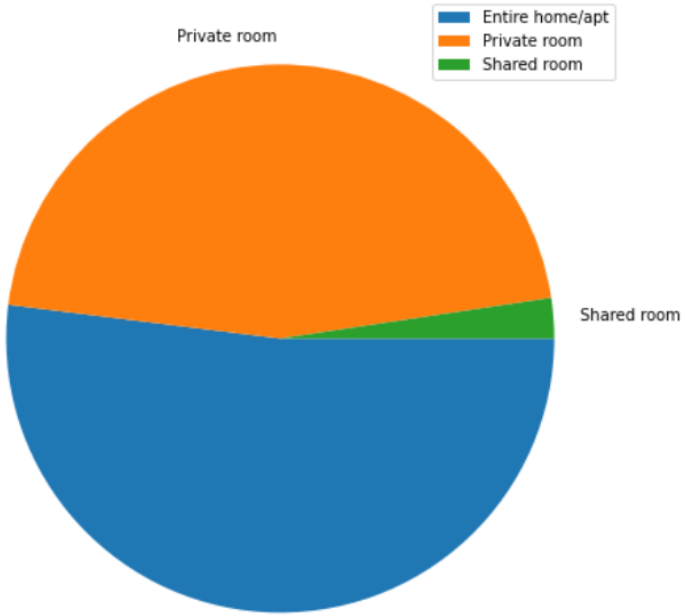
```
Williamsburg            3920
Bedford-Stuyvesant      3714
Harlem                  2658
Bushwick                2465
Upper West Side         1971
                        ...
Fort Wadsworth             1
Richmondtown               1
New Dorp                   1
Rossville                  1
Willowbrook                1
Name: neighbourhood, Length: 221, dtype: int64
```

Room_type

```
data.room_type.value_counts()
```

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

```
plt.figure(figsize=(8,8))
plt.pie(x = data.room_type.value_counts(normalize= True) * 100,labels = data.room_type.value_counts(normalize= True).index,counte
plt.legend()
plt.show()
```



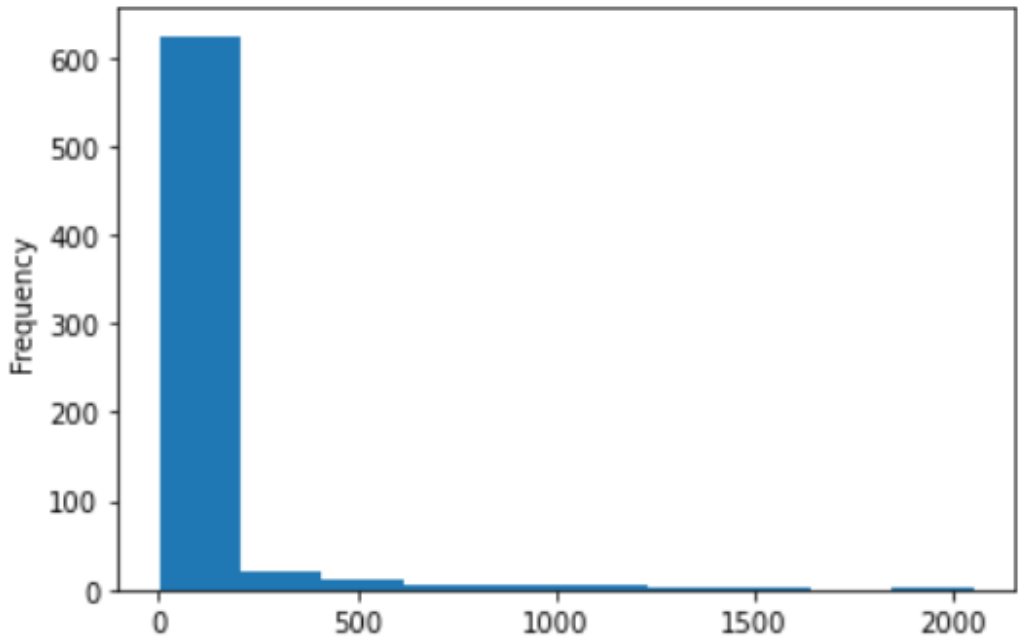
price

```
data.price.value_counts()
```

```
100    2051
150    2047
50     1534
60     1458
200    1401
       ...
780       1
386       1
888       1
483       1
338       1
Name: price, Length: 674, dtype: int64
```

```
data.price.value_counts().plot.hist()
```

```
<AxesSubplot:ylabel='Frequency'>
```



minimum_nights

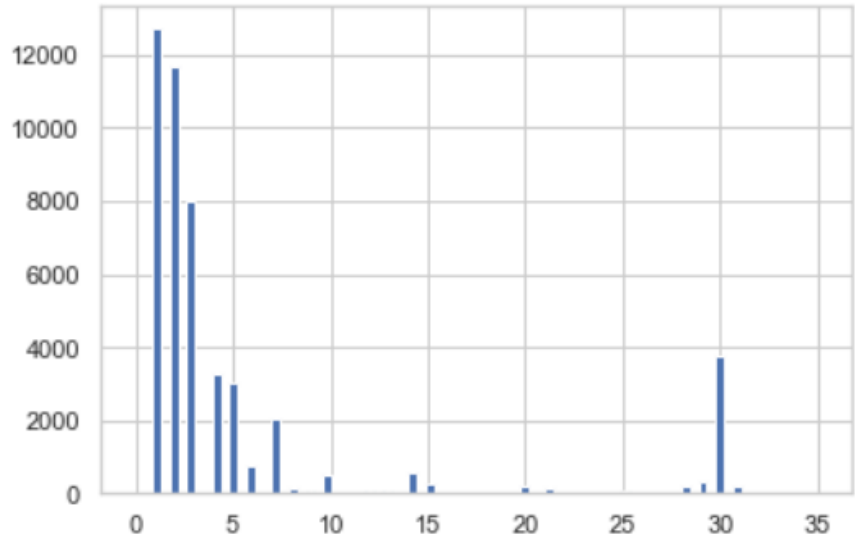
```
data.minimum_nights.value_counts()
```

```
1      12720
2      11696
3       7999
30      3760
4       3303
       ...
186        1
366        1
68         1
87         1
36         1
Name: minimum_nights, Length: 109, dtype: int64
```

```
data.minimum_nights.describe()
```

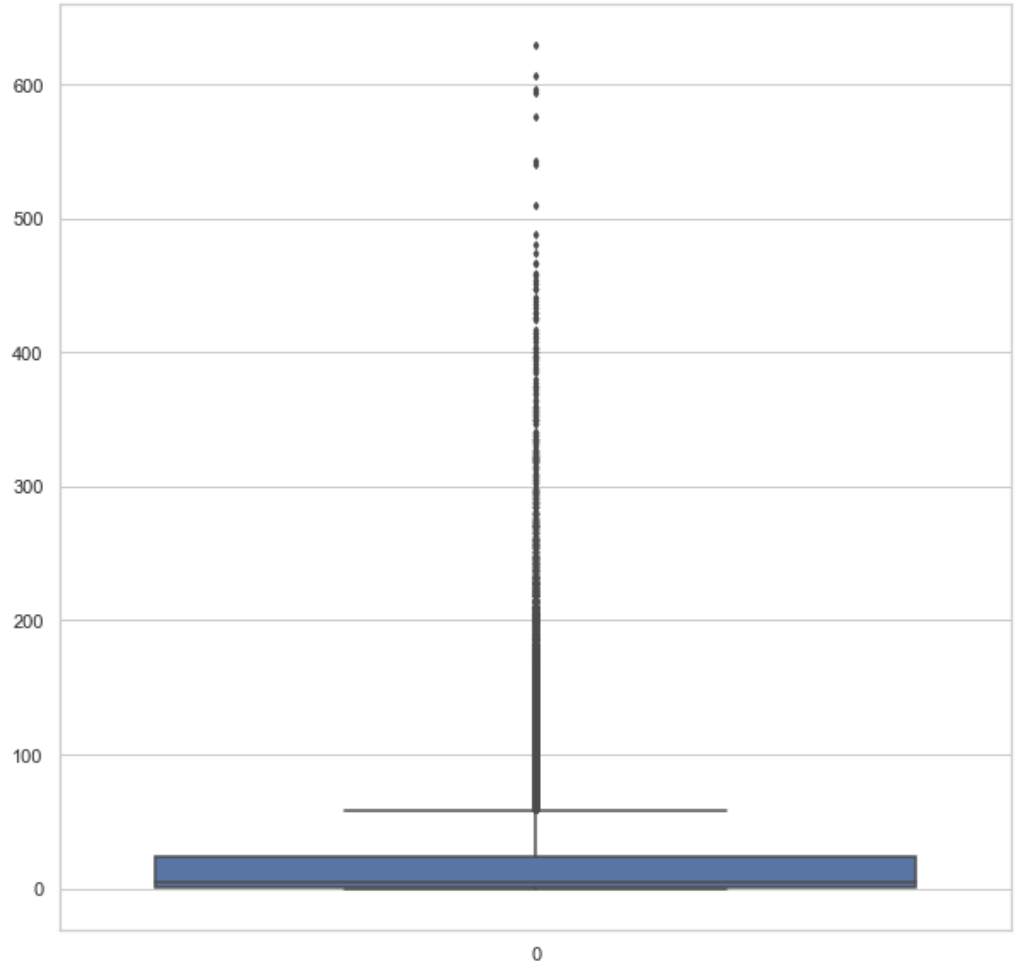
```
count    48895.000000
mean         7.029962
std         20.510550
min          1.000000
25%          1.000000
50%          3.000000
75%          5.000000
max       1250.000000
Name: minimum_nights, dtype: float64
```

```
: plt.hist(data = data, x = 'minimum_nights',bins=80,range=(0,35))
  plt.show()
```

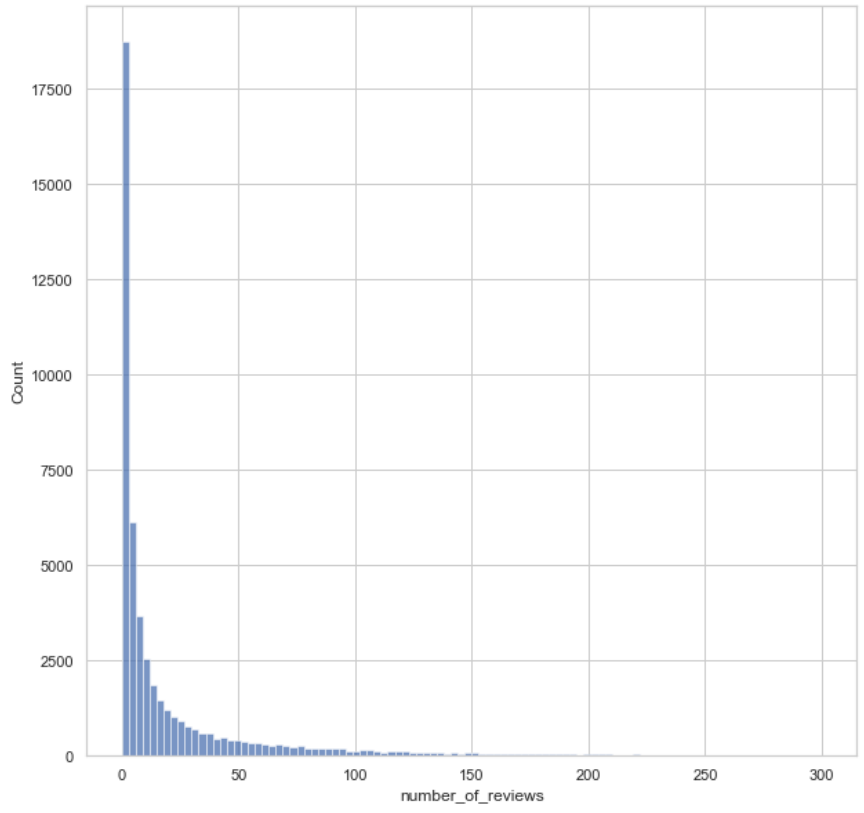


number_of_reviews

```
plt.figure(figsize=(10,10))
sns.boxplot(data = data.number_of_reviews,fliersize=3)
plt.show()
```

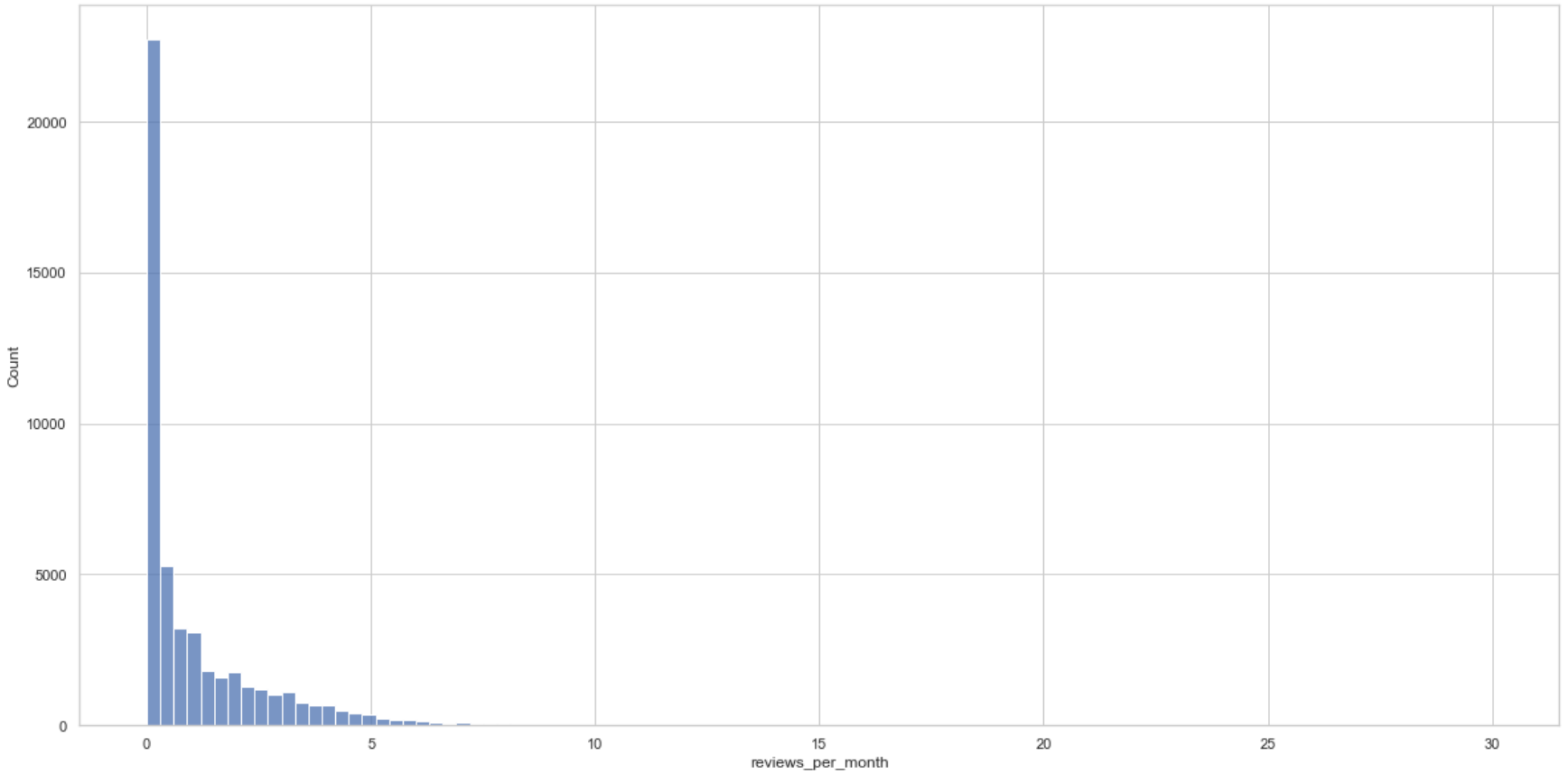


```
plt.figure(figsize = (10,10))
sns.histplot(data = data, x = 'number_of_reviews',bins=100,binrange=(0,300))
plt.show()
```



reviews_per_month

```
plt.figure(figsize = (20,10))
sns.histplot(data = data, x = 'reviews_per_month',bins=100,binrange=(0,30))
plt.show()
```

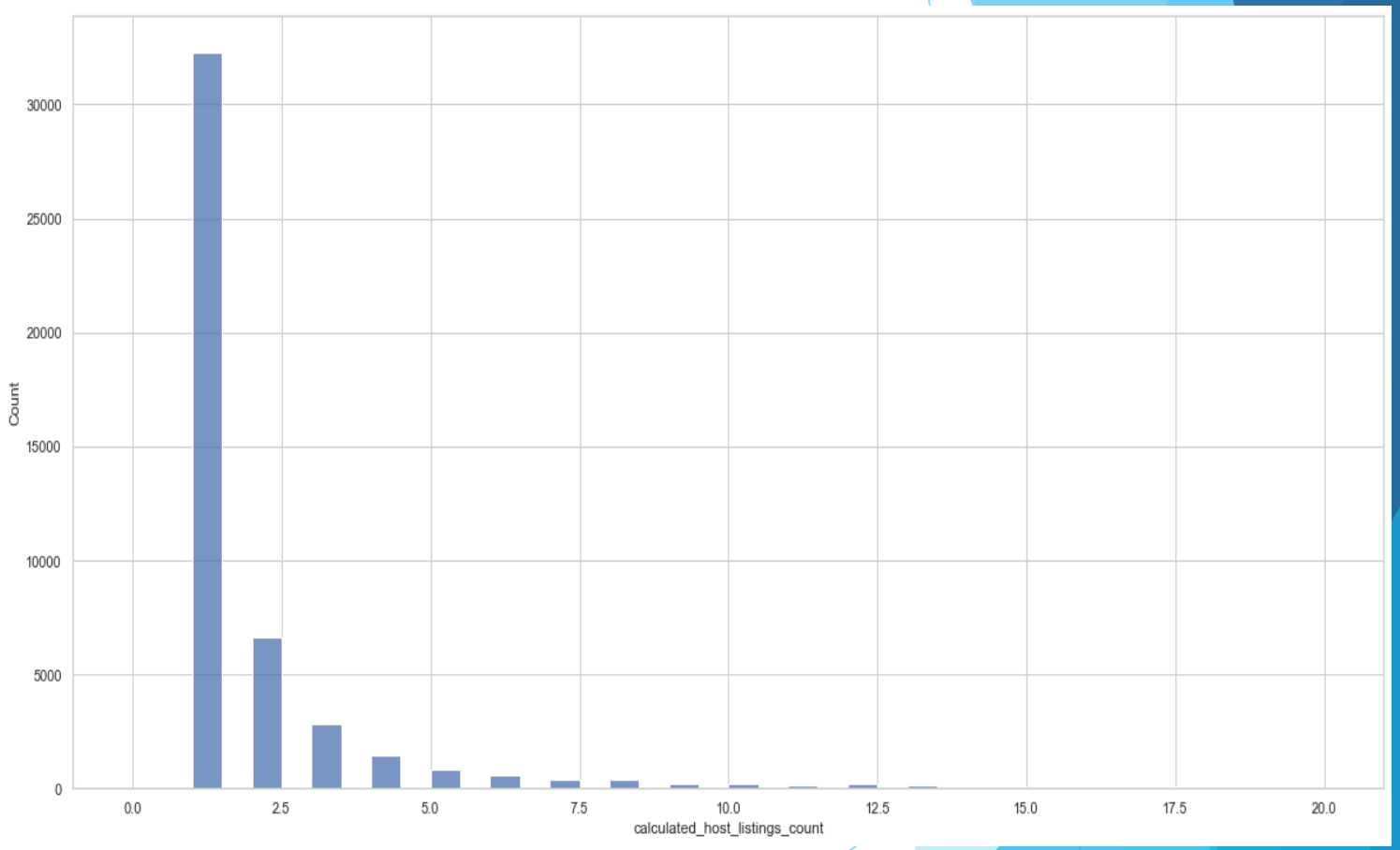


calculated_host_listings_count

```
data.calculated_host_listings_count.describe()
```

```
count    48895.000000
mean         7.143982
std         32.952519
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max        327.000000
Name: calculated_host_listings_count, dtype: float64
```

```
plt.figure(figsize = (20,10))
sns.histplot(data = data, x = 'calculated_host_listings_count',bins=40,binrange=(0,20))
plt.show()
```

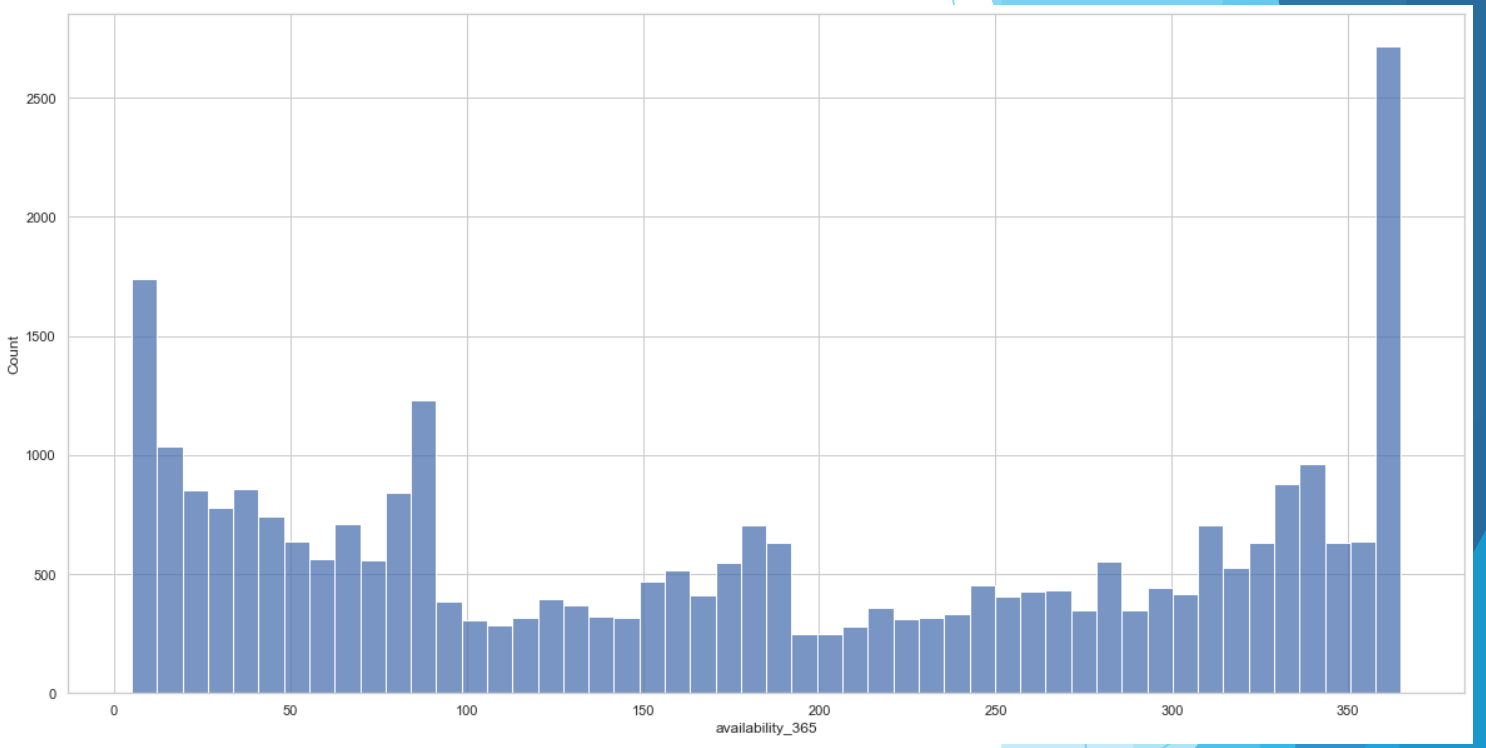


availability_365

```
data.availability_365.describe()
```

```
count    48895.000000
mean       112.781327
std        131.622289
min          0.000000
25%          0.000000
50%         45.000000
75%        227.000000
max        365.000000
Name: availability_365, dtype: float64
```

```
plt.figure(figsize = (20,10))
sns.histplot(data = data, x = 'availability_365',bins=50,binrange=(5,365))
plt.show()
```

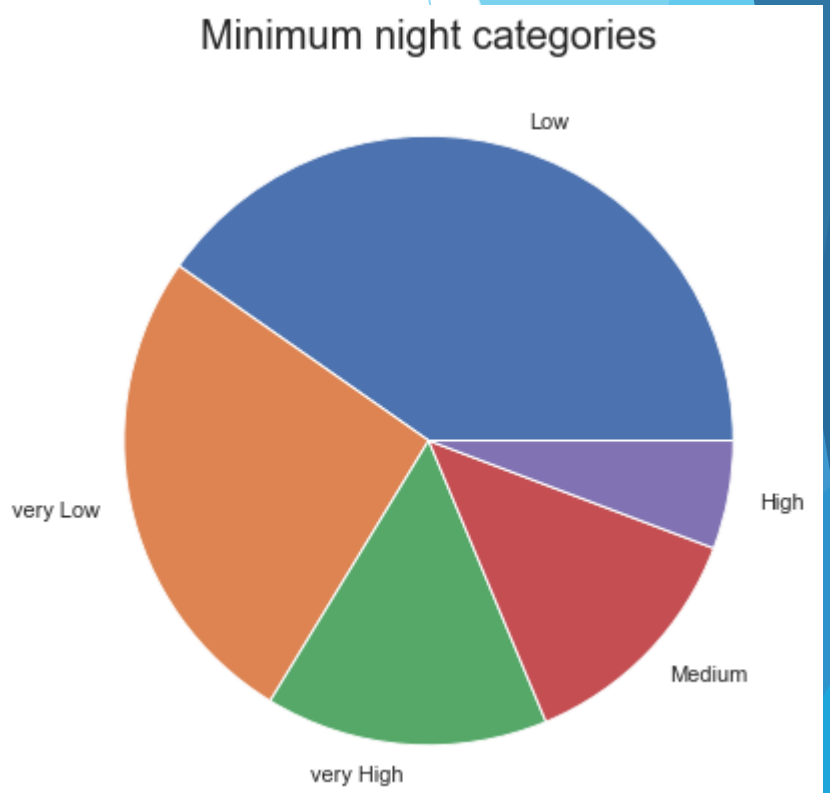


minimum_night_categories

```
data.minimum_night_categories.value_counts(normalize= True)*100
```

```
Low          40.280192
very Low     26.014930
very High    14.997444
Medium       12.960425
High          5.747009
Name: minimum_night_categories, dtype: float64
```

```
plt.figure(figsize=(12,7))
plt.title('Minimum night categories', fontdict={'fontsize': 20})
plt.pie(x = data.minimum_night_categories.value_counts(),labels=data.minimum_night_categories.value_counts().index)
plt.show()
```

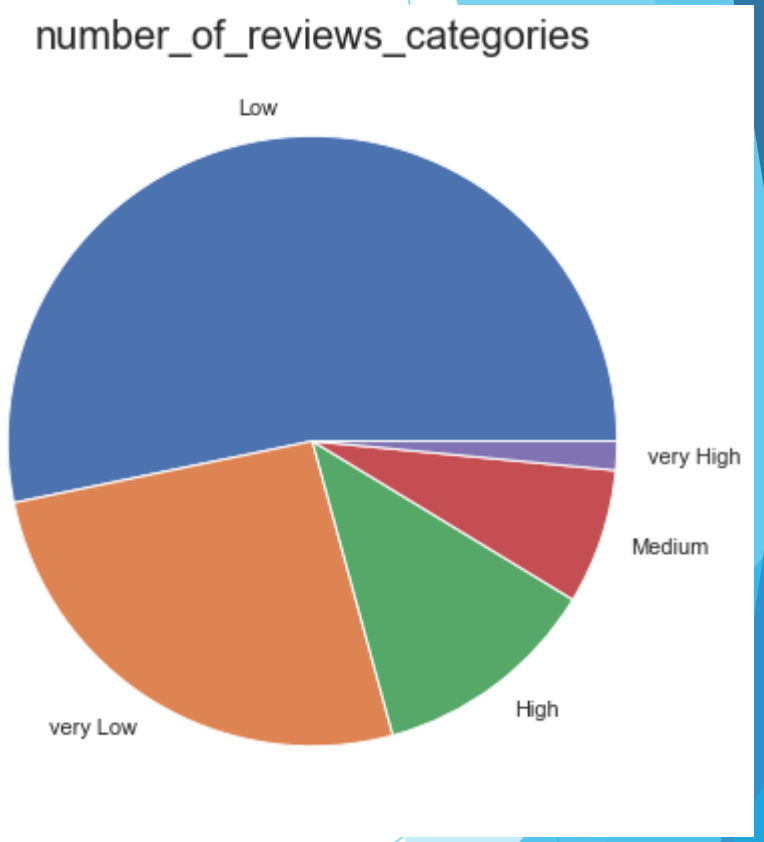


number_of_reviews_categories

```
data.number_of_reviews_categories.value_counts()
```

```
Low          26032
very Low     12720
High          5893
Medium        3503
very High      747
Name: number_of_reviews_categories, dtype: int64
```

```
plt.figure(figsize=(12,7))
plt.title('number_of_reviews_categories', fontdict={'fontsize': 20})
plt.pie(x = data.number_of_reviews_categories.value_counts(),labels=data.number_of_reviews_categories.value_counts().index)
plt.show()
```



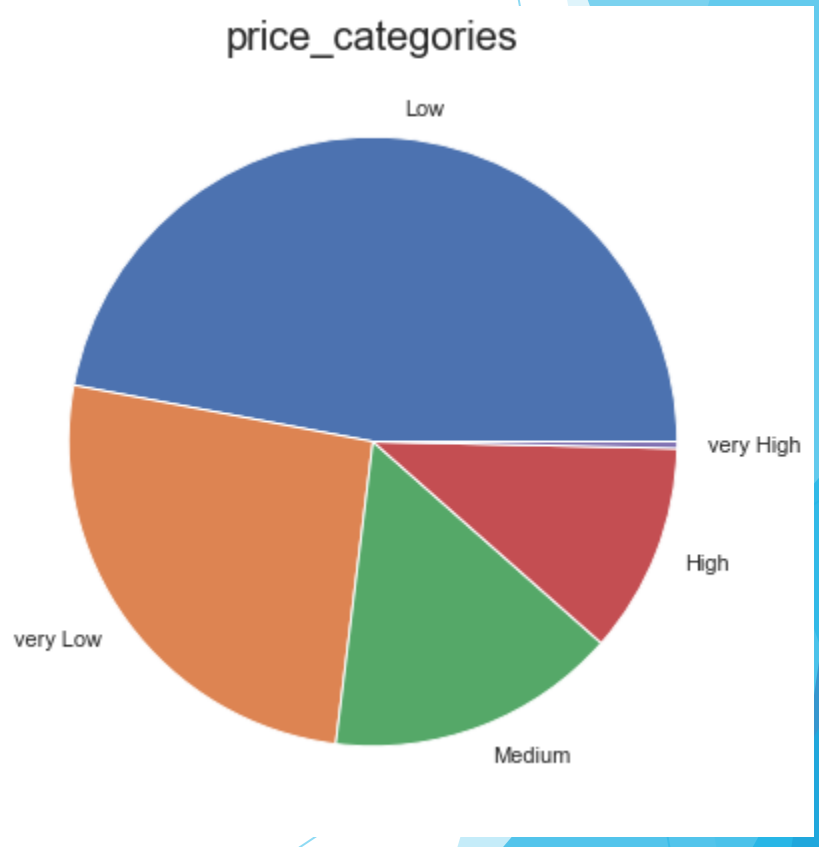
Price_categories

What is the pricing ranges preferred by customers? 'Low' price ranges are preferred by customers followed by very 'Low' price ranges.

```
data['price_categories'].value_counts()
```

```
Low          22998
very Low     12720
Medium        7556
High          5447
very High      174
Name: price_categories, dtype: int64
```

```
plt.figure(figsize=(12,7))
plt.title('price_categories', fontdict={'fontsize': 20})
plt.pie(x = data.price_categories.value_counts(),labels=data.price_categories.value_counts().index,)
plt.show()
```

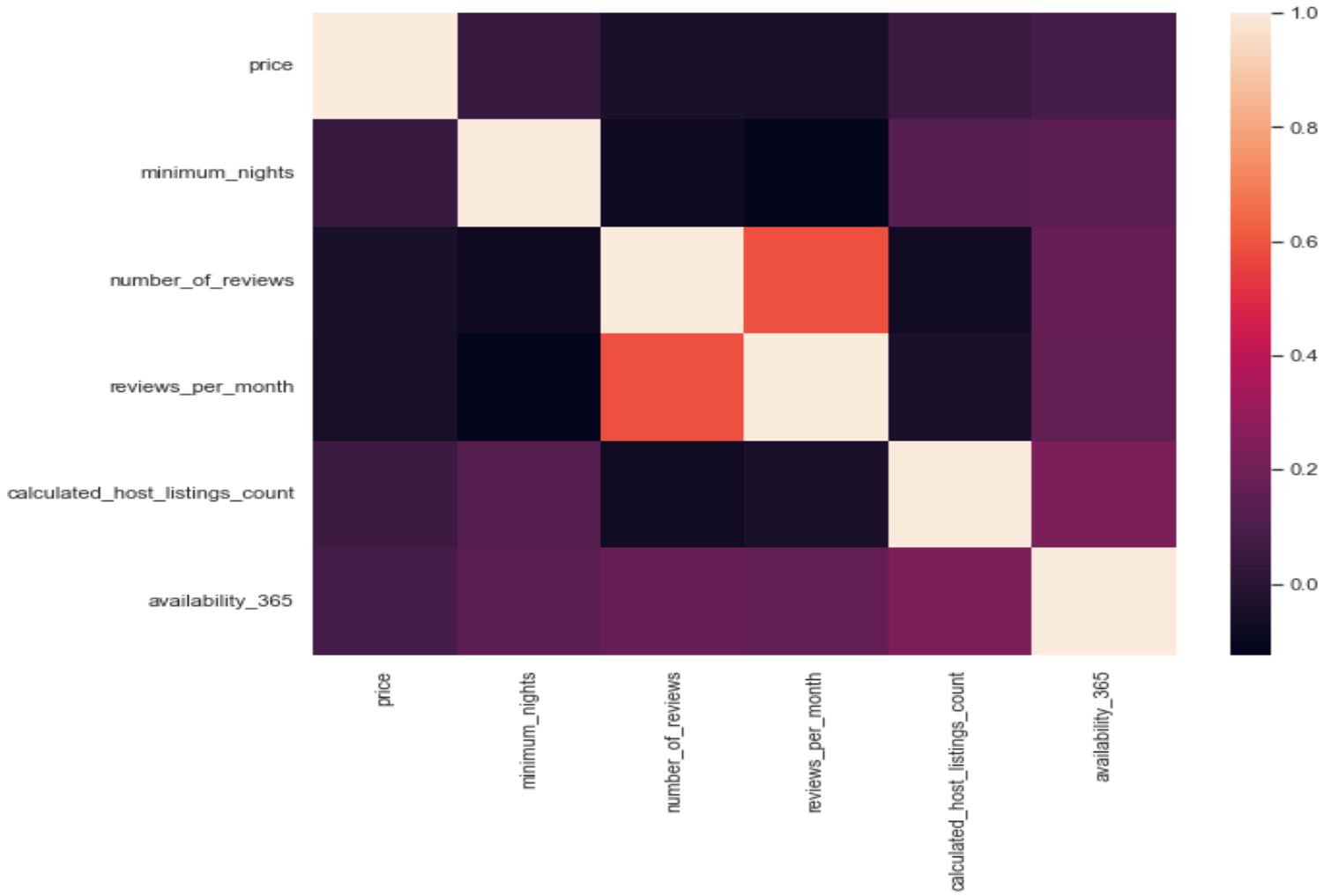


Bivariate and Multivariate Analysis
Finding the correlations

```
data[numerical_columns].corr()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
price	1.000000	0.042799	-0.047954	-0.050564	0.057472	0.081829
minimum_nights	0.042799	1.000000	-0.080116	-0.124905	0.127960	0.144303
number_of_reviews	-0.047954	-0.080116	1.000000	0.589407	-0.072376	0.172028
reviews_per_month	-0.050564	-0.124905	0.589407	1.000000	-0.047312	0.163732
calculated_host_listings_count	0.057472	0.127960	-0.072376	-0.047312	1.000000	0.225701
availability_365	0.081829	0.144303	0.172028	0.163732	0.225701	1.000000

```
plt.figure(figsize=(10,8))
sns.heatmap(data = data[numerical_columns].corr())
plt.show()
```



Finding Top correlations

```
corr_matrix = data[numerical_columns].corr().abs()

#the matrix is symmetric so we need to extract upper triangle matrix without diagonal (k = 1)

sol = (corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
                  .stack()
                  .sort_values(ascending=False))
```

```
corr_matrix
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
price	1.000000	0.042799	0.047954	0.050564	0.057472	0.081829
minimum_nights	0.042799	1.000000	0.080116	0.124905	0.127960	0.144303
number_of_reviews	0.047954	0.080116	1.000000	0.589407	0.072376	0.172028
reviews_per_month	0.050564	0.124905	0.589407	1.000000	0.047312	0.163732
calculated_host_listings_count	0.057472	0.127960	0.072376	0.047312	1.000000	0.225701
availability_365	0.081829	0.144303	0.172028	0.163732	0.225701	1.000000

```
sol[1:8]
```

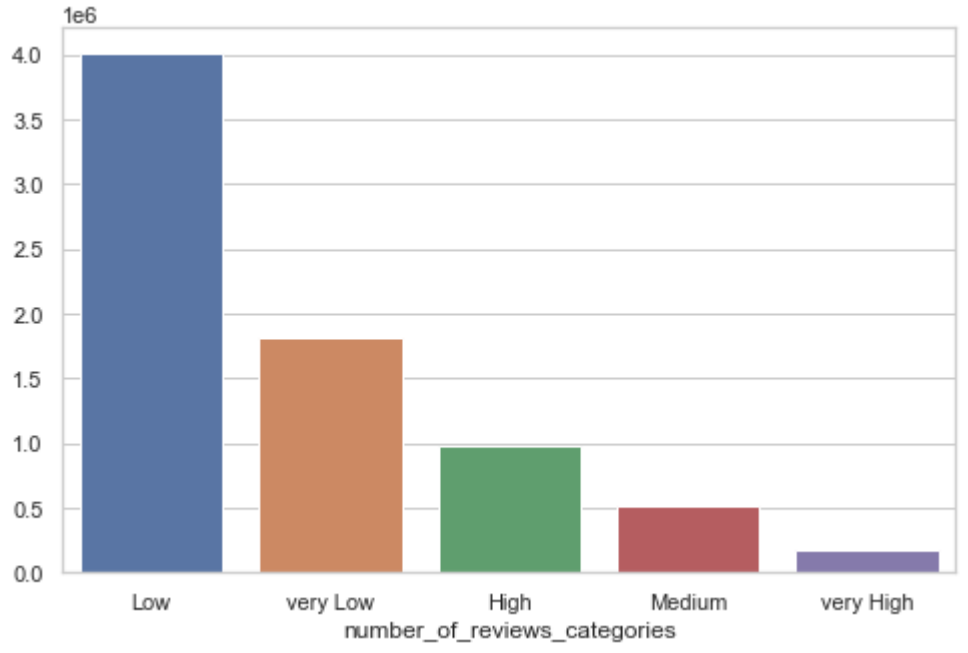
```
calculated_host_listings_count  availability_365                0.225701
number_of_reviews               availability_365                0.172028
reviews_per_month               availability_365                0.163732
minimum_nights                  availability_365                0.144303
                                calculated_host_listings_count  0.127960
                                reviews_per_month               0.124905
price                           availability_365                0.081829
dtype: float64
```

number_of_reviews_categories and prices
What is the pricing ranges preferred by customers? The total price for 'Low' or 'very Low' number_of_reviews_categories are high.

```
# prices for each of reviews_categories
x1 = data.groupby('number_of_reviews_categories').price.sum().sort_values(ascending = False)
x1
```

```
number_of_reviews_categories
Low          4002323
very Low     1806531
High          971346
Medium        508647
very High     178431
Name: price, dtype: int64
```

```
plt.figure(figsize=(8,5))
sns.barplot(x = x1.index,y = x1.values)
plt.show()
```



('room_type' and 'number_of_reviews_categories')

- The various kinds of properties that exist w.r.t. customer preferences.? Entire home/apt have more reviews than Shared rooms 'Shared room' are less likely to give reviews. only 16 %

```
data.room_type.value_counts()
```

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

```
pd.crosstab(data['room_type'], data['number_of_reviews_categories'])
```

number_of_reviews_categories room_type	High	Low	Medium	very High	very Low
Entire home/apt	3809	14909	1960	504	4227
Private room	1950	10769	1494	226	7887
Shared room	134	354	49	17	606

room_type' and 'price_categories

```
pd.crosstab(data['room_type'], data['number_of_reviews_categories'])
```

number_of_reviews_categories	High	Low	Medium	very High	very Low
room_type					
Entire home/apt	3809	14909	1960	504	4227
Private room	1950	10769	1494	226	7887
Shared room	134	354	49	17	606

'room_type' and ‘reviews_per_month’
For each 'room_type' there are ~1.4 reviews per month on average.

```
: data.room_type.value_counts()

: Entire home/apt    25409
  Private room       22326
  Shared room         1160
  Name: room_type, dtype: int64

: data.groupby('room_type').reviews_per_month.mean()

: room_type
  Entire home/apt    1.045509
  Private room       1.143493
  Shared room        1.073345
  Name: reviews_per_month, dtype: float64

: data.groupby('room_type').reviews_per_month.median()

: room_type
  Entire home/apt    0.350
  Private room       0.400
  Shared room        0.405
  Name: reviews_per_month, dtype: float64
```

7 minimum_night_categories and reviews_per_month

- Customers are more likely to leave reviews for low number of minimum nights Adjustments in the existing properties to make it more customer-oriented. ? minimum_nights should be on the lower side to make properties more customer-oriented

```
data.groupby('minimum_night_categories').reviews_per_month.sum().sort_values()
```

```
minimum_night_categories
High          1227.57
very High     2235.19
Medium        4689.73
very Low     20395.49
Low          24792.06
Name: reviews_per_month, dtype: float64
```

availability_365_categories', 'price_categories' and 'reviews_per_month'

- If the combination of availability and price is very high, reviews_per_month will be low on average. Very high availability and very low price are likely to get more reviews.

```
data.availability_365_categories.value_counts()
```

```
very Low     17941
Low          11829
very High     8108
Medium        5792
High          5225
Name: availability_365_categories, dtype: int64
```

very High	**High**	0.225445
	Low	1.309856
	Medium	0.560150
	very High	0.124103
	very Low	1.801516
very Low	**High**	0.205953
	Low	0.407902
	Medium	0.186864
	very High	0.255312
	very Low	0.439738

		reviews_per_month
availability_365_categories	**price_categories**	
High	**High**	0.413506
	Low	2.095180
	Medium	0.950500
	very High	0.211905
	very Low	2.986492
Low	**High**	0.407565
	Low	1.583401
	Medium	0.700449
	very High	0.612381
	very Low	2.515795
Medium	**High**	0.401201
	Low	1.797536
	Medium	0.971300
	very High	0.188182
	very Low	2.532178