REVEALING THE SECRETS OF AIRBNB IN NYC: DATA METHODOLOGY

1. Importing Libraries and reading the data:

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

```
In [2]: airbnb = pd.read_csv("AB_NYC_2019.csv")
        airbnb.head(10)
```

Out[2]:

	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	[illegible]
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	

2. Binning the Categorical Variables into Groups:

- **2.1 Categorization of the "price" column into 5 groups**

```
In [10]: def price_categorization(row):

             if row <= 50:
                 return 'very Low'
             elif row <= 100:
                 return 'Low'
             elif row <= 200:
                 return 'Medium'
             elif (row <= 250):
                 return 'High'
             else:
                 return 'very High'
```

- **2.2 Categorization of the "minimum_nights" column into 5 groups**

In [15]:
```
def minimum_night_categorization(row):

    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'
```

- **2.3 Categorization of the "number_of_reviews" column into 5 groups**

In [20]:
```
def number_of_reviews_categorization(row):

    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'
```

- **2.4 Categorization of the "availability_365" column into 5 groups**

In [25]:
```
def availability_365_categorization(row):

    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'
```

3. Fixing columns and reviewing data types

```
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 20 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
 16  price_categories                48895 non-null  object
 17  minimum_night_categories        48895 non-null  object
 18  number_of_reviews_categories    48895 non-null  object
 19  availability_365_categories     48895 non-null  object
dtypes: float64(3), int64(7), object(10)
memory usage: 7.5+ MB
```

```
airbnb.last_review = pd.to_datetime(airbnb.last_review)
airbnb.last_review
```

```
0       2018-10-19
1       2019-05-21
2              NaT
3       2019-07-05
4       2018-11-19
           ...
48890          NaT
48891          NaT
48892          NaT
48893          NaT
48894          NaT
Name: last_review, Length: 48895, dtype: datetime64[ns]
```

4. Variable Categories

- ## 4.1 Numerical variables

```
num_cols = airbnb.columns[[9,10,11,13,14,15]]
num_cols
```

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

```
airbnb[num_cols].describe()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	48895.000000	48895.000000	48895.000000	38843.000000	48895.000000	48895.000000
mean	152.720687	7.029962	23.274466	1.373221	7.143982	112.781327
std	240.154170	20.510550	44.550582	1.680442	32.952519	131.622289
min	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	69.000000	1.000000	1.000000	0.190000	1.000000	0.000000
50%	106.000000	3.000000	5.000000	0.720000	1.000000	45.000000
75%	175.000000	5.000000	24.000000	2.020000	2.000000	227.000000
max	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

- ## 4.2 Categorical variable

```
cat_cols = airbnb.columns[[0,1,3,4,5,8,16,17,18,19]]
cat_cols
```

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

```
airbnb[cat_cols].head(3)
```

	id	name	host_name	neighbourhood_group	neighbourhood	room_type	price_categories	minimum_night_categories	number_of_reviews_catego
0	2539	Clean & quiet apt home by the park	John	Brooklyn	Kensington	Private room	Medium	very Low	Med
1	2595	Skylit Midtown Castle	Jennifer	Manhattan	Midtown	Entire home/apt	High	very Low	very H
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	Elisabeth	Manhattan	Harlem	Private room	Medium	Low	very l

- ## 4.3 Location variable

```
location = airbnb.columns[[5,6,7]]
airbnb[location].head()
```

	neighbourhood	latitude	longitude
0	Kensington	40.64749	-73.97237
1	Midtown	40.75362	-73.98377
2	Harlem	40.80902	-73.94190
3	Clinton Hill	40.68514	-73.95976
4	East Harlem	40.79851	-73.94399

- **4.4 Date variable**

```
Date =airbnb.columns[[12]]
airbnb[Date].head()
```

	last_review
0	2018-10-19
1	2019-05-21
2	NaT
3	2019-07-05
4	2018-11-19

5. Missing values

```
missing_perc=round(airbnb.isna().sum()/len(airbnb)*100,2
missing_perc.sort_values(ascending=False)
```

```
reviews_per_month                 20.56
last_review                       20.56
host_name                          0.04
name                               0.03
id                                 0.00
number_of_reviews                  0.00
number_of_reviews_categories       0.00
minimum_night_categories           0.00
price_categories                   0.00
availability_365                   0.00
calculated_host_listings_count     0.00
minimum_nights                     0.00
price                              0.00
room_type                          0.00
longitude                          0.00
latitude                           0.00
neighbourhood                      0.00
neighbourhood_group                0.00
host_id                            0.00
availability_365_categories        0.00
dtype: float64
```

Observations :

The dataset contains two columns, 'last_review' and 'reviews_per_month,' which exhibit approximately 20.56% missing values. Additionally, the 'name' and 'host_name' columns have 0.3% and 0.4% missing values, respectively.

Our objective is to determine whether these missing values are MCAR (Missing Completely at Random) or MNAR (Missing Not at Random). The former implies that the absence of data is not related to any other features, while the latter indicates a specific reason behind the missing data

It is imperative to highlight that we will neither drop nor impute any columns, as our primary focus is on analyzing the dataset rather than constructing a model. Additionally, the majority of the features hold significant importance for our analysis.

- **5.1 Missing values Analysis - last_review**

```
# Selecting the data with missing values for 'last_review' Attribute
airbnb_1 = airbnb.loc[airbnb.last_review.isnull(),:]
airbnb_1.head(3)
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_revie
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
19	7750	Huge 2 BR Upper East Cental Park	17985	Sing	Manhattan	East Harlem	40.79685	-73.94872	Entire home/apt	190	7	
26	8700	Magnifique Suite au N de Manhattan - vue Cloitres	26394	Claude & Sophie	Manhattan	Inwood	40.86754	-73.92639	Private room	80	4	

- **5.2 Missing values Analysis ('last_review' vs 'neighbourhood_group')**

```
# identifying the Count of'neighbourhood_group' with missing values using newly created dataframe
airbnb_1.groupby('neighbourhood_group').neighbourhood_group.count()
```

```
neighbourhood_group
Bronx             215
Brooklyn         3657
Manhattan        5029
Queens           1092
Staten Island      59
Name: neighbourhood_group, dtype: int64
```

```
# identifying the Count of 'neighbourhood_group' using original dataframe
airbnb.groupby('neighbourhood_group').neighbourhood_group.count()
```

```
neighbourhood_group
Bronx             1091
Brooklyn         20104
Manhattan        21661
Queens            5666
Staten Island      373
```

```
(airbnb_1.groupby('neighbourhood_group').neighbourhood_group.count()/airbnb.groupby('neighbourhood_group').neighbourhood_group.co
```

```
neighbourhood_group
Bronx            19.706691
Brooklyn         18.190410
Manhattan        23.216841
Queens           19.272856
Staten Island    15.817694
Name: neighbourhood_group, dtype: float64
```

```
((airbnb_1.groupby('neighbourhood_group').neighbourhood_group.count()/airbnb.groupby('neighbourhood_group').neighbourhood_group.c
```

```
19.240898461107257
```

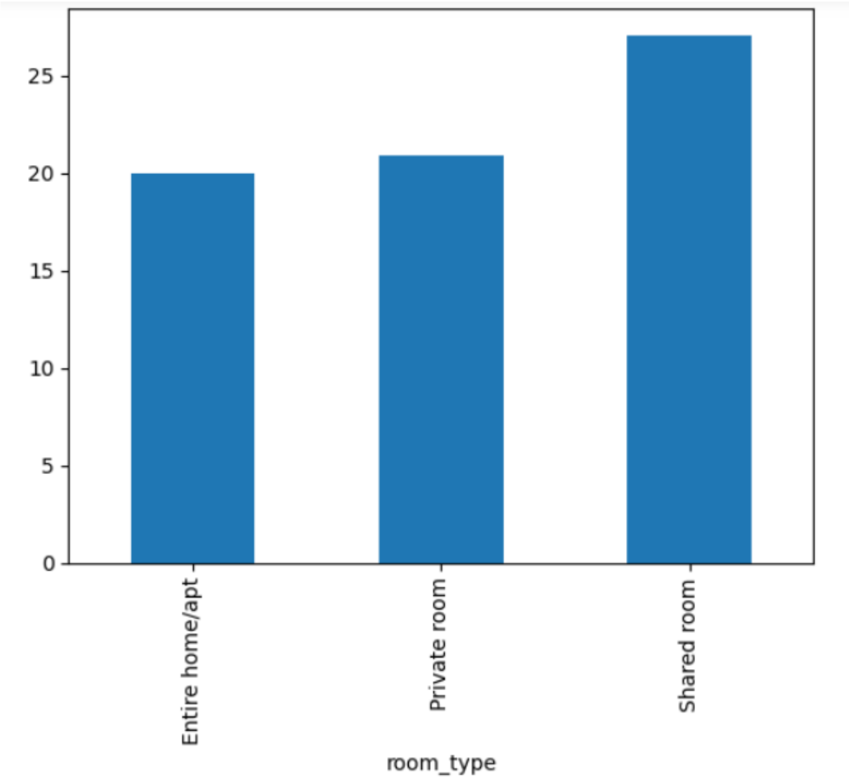
- We found that the 'neighbourhood_group' has approximately 19.2 % missing values with respect to the 'last_review' Attribute and Manhattan has highest missing value 23.2 %

- **5.3 Missing values Analysis ('last_review vs 'room_type')**

```
airbnb_1.groupby('room_type').room_type.count()/airbnb.groupby('room_type').room_type.count()*100
```

```
room_type
Entire home/apt     19.981109
Private room        20.877004
Shared room         27.068966
Name: room_type, dtype: float64
```

```
(airbnb_1.groupby('room_type').room_type.count()/airbnb.groupby('room_type').room_type.count()*100).plot.bar()
```



From the above plot we can see that the missing value percentage is high for shared rooms 27% w.r.t to last_review Attribute.

- **5.4 Missing values Analysis ('price vs ' last_review')**

```
## To get the Mean and Median of prices w.r.t missing values in the last_review attribute.

print('Mean   = ', airbnb[airbnb['last_review'].isnull()].price.mean())
print('Median = ', airbnb[airbnb['last_review'].isnull()].price.median())
```

```
Mean   =  192.9190210903303
Median =  120.0
```

```
## To get the Mean and Median of prices w.r.t non missing values in the last_review attribute.

print('Mean   = ', airbnb[airbnb['last_review'].notnull()].price.mean())
print('Median = ', airbnb[airbnb['last_review'].notnull()].price.median())
```

```
Mean   =  142.317946605566
Median =  101.0
```

Observations :

- 'last_review' attribute is missing is almost same for all the neighbourhood

When prices are High, 'last_review' attribues are missing, which implies that if a airbnb property has high cost, reviews are less likely to be given

- For the shared rooms , 'last_review' missing value percentage is highest 27% , which implies that reviews are less likely to be given for shared rooms.

6. Univariate Analysis:

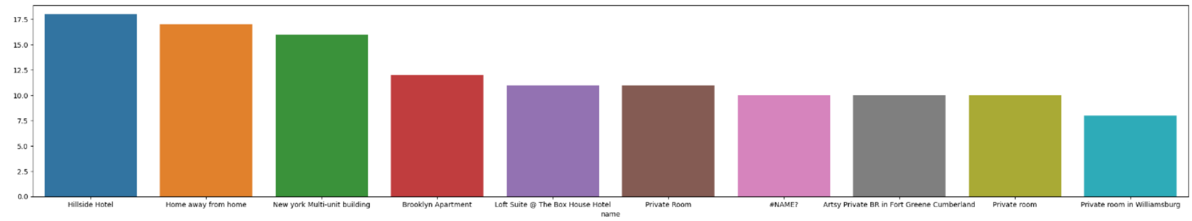
- **6.1 Name of the Airbnb Property**

```
airbnb.name.value_counts().index[:10]
```

```
Index(['Hillside Hotel', 'Home away from home', 'New york Multi-unit building', 'Brooklyn Apartment', 'Loft Suite @ The Box Hou
se Hotel', 'Private Room', '#NAME?', 'Artsy Private BR in Fort Greene Cumberland', 'Private room', 'Private room in Williamsbur
g'], dtype='object', name='name')
```

Names of top 10 Airbnb property are displayed in below bar graph

```
plt.figure(figsize=(30,5))
sns.barplot(x = airbnb.name.value_counts().index[:10] , y = airbnb.name.value_counts().values[:10])
plt.show()
```



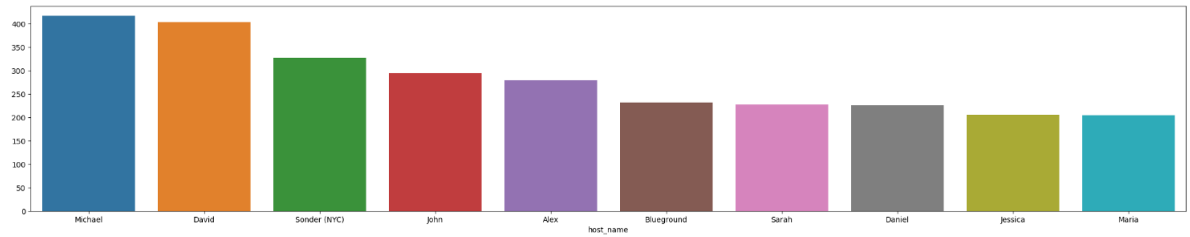
- **6.2 host_name of Airbnb Property**

```
airbnb.host_name.value_counts().index[0:10]
```

```
Index(['Michael', 'David', 'Sonder (NYC)', 'John', 'Alex', 'Blueground', 'Sarah', 'Daniel', 'Jessica', 'Maria'], dtype='objec
t', name='host_name')
```

Names of top 10 Airbnb Hosts are displayed in below bar graph

```
plt.figure(figsize=(28,5))
sns.barplot(x = airbnb.host_name.value_counts().index[:10] , y = airbnb.host_name.value_counts().values[:10])
plt.show()
```



- **6.3 neighbourhood_group in which we have the Airbnb Properties**

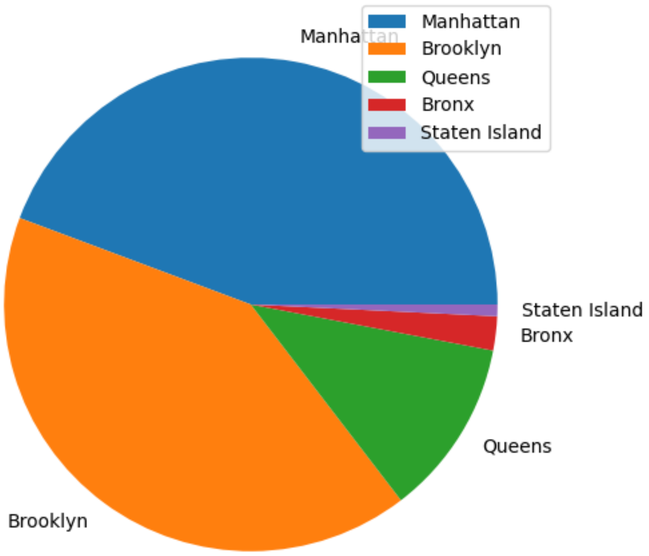
```
airbnb.neighbourhood_group.value_counts()
```

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

```
airbnb.neighbourhood_group.value_counts(normalize= True) * 100
```

```
neighbourhood_group
Manhattan        44.301053
Brooklyn         41.116679
Queens           11.588097
Bronx             2.231312
Staten Island     0.762859
Name: proportion, dtype: float64
```

```
plt.figure(figsize=(8,6))
plt.pie(x = airbnb.neighbourhood_group.value_counts(normalize= True) * 100,labels = airbnb.neighbourhood_group.value_counts(norma
plt.legend()
plt.show()
```



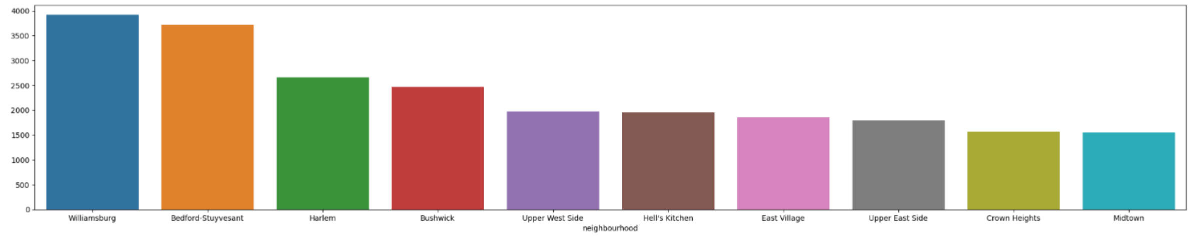
Majority of listing are in Manhattan and Brooklyn neighbourhood groups

44.3 % of listing are in Manhattan and 41.1% listining are in Brooklyn.

- **6.4 neighbourhood**

Names of top 10 Neighbourhood are displayed in below bar graph

```
plt.figure(figsize=(28,5))
sns.barplot(x = airbnb.neighbourhood.value_counts().index[:10] , y = airbnb.neighbourhood.value_counts().values[:10])
plt.show()
```



- **6.5 Room_type of in Airbnb**

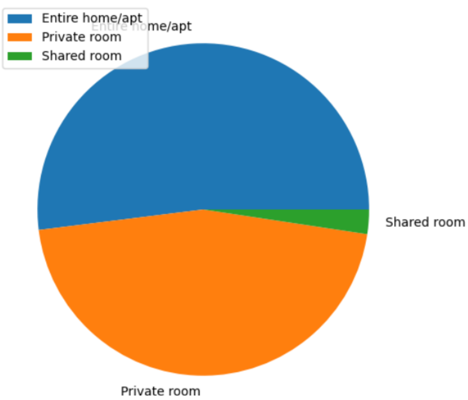
```
airbnb.room_type.value_counts()
```

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

```
airbnb.room_type.value_counts(normalize= True) * 100
```

```
room_type
Entire home/apt    51.966459
Private room       45.661111
Shared room         2.372431
Name: proportion, dtype: float64
```

```
plt.figure(figsize=(8,6))
plt.pie(x = airbnb.room_type.value_counts(normalize= True) * 100,labels = airbnb.room_type.value_counts(normalize= True).index)
plt.legend()
plt.show()
```

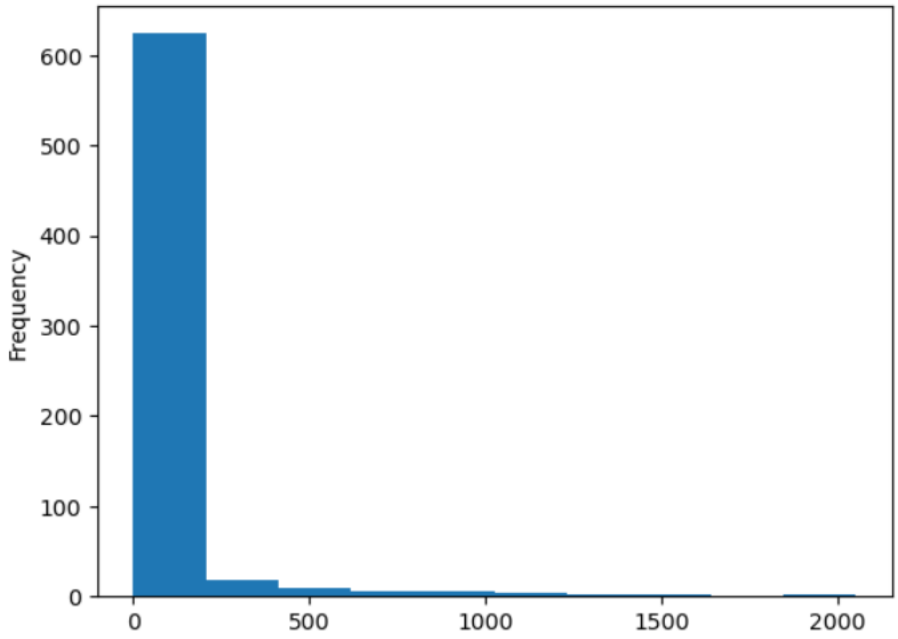


Entire home/apt (51.9%) and Private rooms (45.6%) forms Majority of Listing space type, whereas shared rooms forms only 2.3% of listing space type.

- **6.6 price**

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

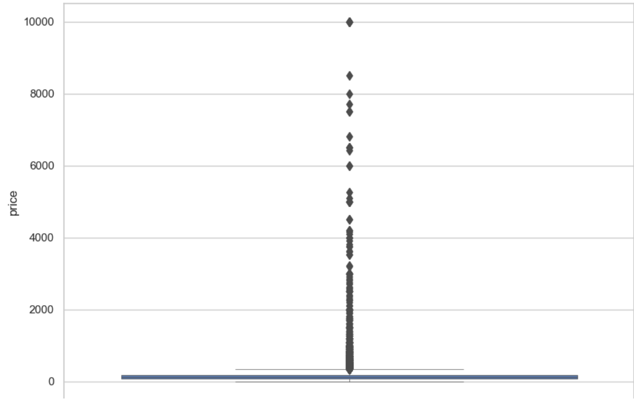
<Axes: ylabel='Frequency'>



We can observe that majority of listing is below $500

```
# checking the outliers

plt.figure(figsize=(10,7))
sns.set_theme(style="whitegrid")
#tips = sns.load_dataset("tips")
sns.boxplot(y = airbnb.price,width=0.8,
    dodge=True,
    fliersize=6,
    linewidth=.5,
    whis=1.5,
    color=None)
plt.show()
```

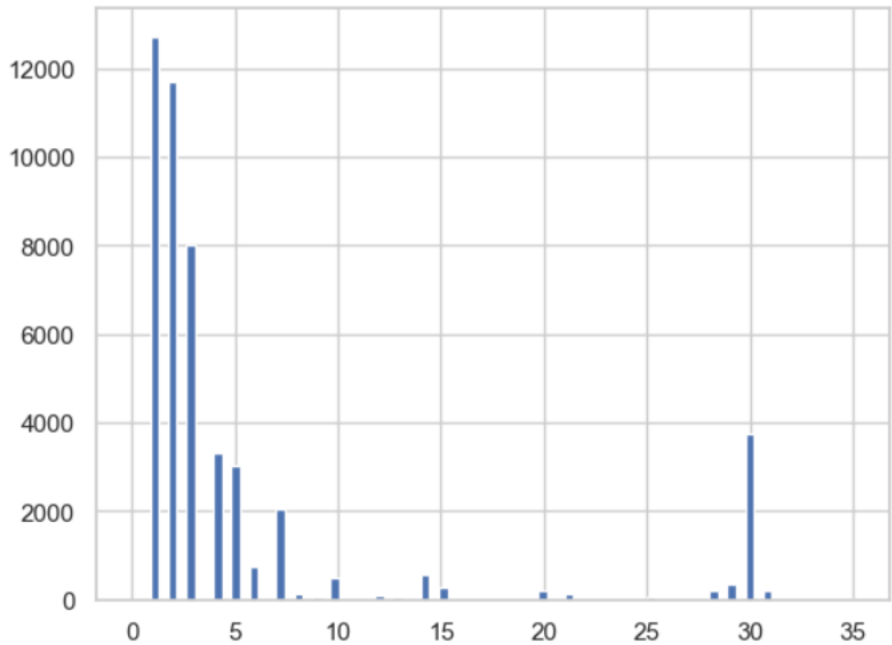


- **6.7 minimum_nights**

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



From above graph we can observe that majority of listing has minimun nights to be from 1 night to 5 nights.

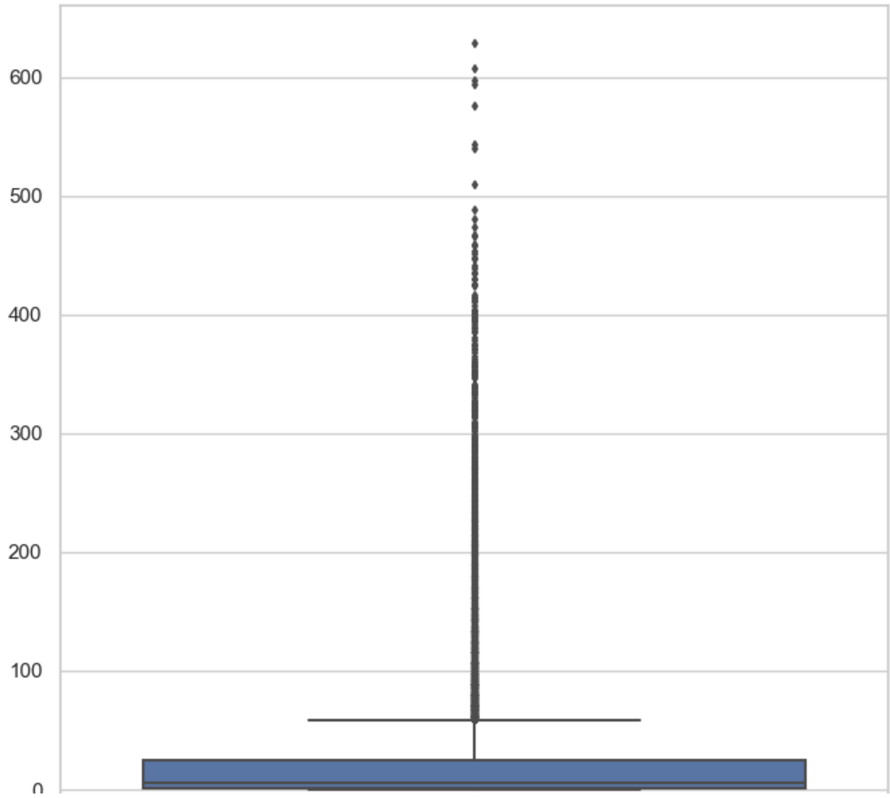
We also observe a significant number of listing with minimum 30 nights (monthly booking).

- **6.8 number_of_reviews**

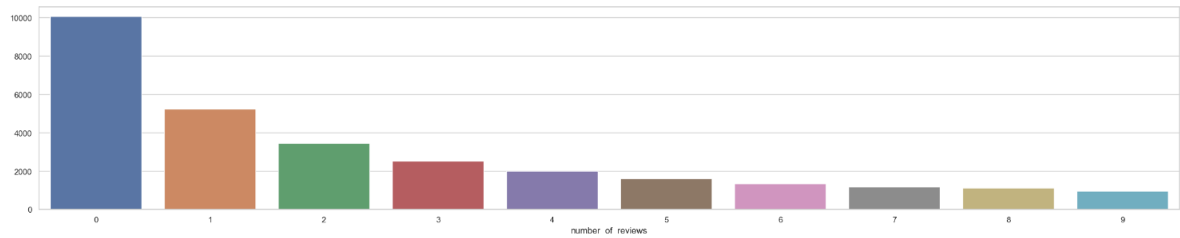
```
airbnb.number_of_reviews.describe()
```

```
count    48895.000000
mean        23.274466
std         44.550582
min          0.000000
25%          1.000000
50%          5.000000
75%         24.000000
max        629.000000
Name: number_of_reviews, dtype: float64
```

```
plt.figure(figsize=(8,8))
sns.boxplot(data = airbnb.number_of_reviews,fliersize=3)
plt.show()
```




```
plt.figure(figsize=(28,5))
sns.barplot(x = airbnb.number_of_reviews.value_counts().index[:10] , y = airbnb.number_of_reviews.value_counts().values[:10])
plt.show()
```

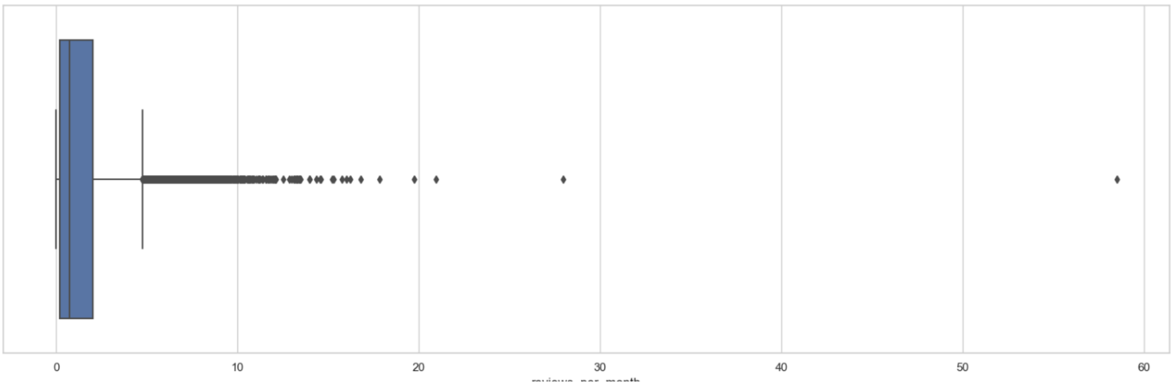


- **6.9 reviews_per_month**

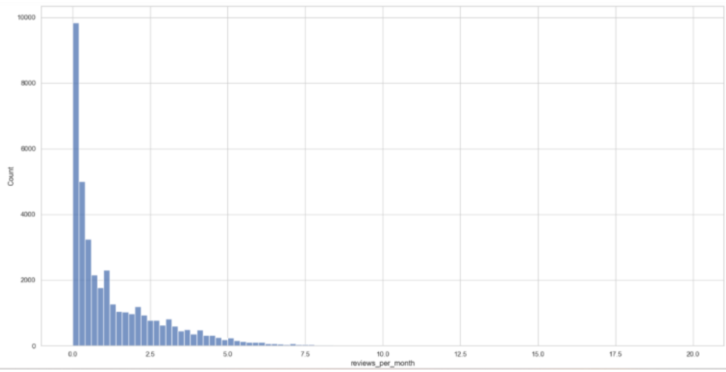
```
airbnb.reviews_per_month.describe()
```

```
count    38843.000000
mean         1.373221
std          1.680442
min          0.010000
25%          0.190000
50%          0.720000
75%          2.020000
max         58.500000
Name: reviews_per_month, dtype: float64
```

```
plt.figure(figsize = (20,6))
sns.boxplot(data = airbnb , x = 'reviews_per_month')
plt.show()
```




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



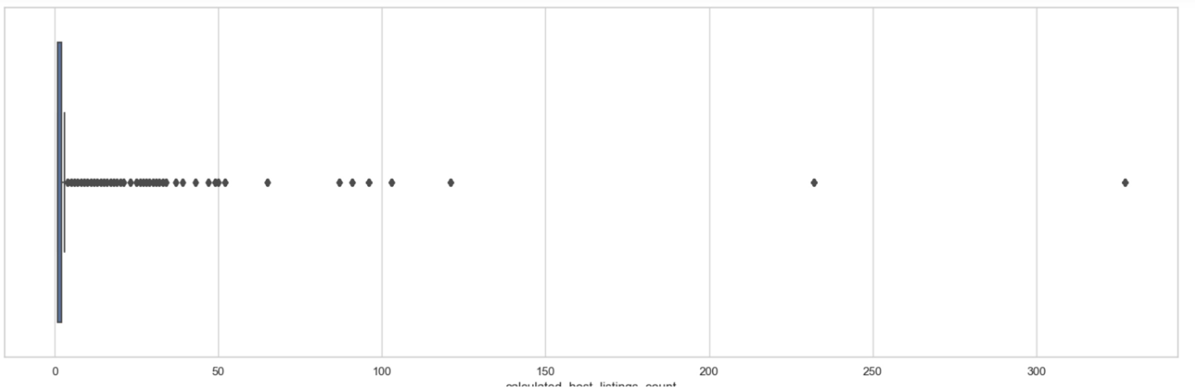
majority of listing receives 0 to 1 review per month.

- **6.10 calculated_host_listings_count**

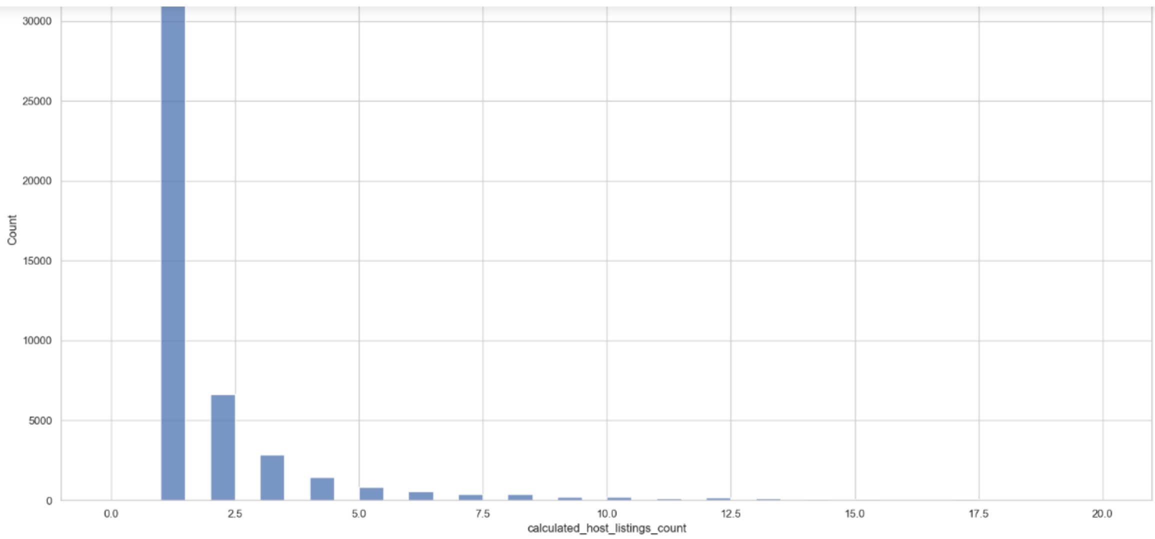
```
airbnb.calculated_host_listings_count.describe()
```

```
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 = (20,6))
sns.boxplot(data = airbnb , x = 'calculated_host_listings_count')
plt.show()
```




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



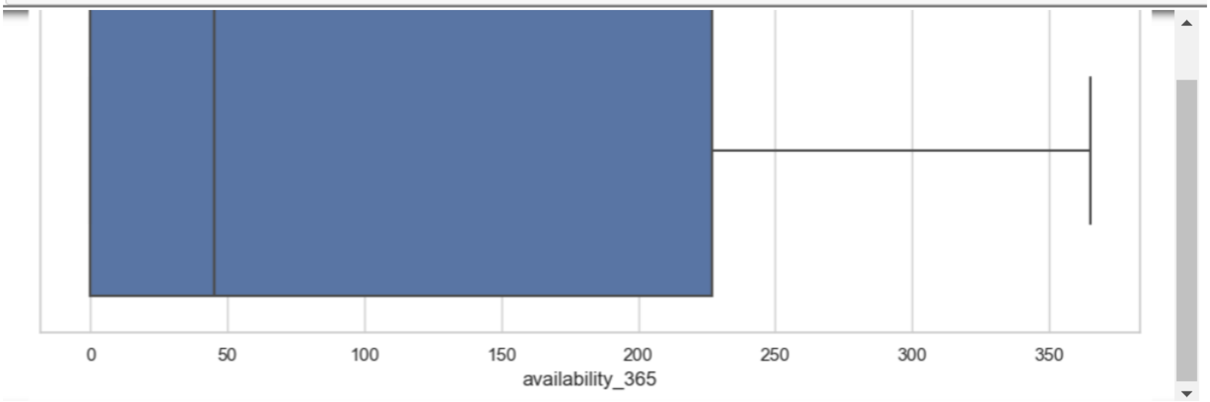
Majority of hosts have listing_count = 1

- **6.11 availability_365**

```
airbnb.availability_365.describe()
```

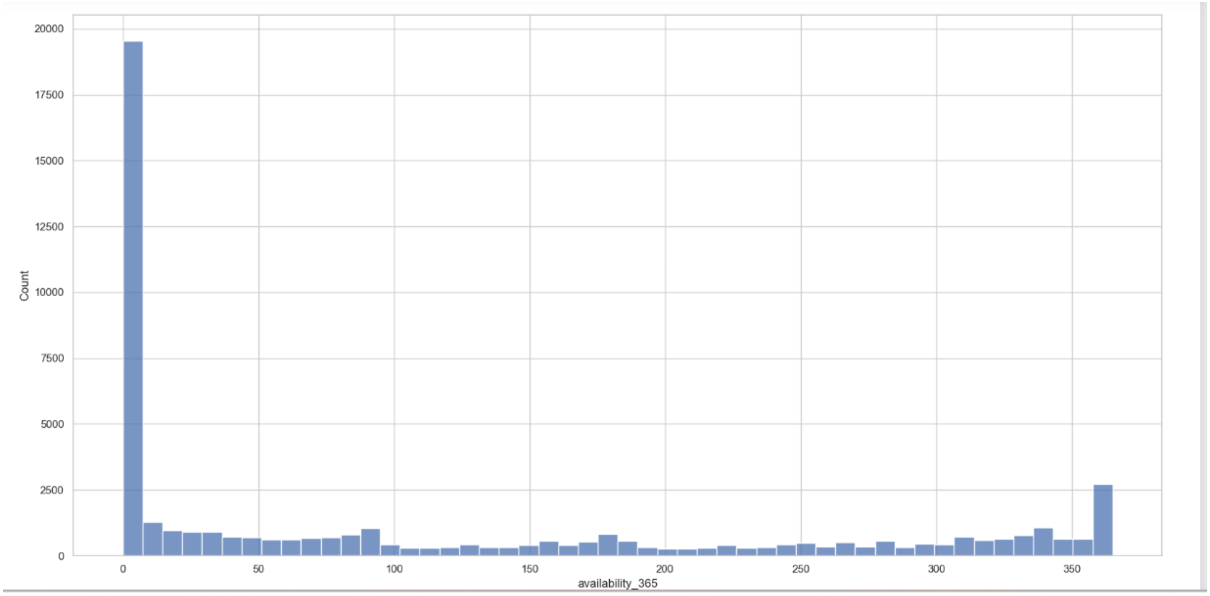
```
count    48895.000000
mean       112.781327
std        131.622289
min          0.000000
25%          0.000000
50%         45.000000
75%        227.000000
max        365.000000
Name: availability_365, dtype: float64
```

```
plt.figure(figsize = (12,4))
sns.boxplot(data = airbnb , x = 'availability_365')
plt.show()
```




```
# plotting the histogram

plt.figure(figsize = (20,10))
sns.histplot(data = airbnb, x = 'availability_365',bins=50,binrange=(0,365))
plt.show()
```



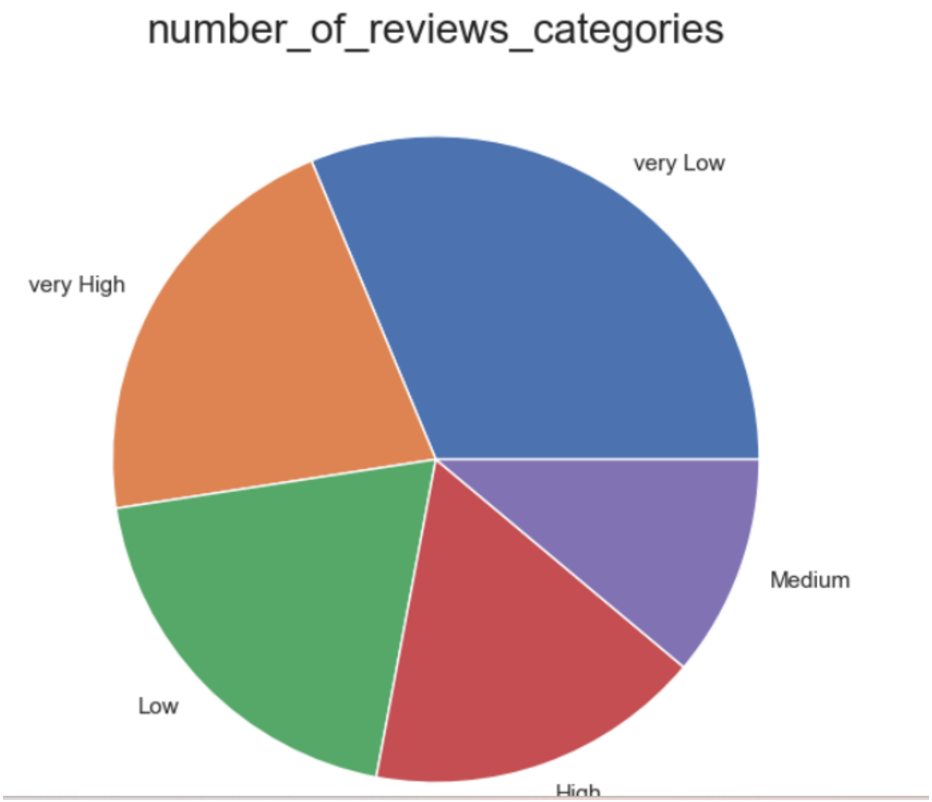
Since we are interested in number of days when the listing is available for booking. We can eliminate all the listing that avaliablity_360 = 0

- **6.12 minimum_night_categories**

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

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

```
plt.figure(figsize=(28,5))
sns.barplot(x = airbnb.number_of_reviews.value_counts().index[:10] , y = airbnb.number_of_reviews.value_counts().values[:10])
plt.show()
```



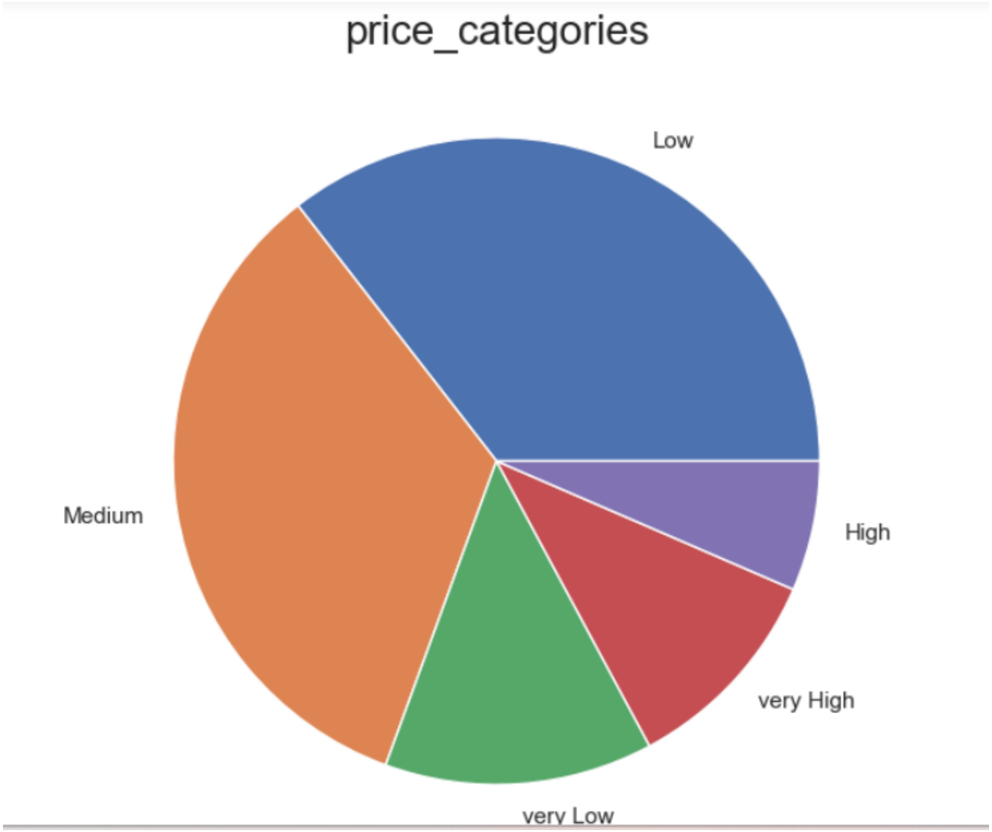
listings have 31.28 % of very low reviews

- **6.13 price_categories**

```
airbnb.price_categories.value_counts(normalize=True)*100
```

```
price_categories
Low          35.518969
Medium       33.915533
very Low     13.418550
very High    10.651396
High          6.495552
Name: proportion, dtype: float64
```

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



Most of the listing falls under Very low and low price categories

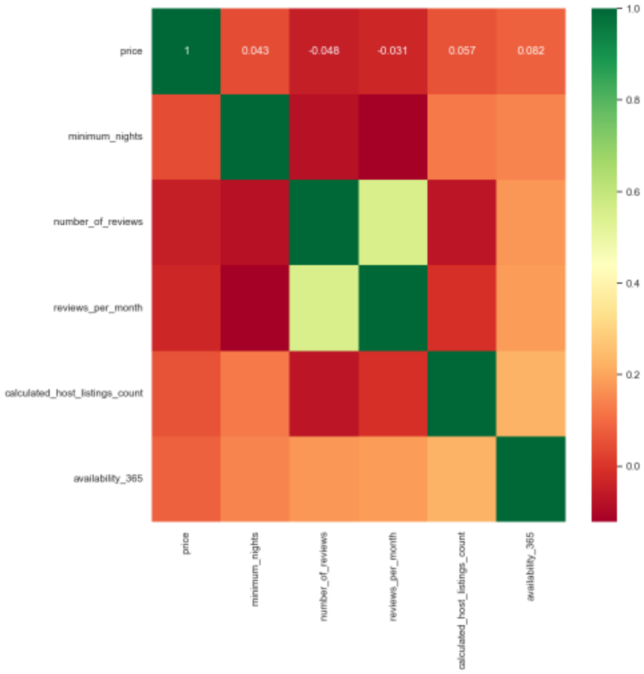
7. Bivariate and Multivariate Analysis

- **7.1 Finding correlation between numerical columns**

```
airbnb[num_cols].corr()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
price	1.000000	0.042799	-0.047954	-0.030608	0.057472	0.081829
minimum_nights	0.042799	1.000000	-0.080116	-0.121702	0.127960	0.144303
number_of_reviews	-0.047954	-0.080116	1.000000	0.549868	-0.072376	0.172028
reviews_per_month	-0.030608	-0.121702	0.549868	1.000000	-0.009421	0.185791
calculated_host_listings_count	0.057472	0.127960	-0.072376	-0.009421	1.000000	0.225701
availability_365	0.081829	0.144303	0.172028	0.185791	0.225701	1.000000

```
plt.figure(figsize=(10,10))
sns.heatmap(data=airbnb[num_cols].corr(), annot=True, cmap='RdYlGn')
plt.show()
```



- **7.2 Finding Top correlations**

```
correlation_matrix = airbnb[num_cols].corr().abs()

#we are extracting the top part of the triangle matrix without diagonal (k = 1)

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

```
sol
```

```
number_of_reviews               reviews_per_month                 0.549868
calculated_host_listings_count  availability_365                  0.225701
reviews_per_month               availability_365                  0.185791
number_of_reviews               availability_365                  0.172028
minimum_nights                  availability_365                  0.144303
                                calculated_host_listings_count    0.127960
                                reviews_per_month                 0.121702
price                           availability_365                  0.081829
minimum_nights                  number_of_reviews                 0.080116
number_of_reviews               calculated_host_listings_count    0.072376
price                           calculated_host_listings_count    0.057472
                                number_of_reviews                 0.047954
                                minimum_nights                    0.042799
                                reviews_per_month                 0.030608
reviews_per_month               calculated_host_listings_count    0.009421
dtype: float64
```

Top correlation are given below

```
sol[0:7]
```

```
number_of_reviews               reviews_per_month                 0.549868
calculated_host_listings_count  availability_365                  0.225701
reviews_per_month               availability_365                  0.185791
number_of_reviews               availability_365                  0.172028
minimum_nights                  availability_365                  0.144303
                                calculated_host_listings_count    0.127960
                                reviews_per_month                 0.121702
dtype: float64
```

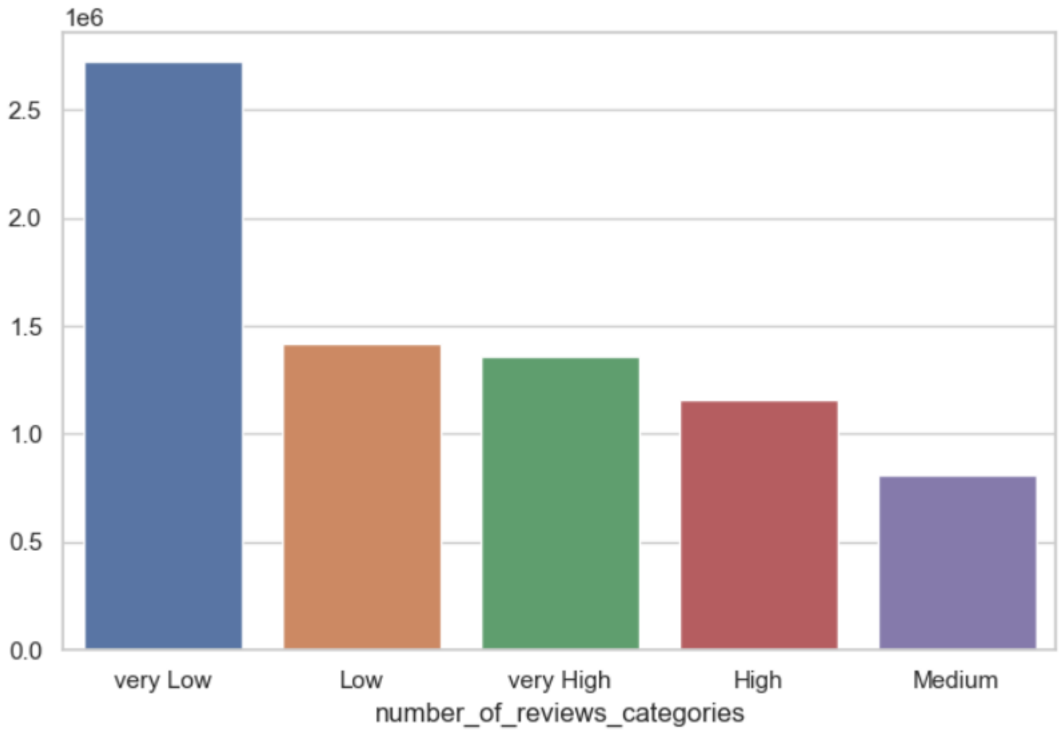
- **7.3 price vs number_of_reviews_categories**

To understand the correlation between price and number of reviews

```
x1 = airbnb.groupby('number_of_reviews_categories').price.sum().sort_values(ascending = False)
x1
```

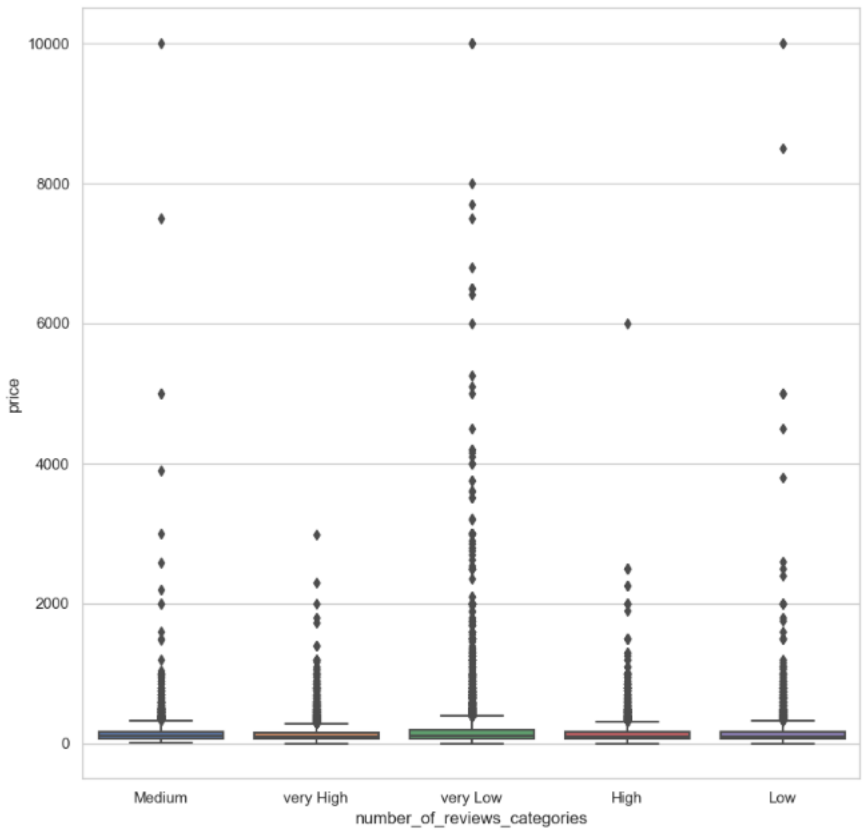
```
number_of_reviews_categories
very Low      2722793
Low           1420309
very High     1356076
High          1155254
Medium         812846
Name: price, dtype: int64
```

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




```
plt.figure(figsize=(10,10))
sns.boxplot(x = airbnb.number_of_reviews_categories , y = airbnb.price)
```

```
<Axes: xlabel='number_of_reviews_categories', ylabel='price'>
```




```
airbnb.groupby('number_of_reviews_categories').price.mean().sort_values()
```

```
number_of_reviews_categories
very High    131.199303
High         140.268820
Low          147.995103
Medium       149.695396
very Low     178.006865
Name: price, dtype: float64
```

```
airbnb.groupby('number_of_reviews_categories').price.median().sort_values()
```

```
number_of_reviews_categories
High         100.0
very High    100.0
Low          105.0
Medium       110.0
very Low     115.0
Name: price, dtype: float64
```

```
((x2.groupby('number_of_reviews_categories').price.sum()/x2.price.sum())*100).sort_values(ascending = True)
```

```
number_of_reviews_categories
Medium       10.885439
High         15.470885
very High    18.160245
Low          19.020438
very Low     36.462992
Name: price, dtype: float64
```

Listing with "very low" and "low" have high number of reviews

- **7.4 room_type vs number_of_reviews_categories**

To understand the relationship between type of rooms and number of reviews.

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

number_of_reviews_categories	High	Low	Medium	very High	very Low
room_type					
Entire home/apt	4281	5177	3015	5306	7630
Private room	3758	4213	2290	4850	7215
Shared room	197	207	125	180	451

```
airbnb.groupby('room_type').number_of_reviews.sum()
```

```
room_type
Entire home/apt    580403
Private room       538346
Shared room         19256
Name: number_of_reviews, dtype: int64
```

```
airbnb.groupby('room_type').number_of_reviews.sum()/airbnb.room_type.value_counts()
```

```
room_type
Entire home/apt    22.842418
Private room       24.112962
Shared room        16.600000
dtype: float64
```

Entire home/apt have more reviews than Shared rooms

- **7.5 'room_type' and 'price_categories'**

To understand the relation between type of room and prices

```
pd.crosstab(airbnb['room_type'], airbnb['price_categories'])
```

price_categories	High	Low	Medium	very High	very Low
room_type					
Entire home/apt	2939	4384	13198	4698	190
Private room	227	12614	3297	480	5708
Shared room	10	369	88	30	663

- The majority of "Entire home/apt" listings fall into the Medium price category, followed by Low and Very High categories. There are relatively few listings in the Very Low category

- Most "Private room" listings are in the Low price category, followed by Very Low and Medium categories. There are very few listings in the High and Very High
- The majority of "Shared room" listings are in the Very Low price category, followed by Low. There are very few listings in the other price categories.

- **7.6 room_type vs reviews_per_month**

To understand how many reviews each room type will receive per month

```
airbnb.groupby('room_type').reviews_per_month.mean()
```

```
room_type
Entire home/apt    1.306578
Private room       1.445209
Shared room        1.471726
Name: reviews_per_month, dtype: float64
```

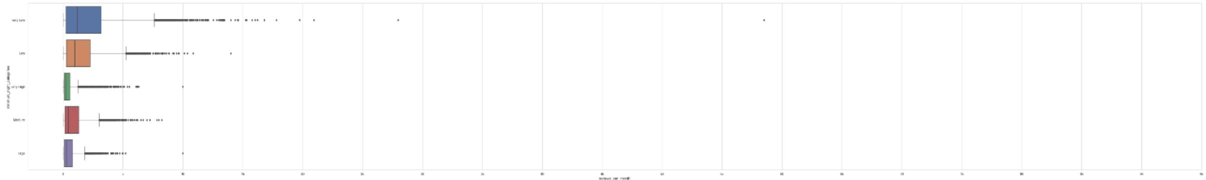
For all the three types of room there are ~1.4 reviews per month on average.

- **7.7 minimum_night_categories vs reviews_per_month**

```
airbnb.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
```

```
plt.figure(figsize=(70,10))
sns.boxplot(data = airbnb, y = 'minimum_night_categories' ,x = 'reviews_per_month')
plt.xticks(np.arange(0,100,5))
plt.show()
```



"Low" and "Very Low" Minimum Night Categories: These categories dominate in terms of the total reviews per month, indicating that listings with

lower minimum night requirements are more popular and receive more reviews.

- **7.8 availability_365_categories vs price_categories vs reviews_per_month**

```
pd.DataFrame(airbnb.groupby(['availability_365_categories','price_categories']).reviews_per_month.mean())
```

		reviews_per_month
availability_365_categories	price_categories	
High	High	1.956243
	Low	2.098447
	Medium	2.122116
	very High	2.062248
	very Low	1.789220
Low	High	1.450251
	Low	2.084674
	Medium	1.700431
	very High	1.350999
	very Low	2.091627
Medium	High	1.706095
	Low	2.130216
	Medium	1.891009
	very High	1.939057
	very Low	2.092531
very High	High	1.277655
	Low	1.642058
	Medium	1.178879
	very High	1.278092
	very Low	1.562514
very Low	High	0.428444
	Low	0.556875
	Medium	0.466402
	very High	0.403031
	very Low	0.498478

- Listings with more availability tend to receive more reviews in the medium and low price categories.
- Listings with less availability generally receive fewer reviews, regardless of price category.
- Lower-priced listings tend to receive higher average reviews across different availability categories.