In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime as dt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('/Users/4star/Desktop/Data Analysis/AB_NYC_2019.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourho
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensingt
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtov
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harle
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton H
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harle

In [4]:
```
df.size
```

Out[4]:
```
782320
```

In [5]:
```
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 [6]:
```
df.describe()
```

Out[6]:

	id	host_id	latitude	longitude	price
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000

Data Clening ...

In [7]:
```
#Checking for null values
df.isna().sum()
```

Out[7]:
```
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 [8]:
```
#there are more than 1000 missing values in other column so removing them

# Calculate the mode of the 'reviews_per_month' column
mode_reviews_per_month = df['reviews_per_month'].mode()[0]

# Fill missing values with the mode
df['reviews_per_month'].fillna(mode_reviews_per_month, inplace=True)
```

In [9]:
```
# Calculate the mode of the 'reviews_last_review ' column
mode_last_review  = df['last_review'].mode()[0]

# Fill missing values with the mode
df['last_review'].fillna(mode_last_review , inplace=True)
```

In [10]:
```
#sice only 16 and 21 values are missing in name and host_name column so b
#it doesn,t effect the data heavily..
data = df.dropna(subset=['name','host_name'])
```

In [11]:
```
data.isna().sum()
```

Out[11]:
```
id                                0
name                              0
host_id                           0
host_name                         0
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

In [12]:
```
data.info()
```

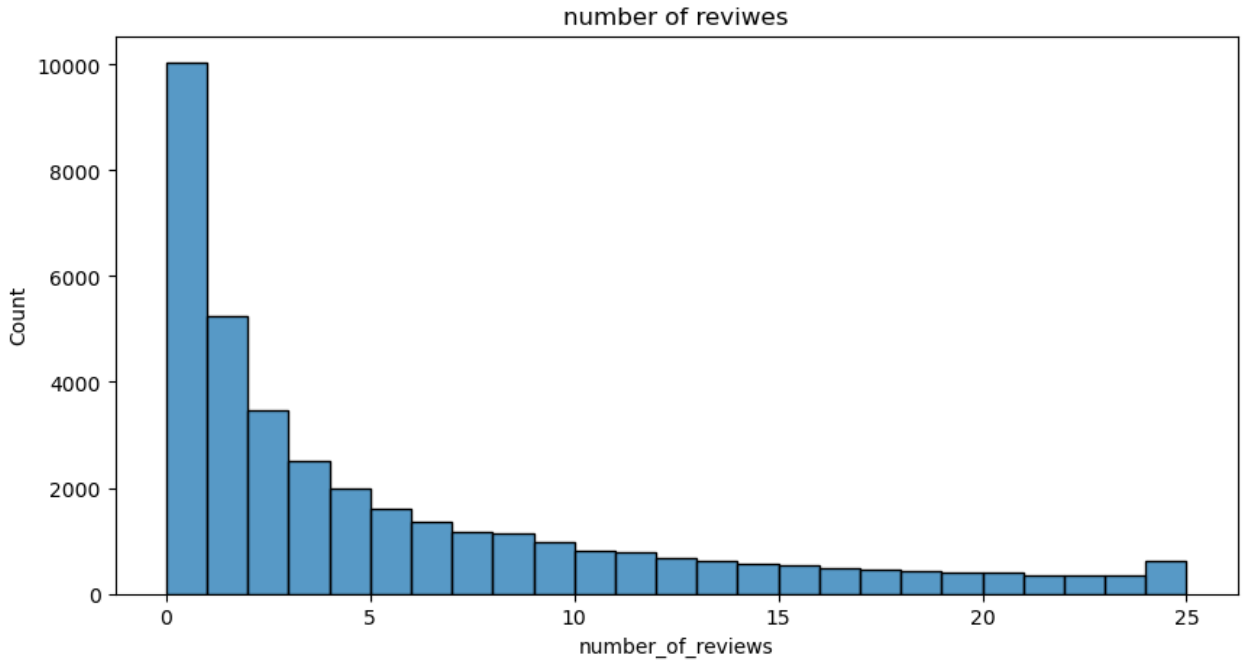
```
<class 'pandas.core.frame.DataFrame'>
Index: 48858 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48858 non-null  int64  
 1   name                            48858 non-null  object 
 2   host_id                         48858 non-null  int64  
 3   host_name                       48858 non-null  object 
 4   neighbourhood_group             48858 non-null  object 
 5   neighbourhood                   48858 non-null  object 
 6   latitude                        48858 non-null  float64
 7   longitude                       48858 non-null  float64
 8   room_type                       48858 non-null  object 
 9   price                           48858 non-null  int64  
 10  minimum_nights                  48858 non-null  int64  
 11  number_of_reviews               48858 non-null  int64  
 12  last_review                     48858 non-null  object 
 13  reviews_per_month               48858 non-null  float64
 14  calculated_host_listings_count  48858 non-null  int64  
 15  availability_365                48858 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.3+ MB
```

Data Visualization...

In [13]:
```
plt.figure(figsize=(10,5))
plt.title('number of reviwes')
sns.histplot(x=data['number_of_reviews'],bins=range(0,26,1))
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

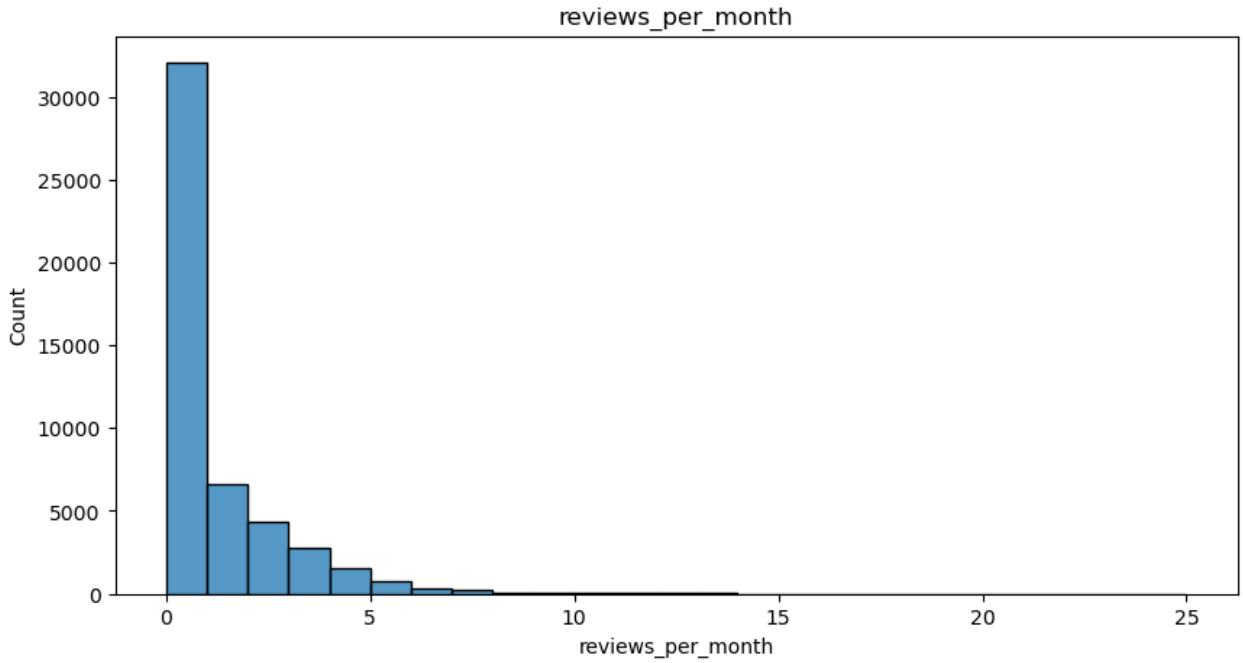
Out[13]:
```
<Axes: title={'center': 'number of reviwes'}, xlabel='number_of_review
s', ylabel='Count'>
```



In [14]:
```
plt.figure(figsize=(10,5))
plt.title('reviews_per_month')
sns.histplot(x=data['reviews_per_month'],bins=range(0,26,1))
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fut
ure version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[14]:
```
<Axes: title={'center': 'reviews_per_month'}, xlabel='reviews_per_mont
h', ylabel='Count'>
```



In [15]:
```
plt.figure(figsize=(10,5))
plt.title('last_review')
sns.histplot(x=data['last_review'],bins=range(0,26,1))
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[15]:
```
<Axes: title={'center': 'last_review'}, xlabel='last_review', ylabel='Count'>
```