In [20]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

In [21]:
```
listings = pd.read_csv(r'C:\Users\jeeva\OneDrive\Desktop\udemy\listings.csv')
```

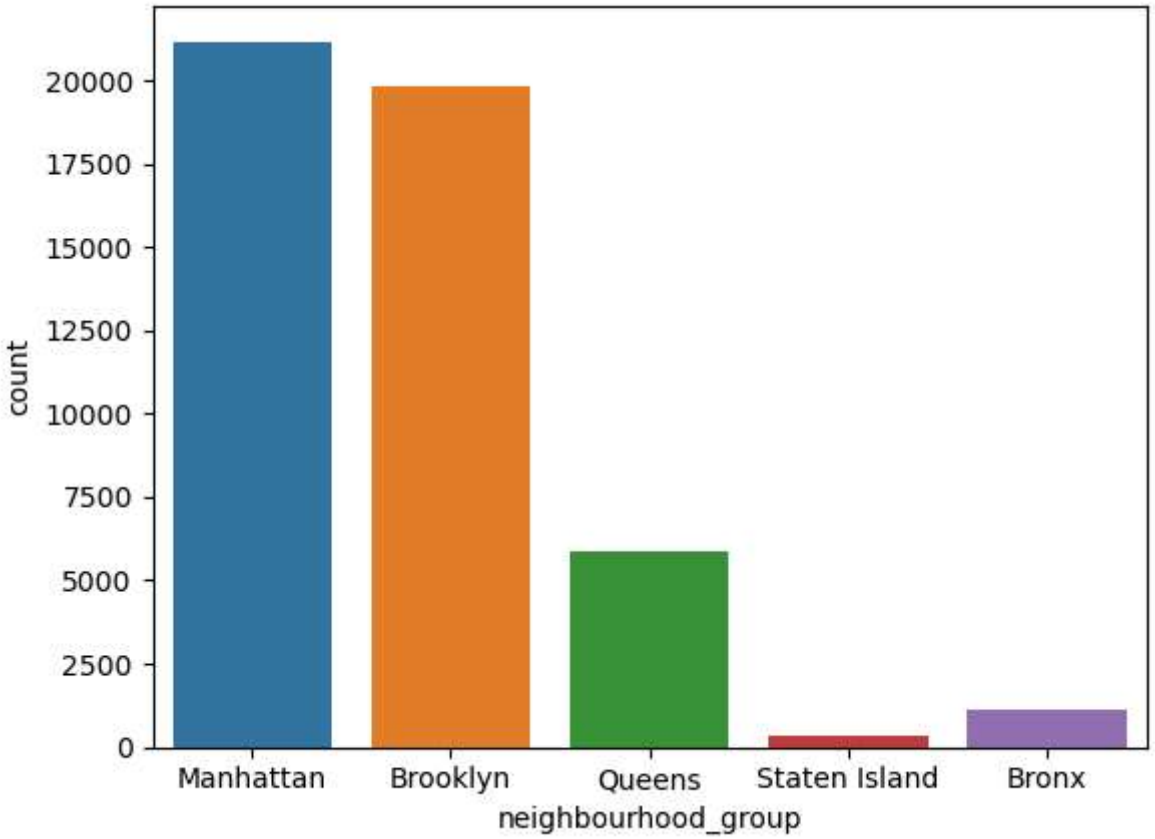
In [22]:
```
listings
```

Out[22]:

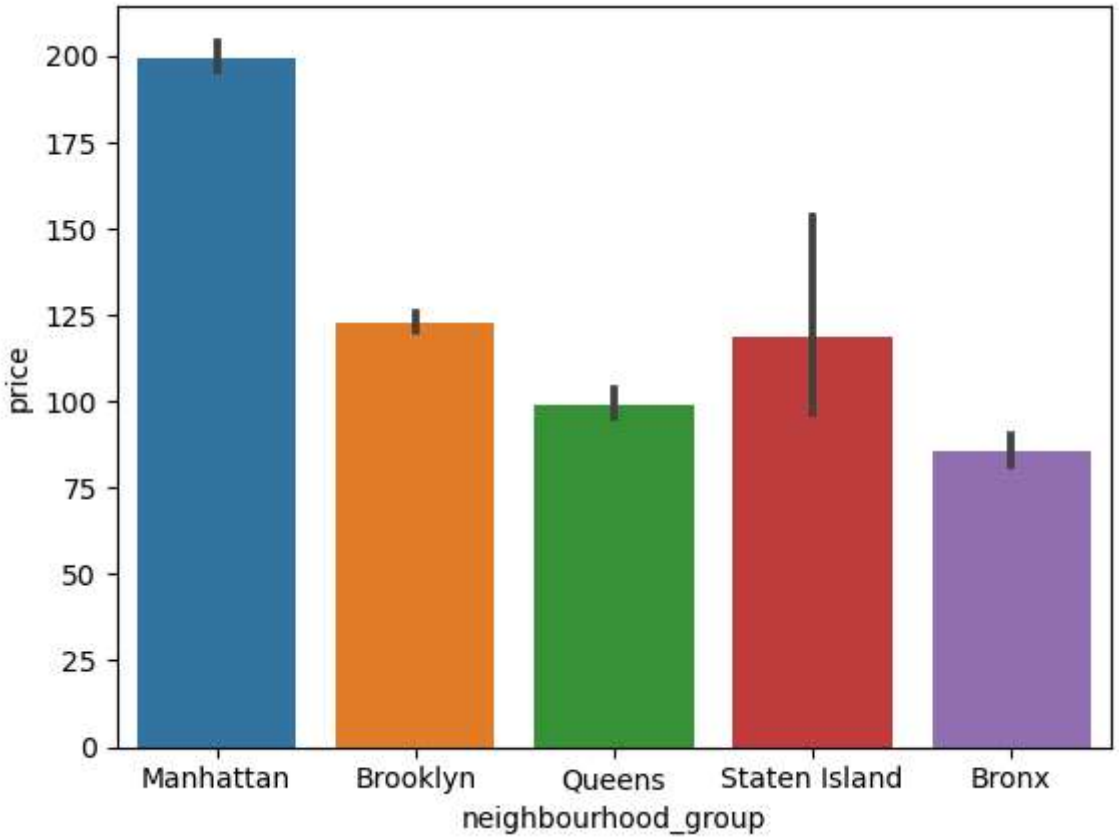
	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_typ
0	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Priva roo
1	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Enti home/a
2	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Enti home/a
3	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray Hill	40.74767	-73.97500	Enti home/a
4	5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	Priva roo
...	...	...	...	...	...	...	...	...	
48372	38564068	Top Floor Delight	207204450	Ade	Brooklyn	Cypress Hills	40.67618	-73.90764	Enti home/a
48373	38564524	Central Park Elegance	1567562	Maggie	Manhattan	Upper East Side	40.77326	-73.96650	Enti home/a
48374	38566777	Quiet room available	159902221	Nancy	Brooklyn	Bushwick	40.69907	-73.93043	Priva roo
48375	38567542	Spacious room in Brooklyn brownstone	68905617	Eric	Brooklyn	Crown Heights	40.66986	-73.93772	Priva roo
48376	38568081	Petrose home by JFK, Beach &St John Hosp.	293596932	Ebangha	Queens	Far Rockaway	40.60315	-73.75722	Priva roo

48377 rows × 16 columns

In [23]:
```
sn.countplot(x='neighbourhood_group',data = listings)
plt.show()
```

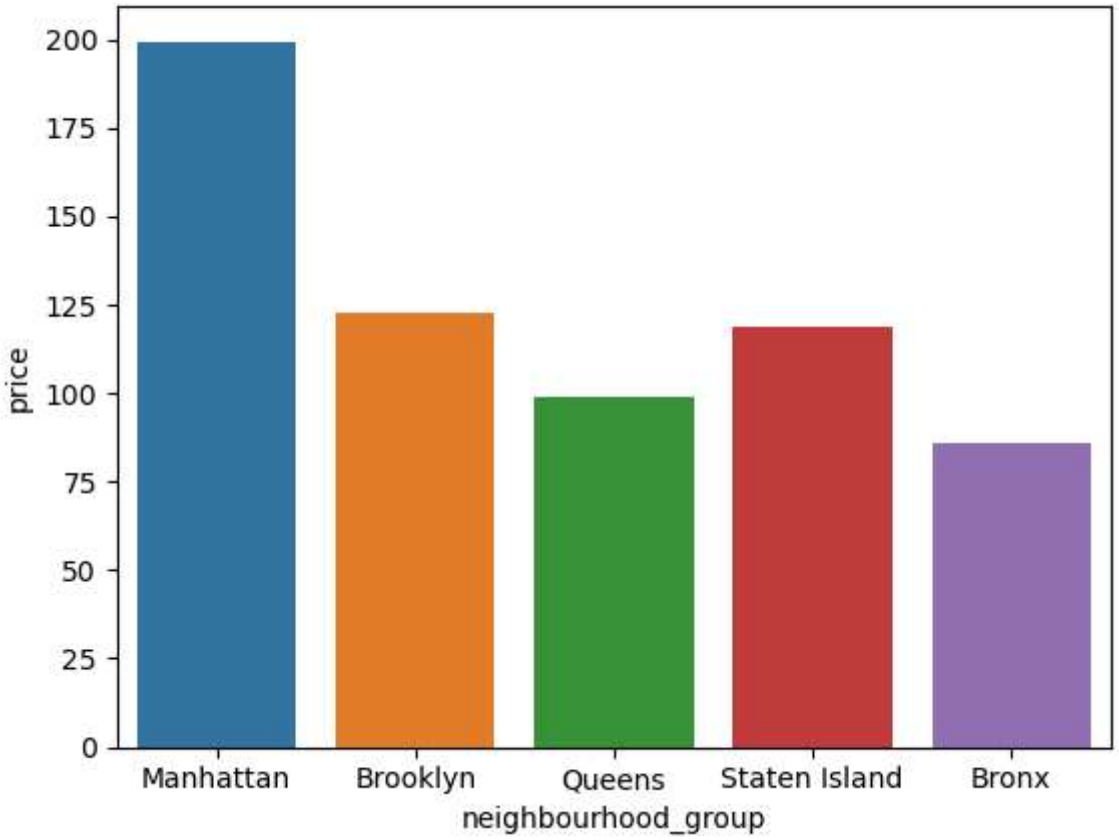


In [24]:
```
sn.barplot(x='neighbourhood_group', y='price',data = listings)
plt.show()
```



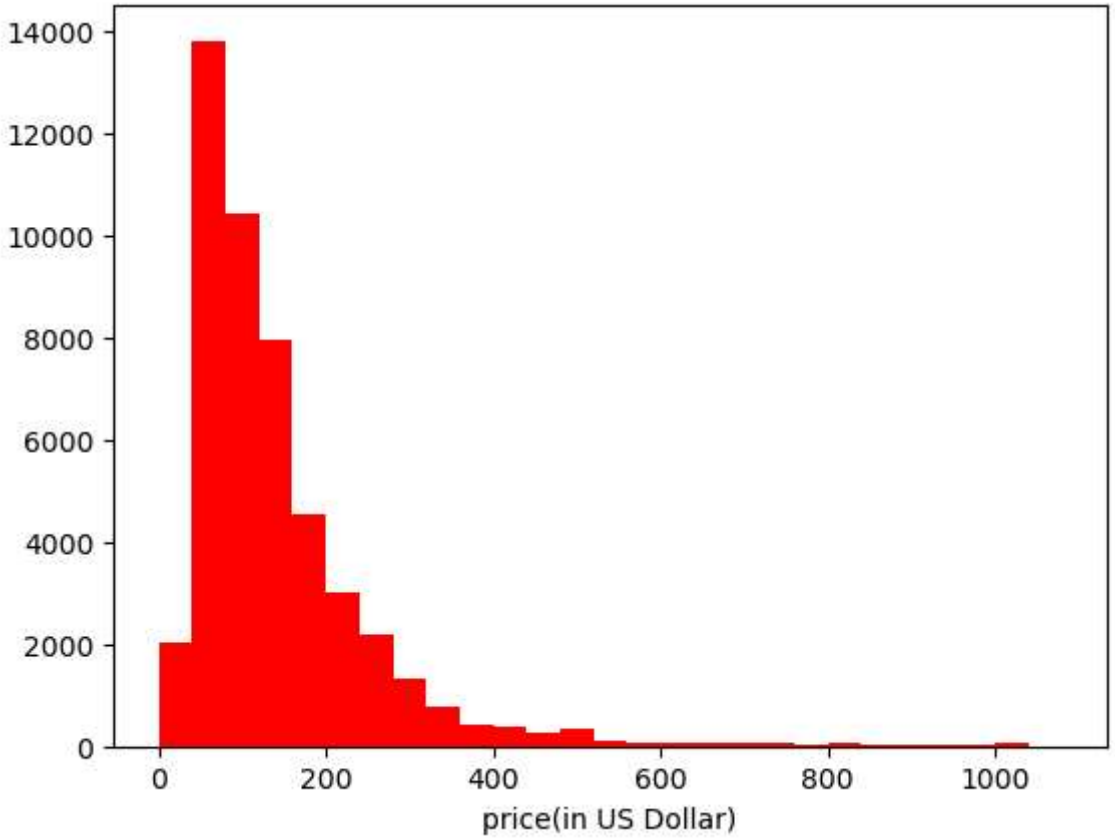
In [25]:
```
#To remove Black line from the bar chart

sn.barplot(x='neighbourhood_group', y='price',data = listings, ci = False)
plt.show()
```

```
C:\Users\jeeva\AppData\Local\Temp\ipykernel_5916\1211985129.py:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', False)` for the same effect.

  sn.barplot(x='neighbourhood_group', y='price',data = listings, ci = False)
```

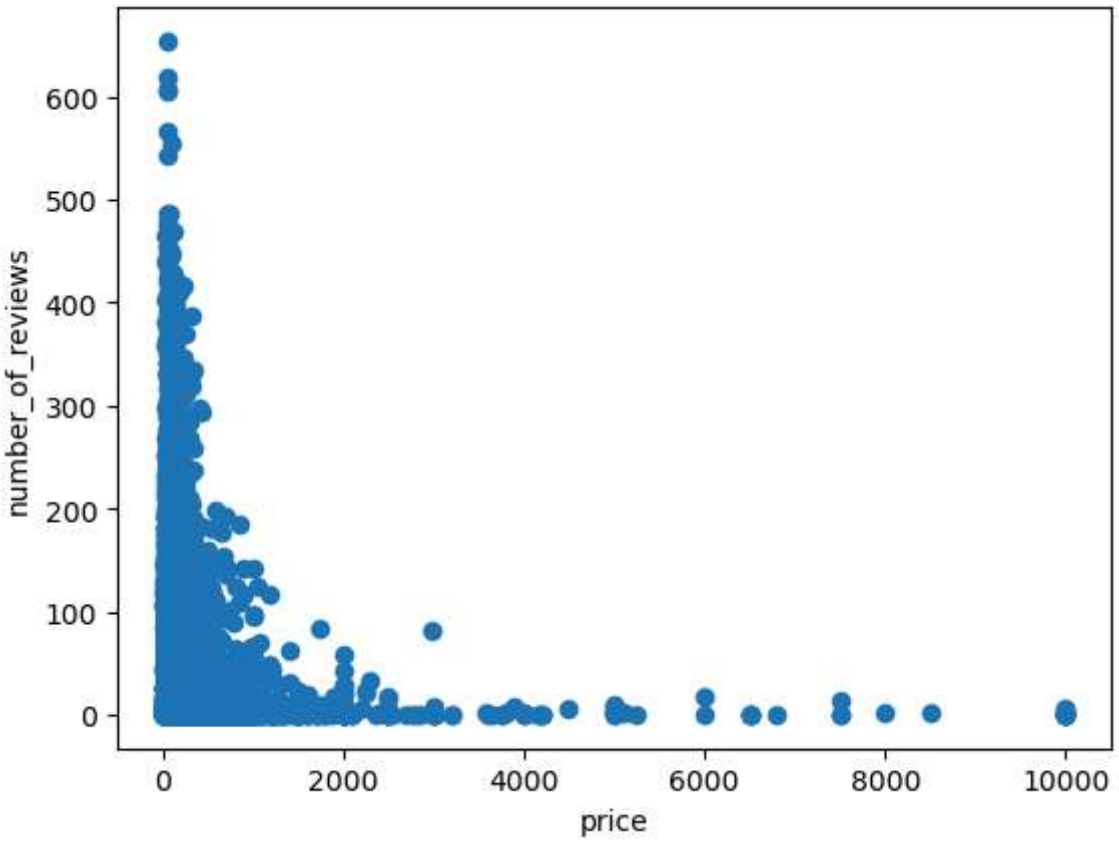


In [39]:
```
#To plot histogram

plt.hist(listings['price'], bins =np.arange(0,1100,40),color = 'RED')
plt.xlabel('price(in US Dollar)')
plt.show
```

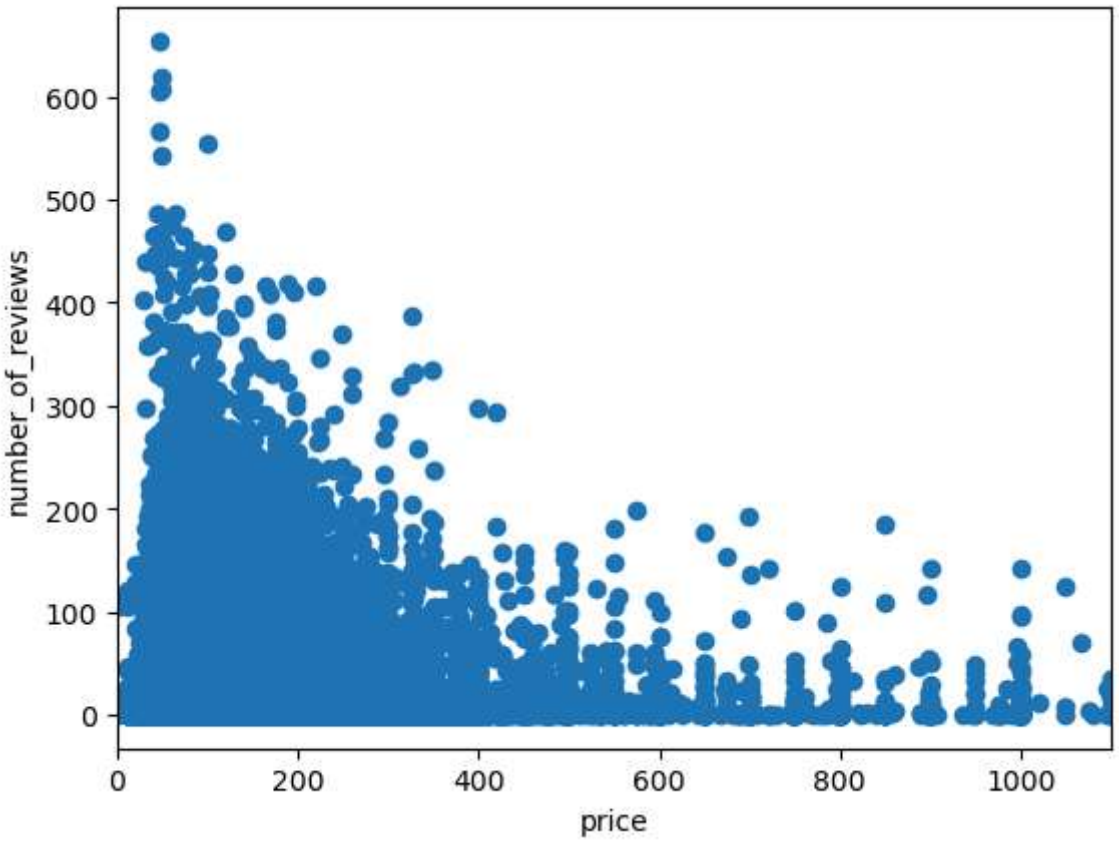
Out[39]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



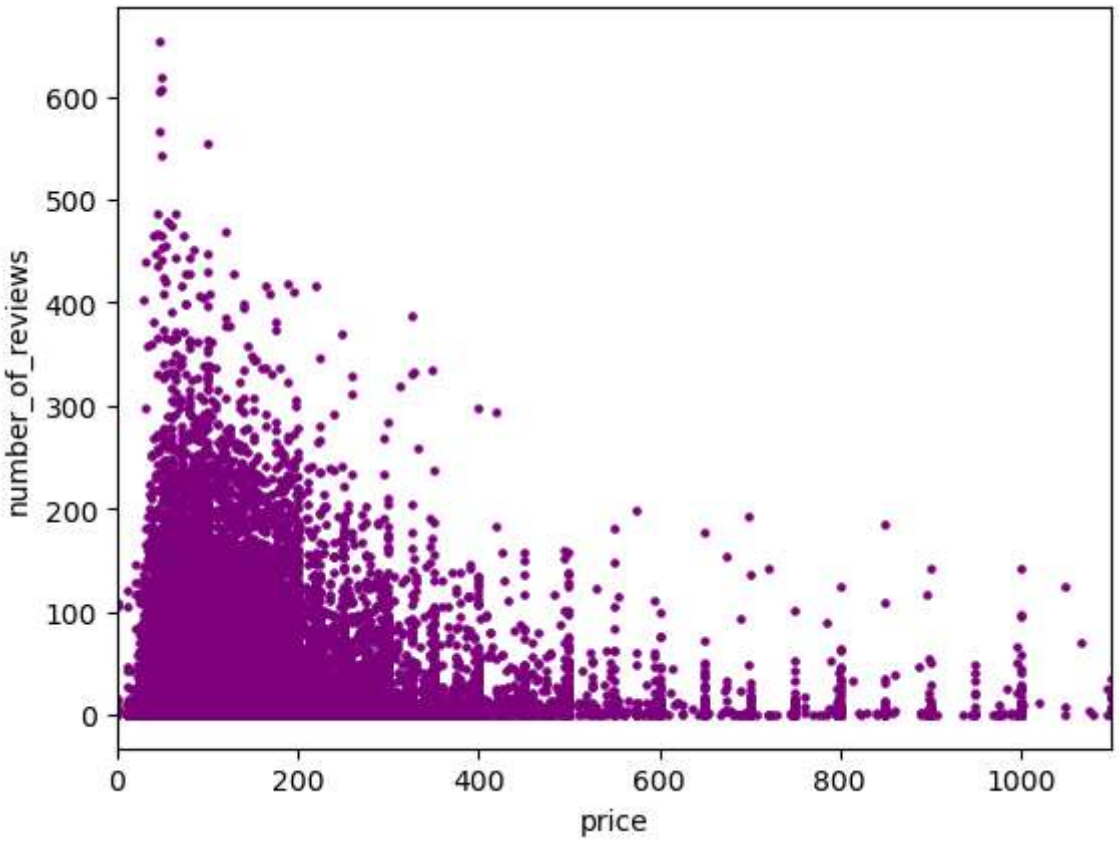
In [32]:
```
#To plot a scatter graph

plt.scatter(x = listings['price'], y = listings['number_of_reviews'])
plt.xlabel('price')
plt.ylabel('number_of_reviews')
plt.show()
```



In [33]:
```
# To ristrict the x axis
plt.scatter(x = listings['price'], y = listings['number_of_reviews'])
plt.xlabel('price')
plt.ylabel('number_of_reviews')
plt.xlim(0,1100)
plt.show()
```



In [37]:
```
#To change the size of dots and color

plt.scatter(x = listings['price'], y = listings['number_of_reviews'], s= 5,color = 'purple')
plt.xlabel('price')
plt.ylabel('number_of_reviews')
plt.xlim(0,1100)
plt.show()
```


number_of_reviews
price
0
100
200
300
400
500
600
0
200
400
600
800
1000

In [ ]: