گزارش تمرین شماره ۱

واحد درسی داده کاوی

اساتید محترم :

جناب آقای دکتر فراهانی

جناب آقای دکتر خردپیشه

هدیه آشوری ۹۹۴۲۲۰۲۲

۱۴۰۰/۰۱/۱۵

- **بررسی داده ها**

در ابتدا پکیج های مورد نیاز را Import

می کنیم :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as  plt
import seaborn as sns
from scipy import stats
import urllib
```

سپس data مورد نظر را با دستور ذیل فراخوانی می کنیم

و

با دستور df.shape ابعاد DataFrame را به دست می آوریم (۱۸ ستون و ۴۸۸۹۵ ردیف)

Loading the data

```
# csv file read/load
df = pd.read_csv('D:\\AB_NYC_2019.csv')
df.shape
```

```
(48895, 18)
```

با دستور df.info() خلاصه مختصری از داده ها را به شرح ذیل دریافت می کنیم .

```
In [26]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 18 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  Unnamed: 16                     0 non-null      float64
 17  Unnamed: 17                     0 non-null      float64
dtypes: float64(5), int64(7), object(6)
memory usage: 6.7+ MB
```

با دستور print(' the field name of data:',df.columns) نام ستون ها را دریافت کردم

```
print(' the field name of data:',df.columns)
```

```
the field name of data: 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', 'Unnamed: 16', 'Unnamed: 17'],
      dtype='object')
```

با دستور df جدولی از داده دریافت کردم

In [28]:
```
# data output
df
```

Out[28]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	numb
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	
...	...	...	...	...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90	7	

48895 rows × 18 columns

با دستور df.dtypes ، نوع دیتا ها را بررسی کردم

```
In [29]: ### Check data types of all columns
         df.dtypes

Out[29]: id                                int64
         name                             object
         host_id                           int64
         host_name                        object
         neighbourhood_group              object
         neighbourhood                    object
         latitude                        float64
         longitude                       float64
         room_type                        object
         price                             int64
         minimum_nights                    int64
         number_of_reviews                 int64
         last_review                      object
         reviews_per_month               float64
         calculated_host_listings_count    int64
         availability_365                  int64
         Unnamed: 16                     float64
         Unnamed: 17                     float64
         dtype: object
```

با دستور df.head()، پنج خط اول داده ها را دریافت کردم

```
In [30]: df.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_rev
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	10/19/2
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	5/21/2
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	7/5/2
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	11/19/2

با دستور df.tail() پنج خط آخر داده ها را مشاهده کردم

```
df.tail()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90	7	

با دستور df.describe () خلاصه ای از اطلاعات عددی data را به دست آوردم

```
df.describe()
```

	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.
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.

با دستور df.isnull().sum() داده های null را بررسی کردم

```
df.isnull().sum()
```

```
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
Unnamed: 16                     48895
Unnamed: 17                     48895
dtype: int64
```

با دستور ذیل ، داده های null را پر کردم

```
df.fillna({'reviews_per_month':0}, inplace=True)
df.fillna({'name':"NoName"}, inplace=True)
df.fillna({'host_name':"NoName"}, inplace=True)
df.fillna({'last_review':"NotReviewed"}, inplace=True)
df.fillna({'Unnamed: 16':0}, inplace=True)
df.fillna({'Unnamed: 17':0}, inplace=True)
```

مجددا با دستور df.isnull().sum() به بررسی نتیجه پرداختم

همانگونه که می بینیم تمامی صفر شده است

```
df.isnull().sum()
```

```
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
Unnamed: 16                       0
Unnamed: 17                       0
dtype: int64
```

با df.price.describe() ستون قیمت را بررسی کردم

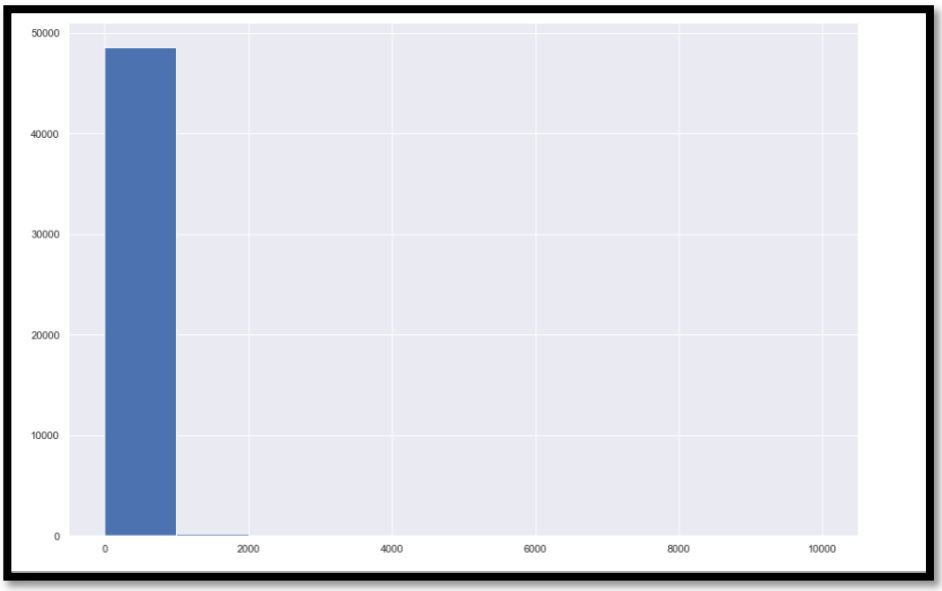
در این جا مشاهده می کنیم که میانگین صفر هست

```
df.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
```

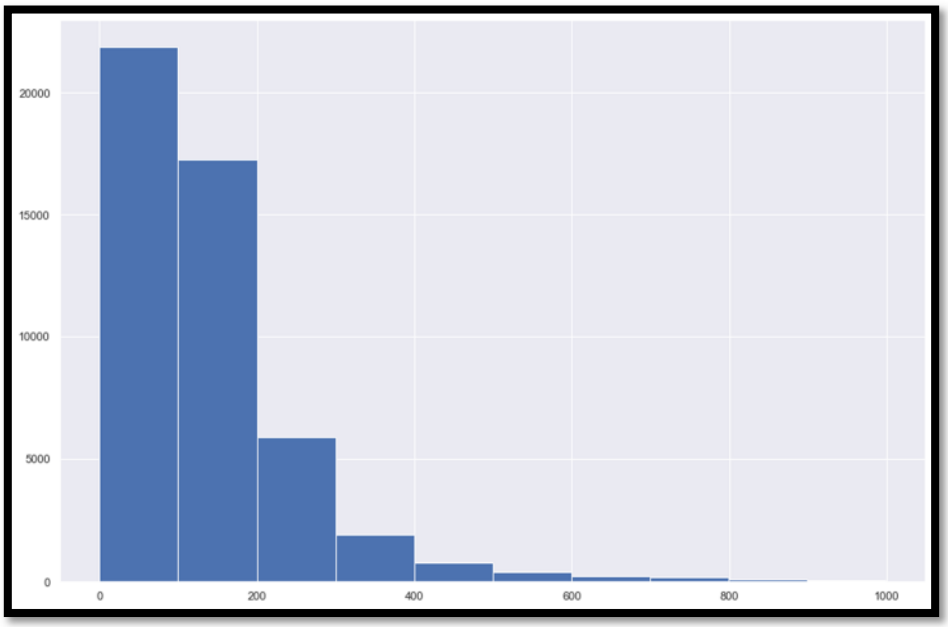
توزیع قیمت را مشاهده خواهیم کرد

```
hist_price=df["price"].hist()
```



در نمودار دریافت شده می بینیم که قیمت اکثر لیست ها کمتر از ۱۰۰۰ دلار هست .

با دستور `hist_price1=df["price"][df["price"]<1000].hist()` نمودار ذیل را داریم و اجازه دادیم که هیستوگرام با قیمت های کمتر از ۱۰۰۰ دلار رسم گردد



تصویر واضح تری داریم اما خوب می بینیم که توزیع به سمت چپ متمایل هست و چولگی دارد

حالا می خواهیم ببینیم که چه تعدادی از لیست ما مبلغ بیشتر از ۱۰۰۰ دلار داشته اند؟

با دستور dataset =df[df["price"]>1000] این مورد را انجام داده و با دستور dataset جدول ذیل را دریافت می کنیم که دارای ۲۳۹ ردیف هست

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	nu
496	174966	Luxury 2Bed/2.5Bath Central Park View	836168	Henry	Manhattan	Upper West Side	40.77350	-73.98697	Entire home/apt	2000	30	
762	273190	6 Bedroom Landmark West Village Townhouse	605463	West Village	Manhattan	West Village	40.73301	-74.00268	Entire home/apt	1300	5	
946	363673	Beautiful 3 bedroom in Manhattan	256239	Tracey	Manhattan	Upper West Side	40.80142	-73.96931	Private room	3000	7	
1105	468613	$ (Phone number hidden by Airbnb) weeks - room f	2325861	Cynthia	Manhattan	Lower East Side	40.72152	-73.99279	Private room	1300	1	
1480	664047	Lux 2Bed/2.5Bath Central Park Views	836168	Henry	Manhattan	Upper West Side	40.77516	-73.98573	Entire home/apt	2000	30	
...	...	...	...	...	...	...	...	...	...	...	...	...
48080	36074198	Luxury apartment 2 min to times square	203565865	Vinícius	Manhattan	SoHo	40.72060	-74.00023	Entire home/apt	1308	2	
48304	36189195	Next to Times Square/Javits/MSG! Amazing 1BR!	270214015	Rogelio	Manhattan	Hell's Kitchen	40.75533	-73.99866	Entire home/apt	2999	30	
48305	36189257	2BR Near Museum Mile! Upper East Side!	272166348	Mary Rotsen	Manhattan	Upper East Side	40.78132	-73.95262	Entire home/apt	1999	30	
48523	36308562	Tasteful & Trendy Brooklyn Brownstone, near Train	217732163	Sandy	Brooklyn	Bedford-Stuyvesant	40.68767	-73.95805	Entire home/apt	1369	1	
48535	36311055	Stunning & Stylish Brooklyn Luxury, near Train	245712163	Urvashi	Brooklyn	Bedford-Stuyvesant	40.68245	-73.93417	Entire home/apt	1749	1	

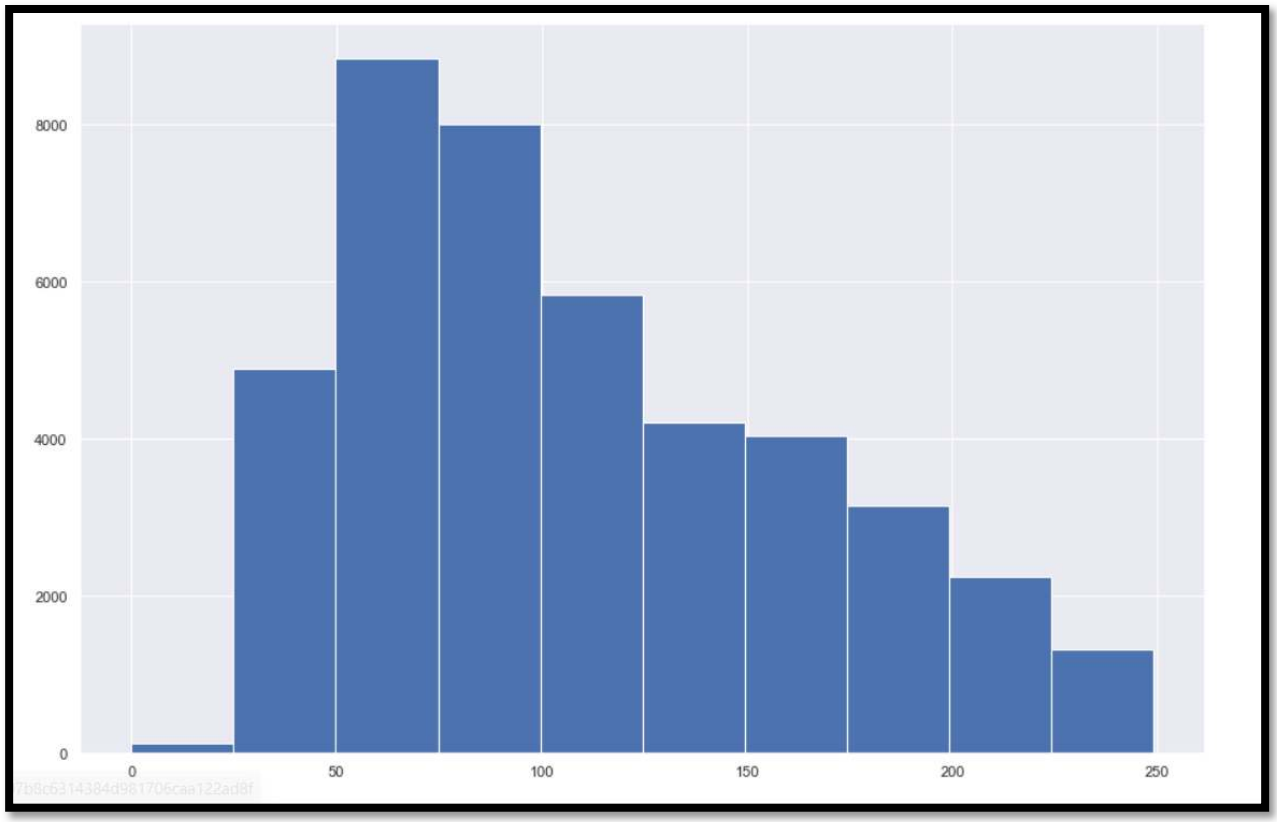
239 rows × 18 columns

این جدول می گوید که ۲۳۹ تا از لیست ما دارای قیمت روزانه بالاتر از ۱۰۰۰ دلار می باشد این لیست بسیار باشکوه و رویایی است اما مطمعنا در زمان ورود خطایی رخ داده است

با دستور dataset=df[df["price"]<1000] قیمت های کمتر از ۱۰۰۰ را در dataset ریختیم

حالا می خواهیم نموداری از قیمت های بالاتر از ۲۵۰ دلار رسم کنیم :

```
hist_price2=dataset["price"][dataset["price"]<250].hist()
```



در این جا ما نمودار توزیع گاوسی بهتری داریم

پس آستانه قیمت را روی ۲۵۰ قرار می دهیم

مجددا با دستور dataset["price"].describe() ستون قیمت را بررسی می کنیم

```
### Looking at the price column again
dataset["price"].describe()
```

```
count    42669.000000
mean       107.897748
std         53.803457
min          0.000000
25%         65.000000
50%         99.000000
75%        150.000000
max        249.000000
Name: price, dtype: float64
```

همانگونه که مشاهده می کنید در این جا میزان min قیمت صفر هست

سوالی که پیش می آید این است که آیا ممکن هست ما خانه ای را اجاره کنیم و هزینه ای پرداخت نکنیم ؟!!

طبیعتا خیر

پس data من مشکل دارد و من باید آن را اصلاح کنم

بنابراین اقدام به data cleaning می کنم و مقادیر price=0 را حذف می کنم

با دستور data = df[df['price']>0] و data.head(5) داده های price=0 را حذف کردم و ۵ ردیف اول نمایش داده شد

```
data = df[df['price']>0]
data.head(5)
```

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

مجددا با دستور data['price'].describe() اطلاعات قیمت را بررسی می کنیم همانطور که مشاهده می کنید min با مقدار صفر حذف شد

```
data['price'].describe()
```

```
count    48884.000000
mean       152.755053
std        240.170260
min         10.000000
25%         69.000000
50%        106.000000
75%        175.000000
max      10000.000000
Name: price, dtype: float64
```

حالا با دستور len طول داده قیمت را به دست می آوریم

```
len(data)
```

```
48884
```

می خواهیم داده های outlier را در ستون قیمت یا price پیدا کنیم

با دستور data1 = data[data['price']<= data['price'].mean() + 3*data['price'].std()]

و رنج ۳- برابر std و ۳ برابر std را حذف می کنیم

مجددا با دستور len طول داده قیمت را به دست می آوریم

همانگونه که مشاهده می کنید تعدادی داده حذف شده است و

طول داده کاهش یافته است

```
len(data1)
```

```
48496
```

مجددا با دستور data['price'].describe()

اطلاعات قیمت را بررسی می کنیم مشاهده می کنید

که مقدار std , ... تغییر کرده است

```
data1.price.describe()
```

```
count    48496.000000
mean       138.778373
std        107.550128
min         10.000000
25%         69.000000
50%        105.000000
75%        175.000000
max        860.000000
Name: price, dtype: float64
```

حالا می خواهیم ببینیم در نواحی پنج گانه نیویورک چقدر آگهی داشته ام با دستور ذیل این مورد را انجام می دهیم

data1['neighbourhood_group'].value_counts()

```
data1['neighbourhood_group'].value_counts()
```

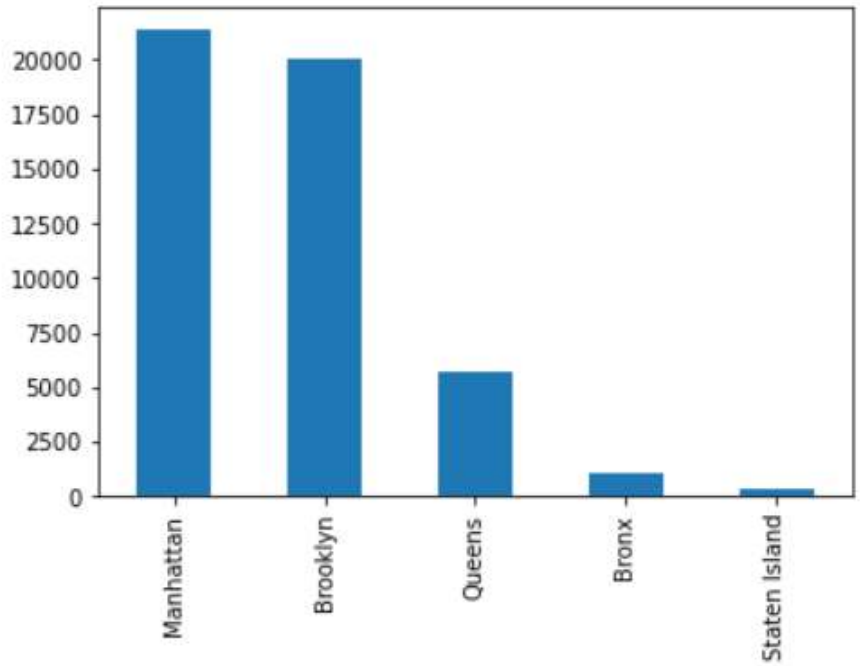
```
Manhattan        21377
Brooklyn         20011
Queens            5650
Bronx             1088
Staten Island      370
Name: neighbourhood_group, dtype: int64
```

با دستور ذیل برای رسم نمودار اقدام می کنیم :

data1['neighbourhood_group'].value_counts().plot(kind = "bar")

```
data1['neighbourhood_group'].value_counts().plot(kind = "bar")
```

```
<AxesSubplot:>
```

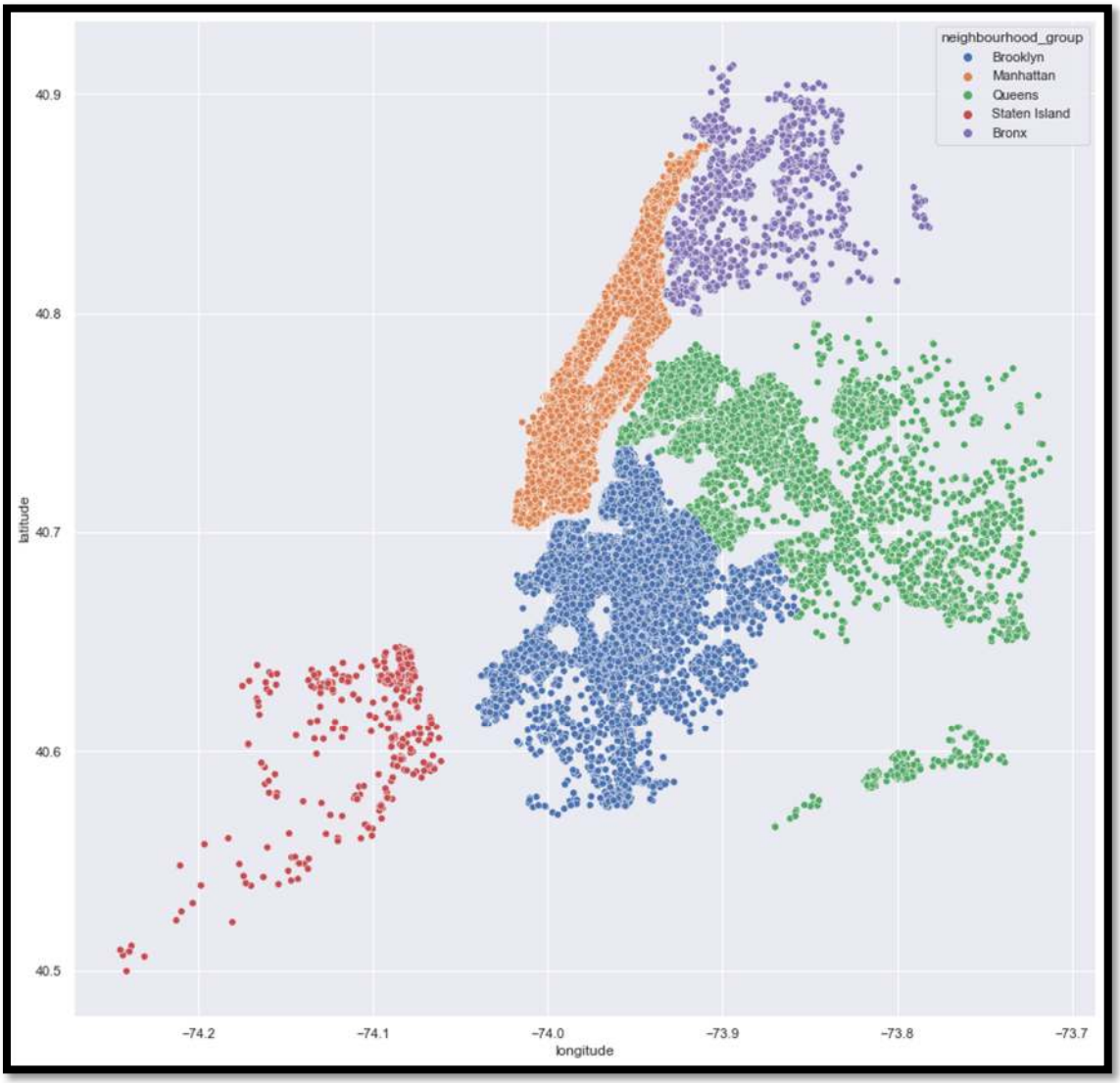


به ترتیب از چپ به راست بیشترین آگهی تا کمترین آگهی نمایش داده شده است .

Manhattan بیشترین Request را داشته است و Staten Island کمترین Request را داشته است

حال می خواهم از طریق scatterplot پراکندگی request ها را در نیویورک نمایش دهم

```
plt.figure(figsize=(15, 15))
sns.scatterplot(x=data.longitude,y=data.latitude,hue=data.neighbourhood_grou)
```



گرانترین و ارزان ترین شهر در نیویورک ؟

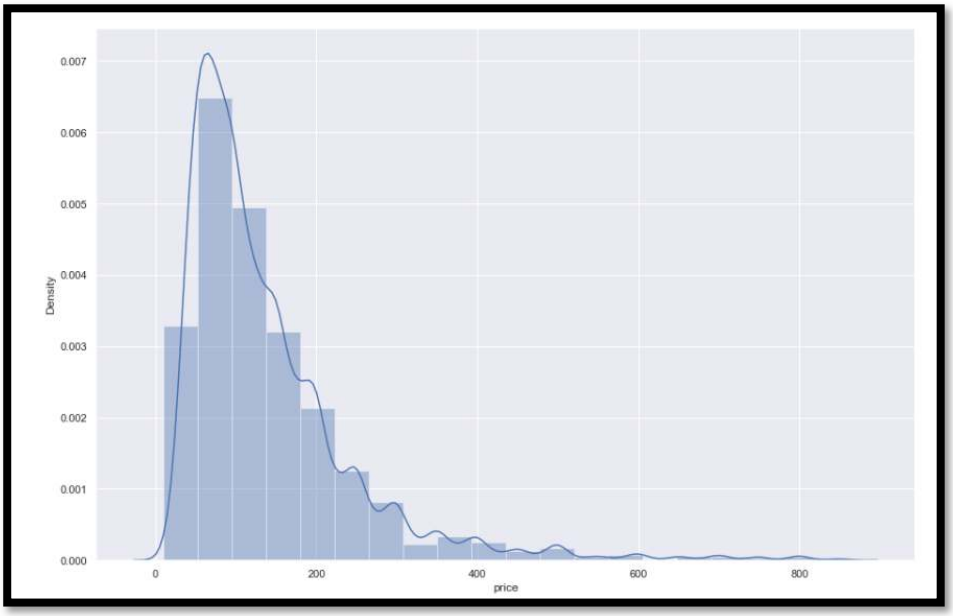
منهتن گران ترین و برانکس کم هزینه ترین مکان برای زندگی است

```
]: ### Lets see the average listing price by neighbourhood group
   ng_price=data1.groupby("neighbourhood_group")["price"].mean()

]: ### Manhattan is most expensive and Bronx is the least expensive place to live
   ng_price

]: neighbourhood_group
   Bronx             84.521140
   Brooklyn         116.576783
   Manhattan        174.867942
   Queens            94.104779
   Staten Island     96.148649
   Name: price, dtype: float64
```

Distribution data

توزیع قیمت را بررسی کنیم :

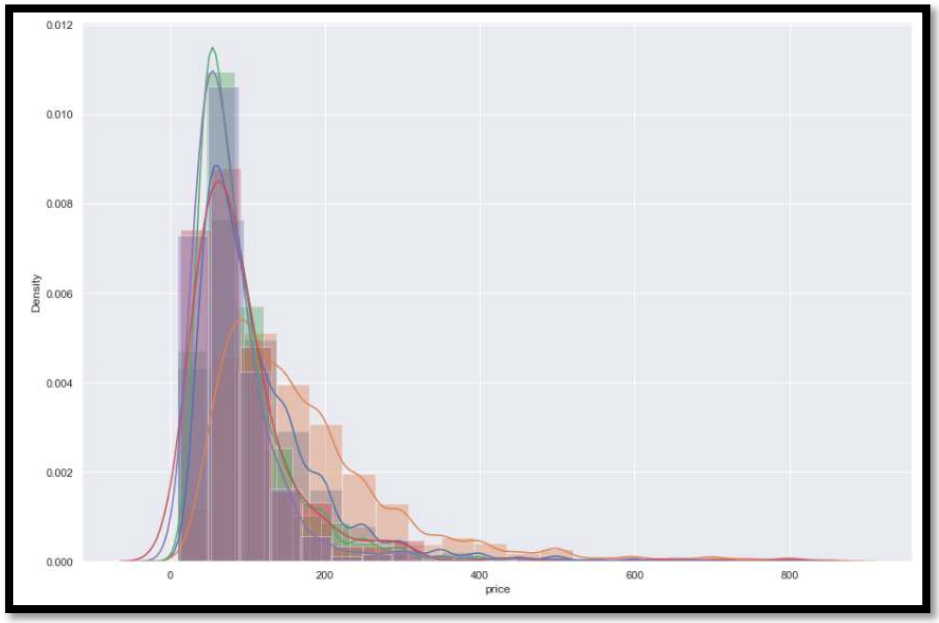
```
sns.set(rc={'figure.figsize(۱۵،۱۰):'})

sns.distplot(data1['price'],kde_kws={"label": 'price'}, bins=20)
```



همانطور که در شکل می بینیم توزیع نرمال نیست و باید روی این data تغییرات بدهیم تا دیتا نرمال گردد

توزیع را برای هر همسایه هم بررسی می کنیم :

```
})(۱۵،۱۰):'sns.set(rc={'figure.figsize

:()for groups in data1.neighbourhood_group.unique


sns.distplot(data1.price[data1['neighbourhood_group']==groups],kde_kws={"labe
l": groups}, bins=20)
```



همانطور که مشاهده می کنید اینجا هم توزیع نرمال نیست و ما چولگی داریم

برای اینکه بتوانیم data را نزدیک به نرمال کنیم از transfer ها استفاده می کنیم

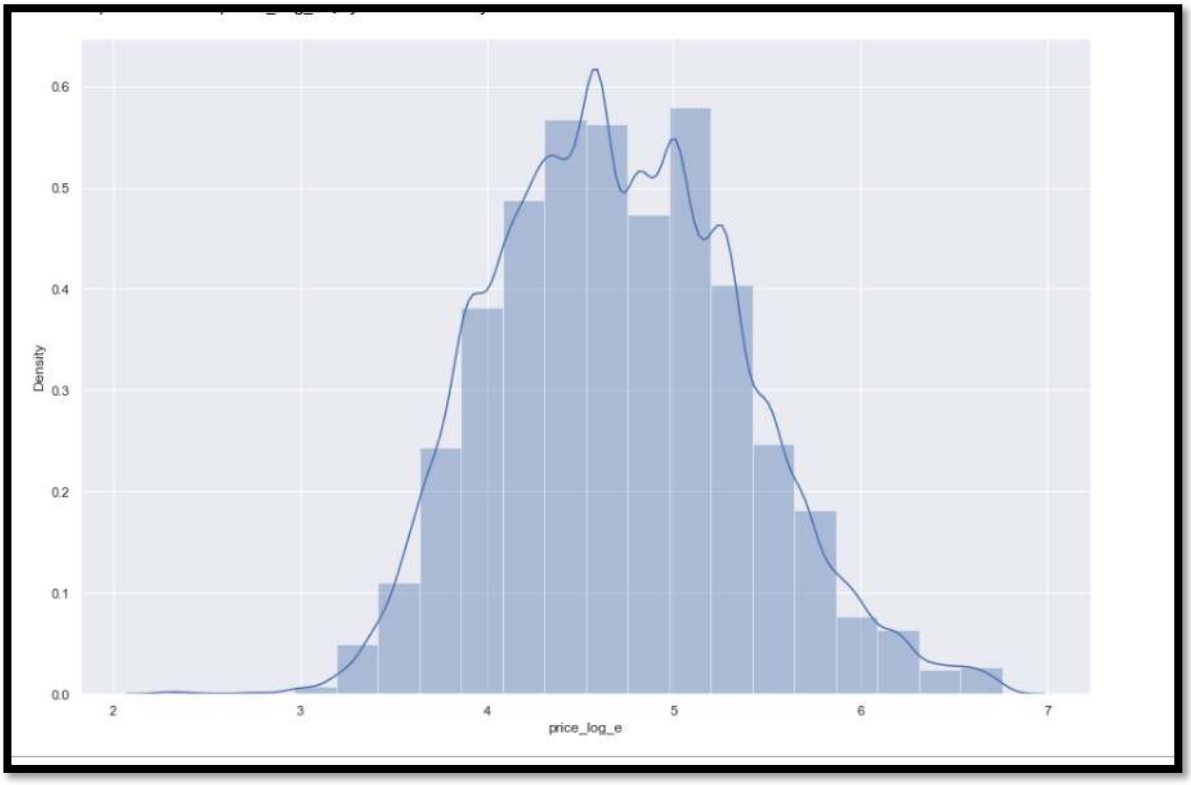
به طورمثال می توان از price، $\log_e$ گرفت . بنابراین یک فیچر جدیدی تعریف می کنیم به نام price_log_e که همان price ما هست و از آن log گرفتیم در پایه e

```
data1['price_log_e'] = np.log(data1['price'])

data1.head(۵)
```

با اعمال کد فوق ، ۵ ردیف اول به شرح ذیل می باشد :

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_rev
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	

```
sns.set(rc={'figure.figsize(۱۵،۱۰):'})

sns.distplot(data1['price_log_e'],kde_kws={"label": 'price in log e'}, bins=20)
```



همانطور که مشاهده می کنیم تقریبا دیتا نرمال سازی شد

از کجا مطمئن شوم که نرمال سازی شده است ؟با آزمون فرض -- Normal test

```
stats.normaltest(data1["price_log_e"])
```

```
NormaltestResult(statistic=562.3974954306634, pvalue=7.532437713364285e-123)
```

میزان pvalue بسیار کم است (نزدیک به صفر هست) و اینطوری فرض نرمال سازی ما رد می شود اما برای ادامه با مقداری خطا آن را نرمال فرض می کنیم .

برای نرمال سازی بهتر از log10 استفاده می کنیم و یک فیچر جدید تحت عنوان price_ log_10 ایجاد می کنیم و

```
data1['price_log_10'] = np.log10(data1['price'])
```

```
data1.head(۵)
```

پیرو این دستور ۵ ردیف اول به شرح ذیل خواهد بود.

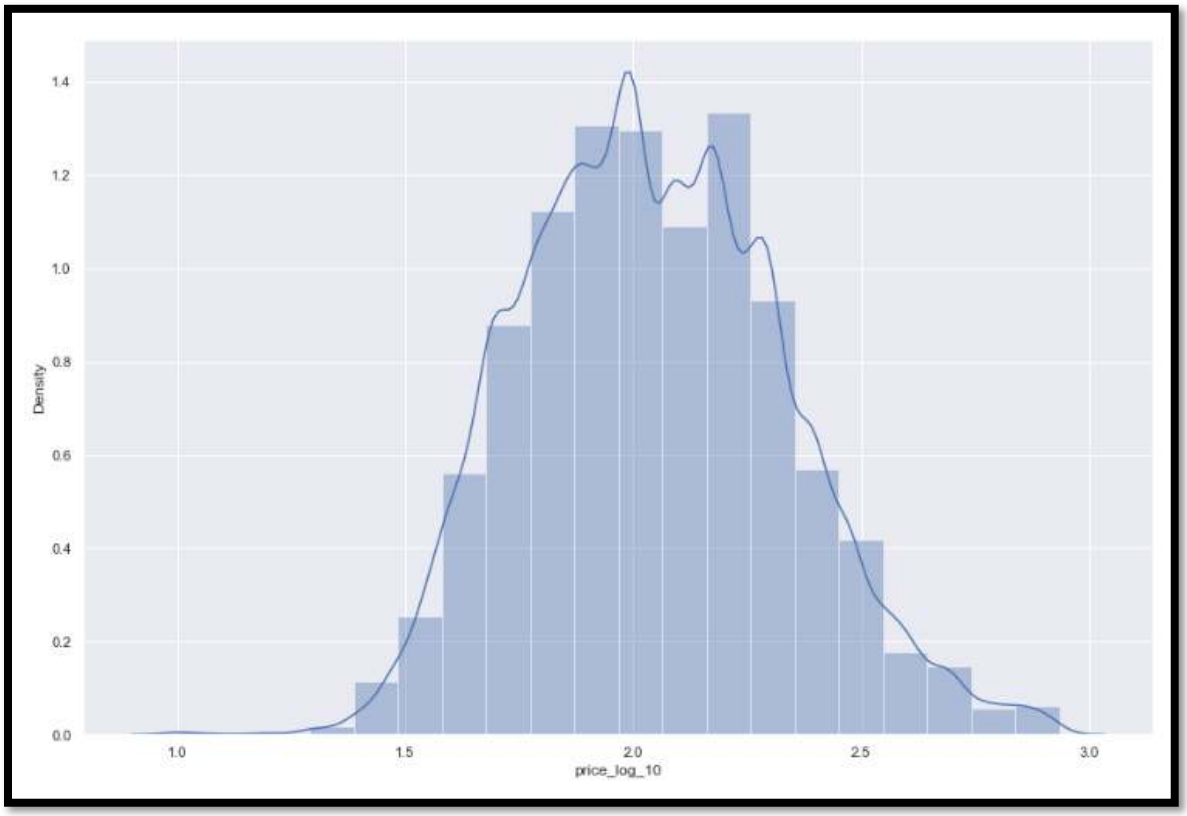
:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	r
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	

از طریق دستور seaborn نمودار می کشیم

```
sns.set(rc={'figure.figsize(۱۵،۱۰):'})
```

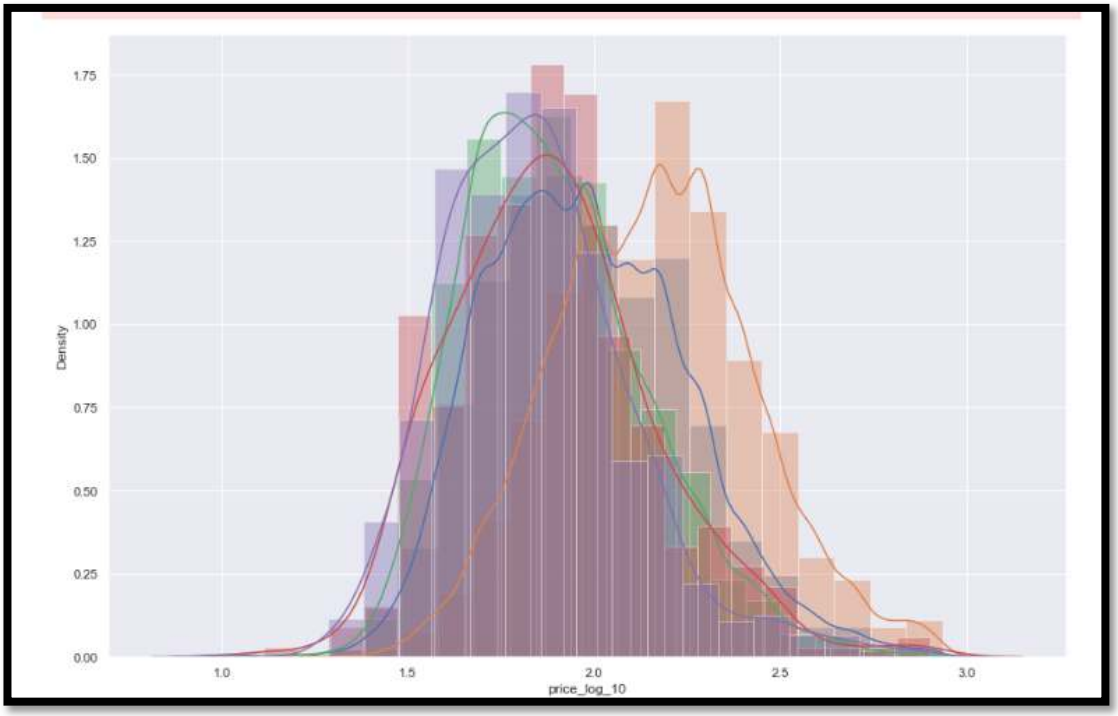
```
sns.distplot(data1['price_log_10'],kde_kws={"label": 'price in log 10'}, bins=20)
```



آزمون نرمال تست : با توجه به میزان pvalue ، price_ log_10 به دیتای نرمال نزدیک تر هست

```
[235]: stats.normaltest(data1["price_log_10"])
t[235]: NormaltestResult(statistic=562.3974954306705, pvalue=7.532437713337645e-123)
```

دیتاهای مختلف را اینجا plot می کنیم

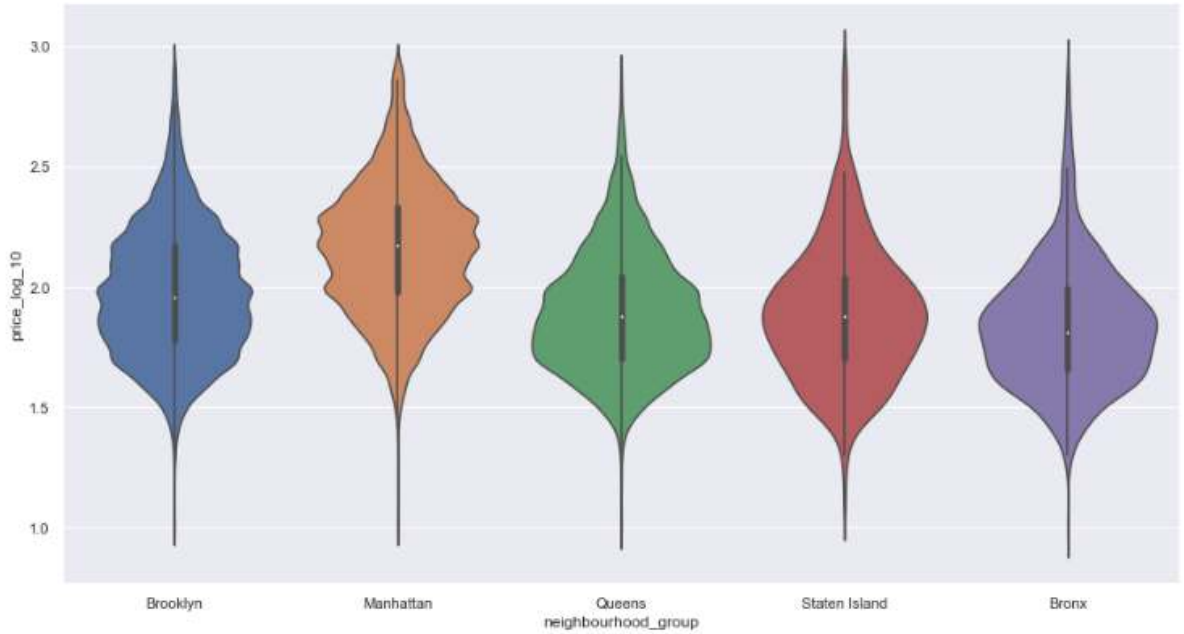


آیا رابطه ای بین price_ log_10 و مناطق پنج گانه هست ؟

```
plt.figure(figsize=(15,8))
sns.violinplot("neighbourhood_group", "price_log_10", data=data1)
```

```
D:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
s as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='neighbourhood_group', ylabel='price_log_10'>
```

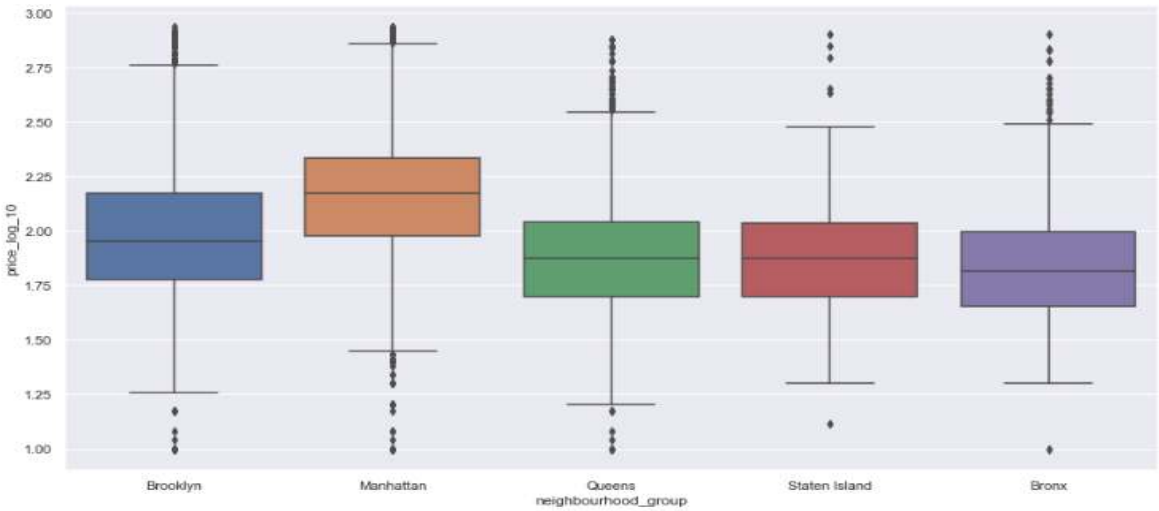


نمودار violinplot می کشیم . نموداریکه در آن mean, max مشخص هست

```
plt.figure(figsize=(15,8))
sns.boxplot("neighbourhood_group", "price_log_10", data=data1)
```

```
D:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
s as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='neighbourhood_group', ylabel='price_log_10'>
```



نمودار plotbox کشیده ایم و سپس از anovatestاستفاده می کنیم

Anova test

```
fstat, pval = stats.f_oneway(*[data1.price_log_10[data1.neighbourhood_group == s]
for s in data1.neighbourhood_group.unique()])
print("Oneway Anova log10(price) ~ neighbourhood_group F=%.2f, p-value=%E" % (fstat, pval))
```

```
Oneway Anova log10(price) ~ neighbourhood_group F=1926.63, p-value=0.000000E+00
```

```
data1[["neighbourhood_group",'price']].groupby("neighbourhood_group").describe()
```

	price							
	count	mean	std	min	25%	50%	75%	max
neighbourhood_group								
Bronx	1088.0	84.521140	72.677670	10.0	45.0	65.0	99.0	800.0
Brooklyn	20011.0	116.576783	88.430146	10.0	60.0	90.0	149.0	860.0
Manhattan	21377.0	174.867942	121.627249	10.0	95.0	149.0	215.0	860.0
Queens	5650.0	94.104779	69.190842	10.0	50.0	75.0	110.0	750.0
Staten Island	370.0	96.148649	84.022175	13.0	50.0	75.0	109.0	800.0

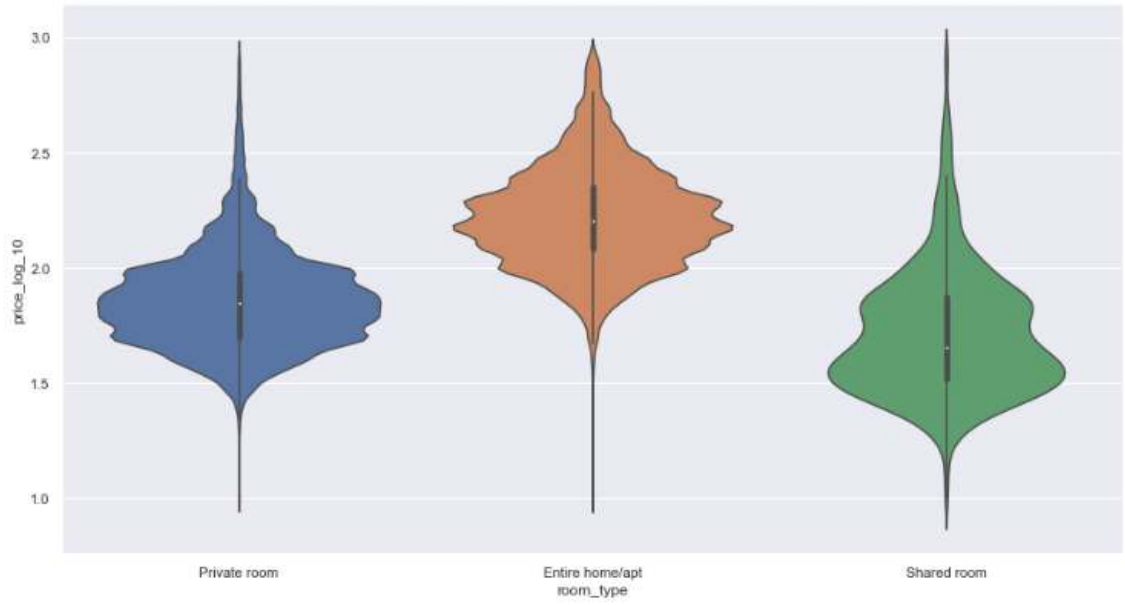
باتوجه به اینکه pvalue صفر شده است فرض خلف ما رد شده و فرض اصلی تایید می گردد . پس رابطه ای وجود دارد و منهتن بیشترین درآمد را داشته است

آیا رابطه ای بین نوع خانه و قیمت هست ؟

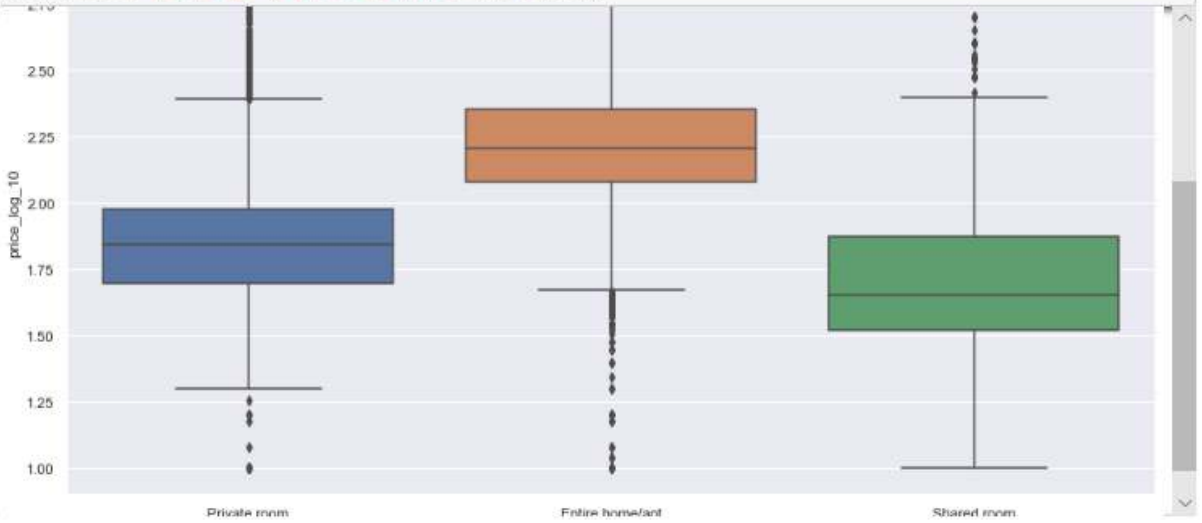
```
plt.figure(figsize=(15,8))
sns.violinplot("room_type", "price_log_10", data=data1)
```

```
D:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
s as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='room_type', ylabel='price_log_10'>
```



```
plt.figure(figsize=(15,8))
sns.boxplot("room_type", "price_log_10", data=data1)
```




```
fstat, pval = stats.f_oneway(*[data1.price_log_10[data1.room_type == s]
for s in data1.room_type.unique()])
print("Oneway Anova log10(price) ~ room_type F=%.2f, p-value=%E" % (fstat, pval))
```

```
Oneway Anova log10(price) ~ room_type F=17381.79, p-value=0.000000E+00
```

باتوجه به اینکه pvalue صفر شده است فرض خلف ما رد شده و فرض اصلی تایید می گردد . پس رابطه ای وجود دارد براساس نوع خانه مبلغ متفاوت هست خانه مستقل گرانترین و اتاق خصوصی میانه و اتاق اشتراکی کمترین هزینه را دارد

فراوانی اتاق ها را محاسبه کنید؟

```
room = data1.groupby('room_type')['id'].agg(['count'])
room.head()
```

room_type	count
Entire home/apt	25100
Private room	22242
Shared room	1154

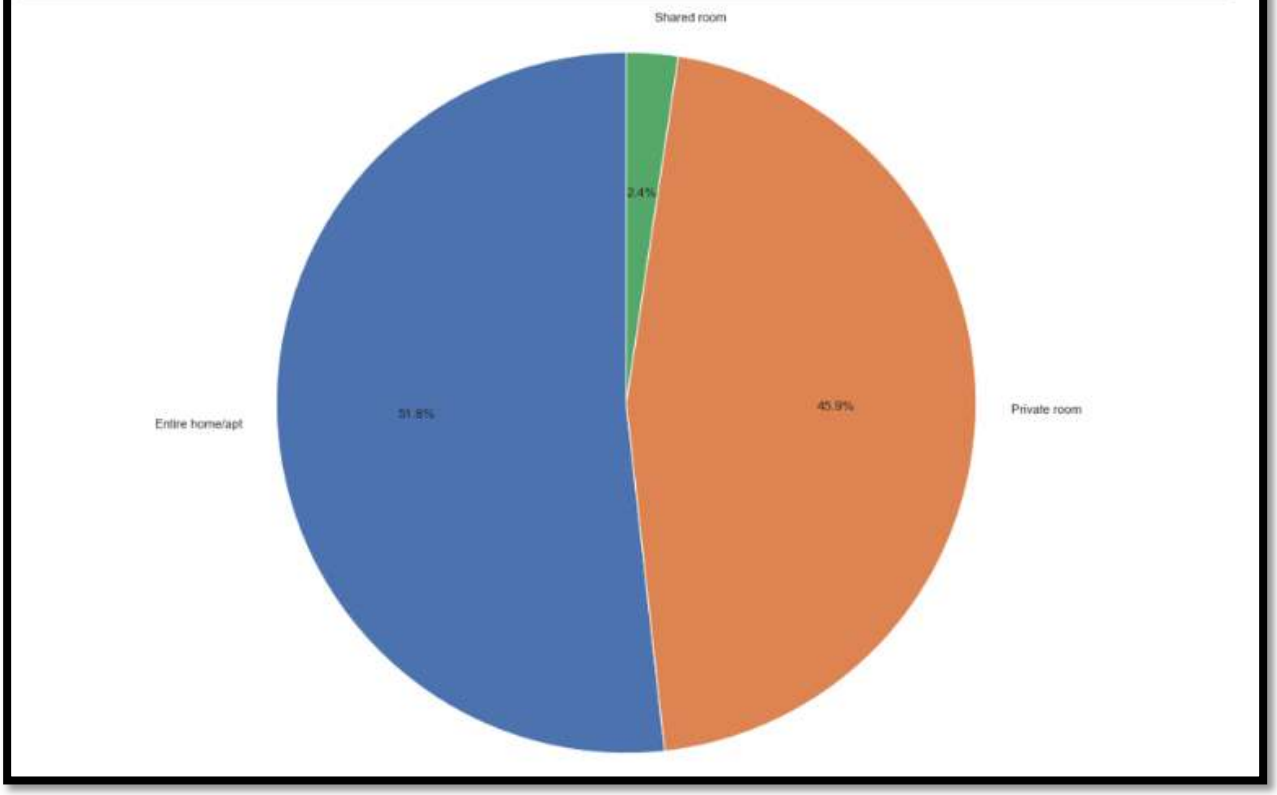
```
room.reset_index(level=0, inplace=True)
room.head()
```

	room_type	count
0	Entire home/apt	25100
1	Private room	22242
2	Shared room	1154

```
room = room[['room_type','count']]
```

نمودار خانه ها :

```
plt.pie(
    room['count'],
    labels=room['room_type'],
    shadow=False,
    startangle=90,
    autopct='%1.1f%%',
    )

plt.axis('equal')

plt.tight_layout()
plt.show()
```



چند نوع خانه داریم ؟

همان طور که مشخص است سه نوع

آپارتمانی یا سوئیت

خصوصی

عمومی

```
In [161]: room = data1.groupby('room_type')['id'].agg(['count'])
          room.head()
```

Out[161]:

room_type	count
Entire home/apt	25100
Private room	22242
Shared room	1154

- تعدادخانه ها ؟

	room_type	count
0	Entire home/apt	25100
1	Private room	22242
2	Shared room	1154

- تعداد خانه در manhatan چند تا هست ؟

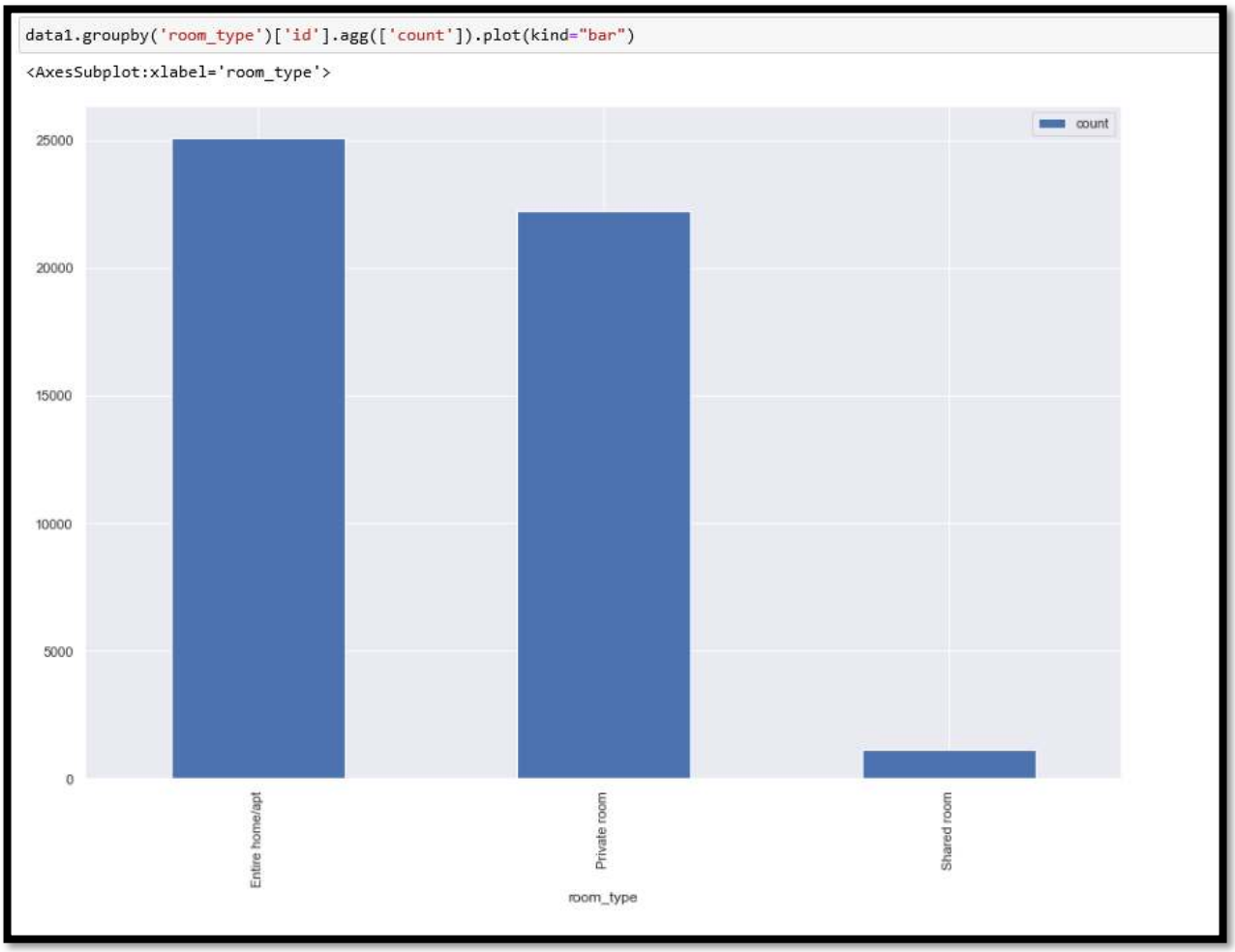
```
temp = data1[data1['neighbourhood_group'] == 'Manhattan']
temp['room_type'].value_counts()
```

```
Entire home/apt    12965
Private room        7933
Shared room          479
Name: room_type, dtype: int64
```

تعداد خانه ها بر اساس host_id مشخص کنید ؟

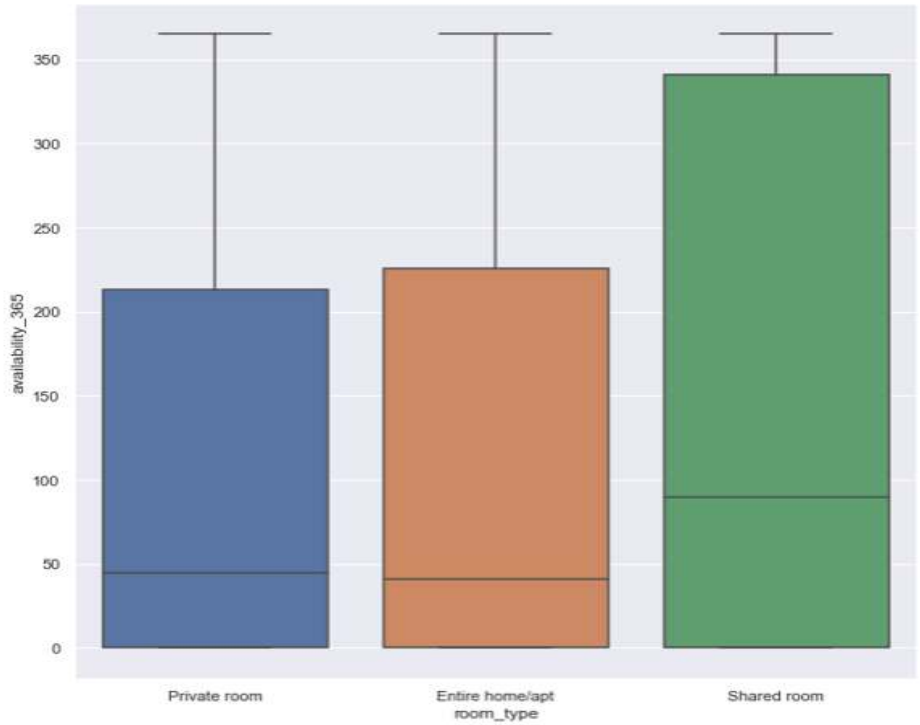
```
data1.groupby('room_type')['id'].agg(['count']).plot(kind="bar")
```

```
<AxesSubplot:xlabel='room_type'>
```



Room type vs availability

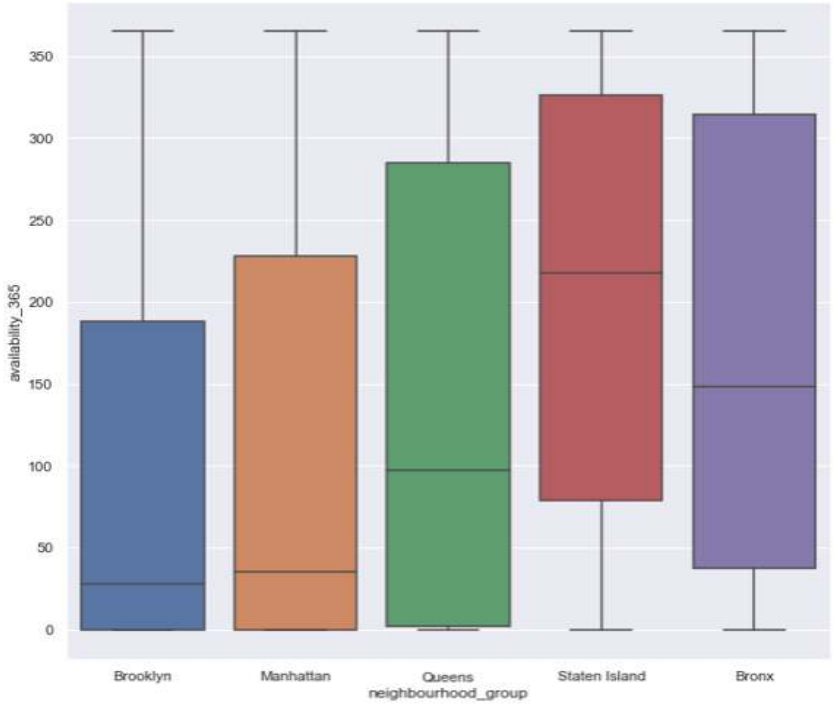
```
]: plt.figure(figsize=(10,10))
   ax = sns.boxplot(data=data1, x='room_type',y='availability_365')
```



جالبه که اتاق اشتراکی در طول سال بیشترین میزان را دارد .

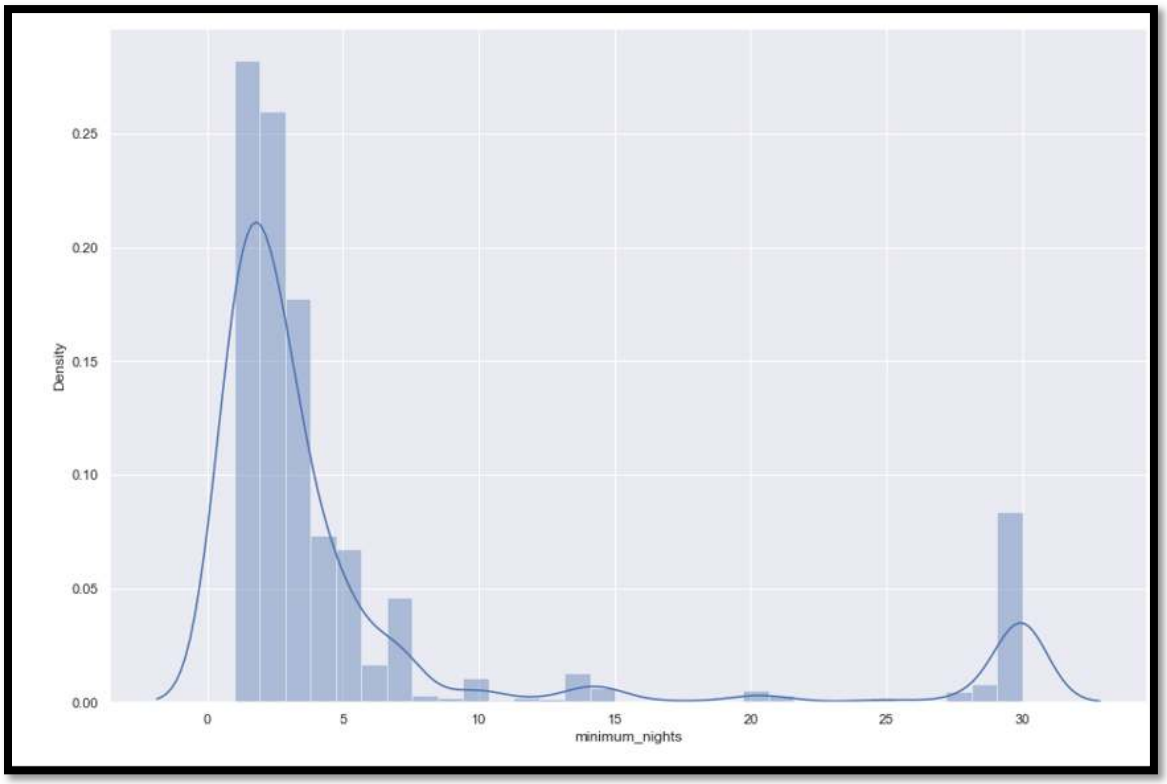
Neighbourhood vs availability

```
n [245]: plt.figure(figsize=(10,10))
         ax = sns.boxplot(data=data1, x='neighbourhood_group',y='availability_365')
```



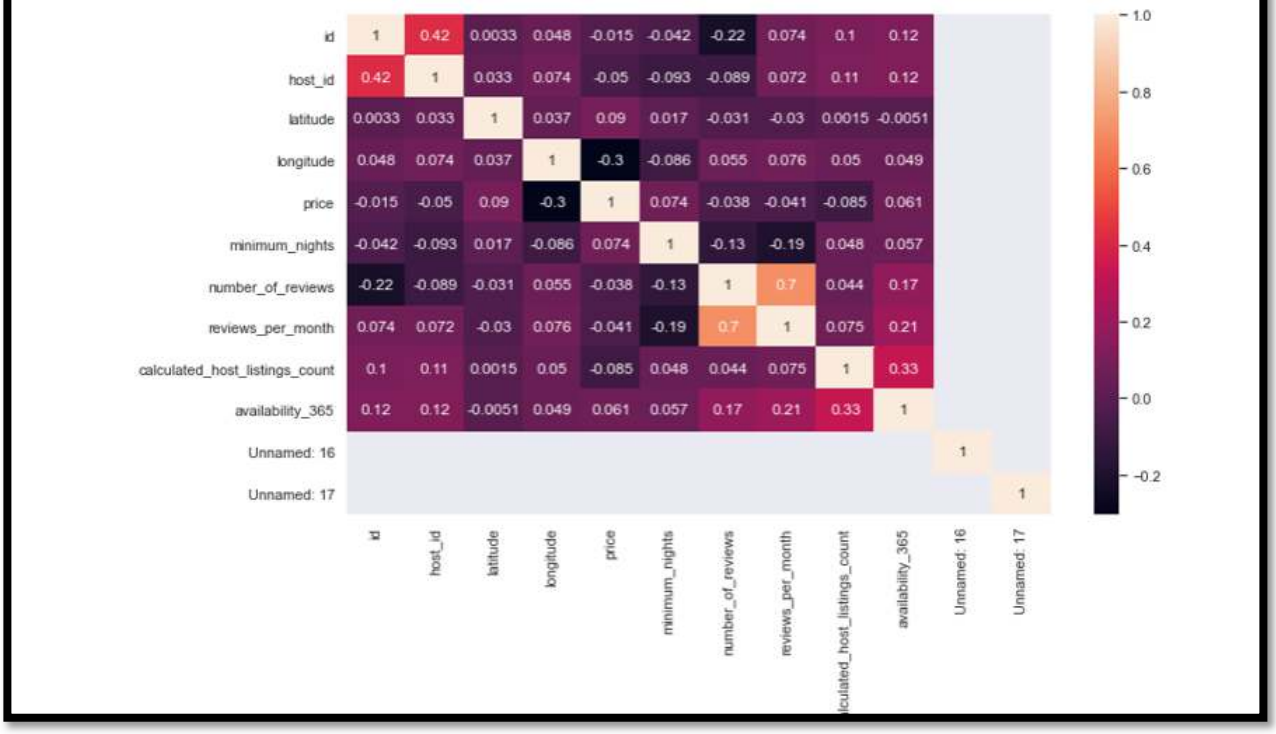
شب های رزرو شده به چه صورت است ؟

```
sns.distplot(df[(df['minimum_nights'] <= 30) & (df['minimum_nights'] > 0)]['minimum_nights'], bins=31)

plt.ioff()
```



Correlation بین فیچرهای مختلف

```
In [176]: #Get correlation between different features
          corr = data.corr(method='kendall')
          plt.figure(figsize=(12,7))
          sns.heatmap(corr, annot=True)
          data.columns

Out[176]: 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', 'Unnamed: 16', 'Unnamed: 17'],
                dtype='object')
```



- What can we learn from predictions? (ex: locations, prices, reviews, etc)

بررسی اطلاعات reviews_per_month

```
In [178]: data1.reviews_per_month.describe()

Out[178]: count    48496.000000
          mean         1.095951
          std          1.600386
          min          0.000000
          25%          0.040000
          50%          0.380000
          75%          1.600000
          max         58.500000
          Name: reviews_per_month, dtype: float64
```

```
In [115]: temp = data1[data1['reviews_per_month']>0]
          temp.head(5)
```

Out[115]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	number_of_reviews	last_review	re
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	...	9	10/19/2018	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	...	45	5/21/2019	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	...	270	7/5/2019	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	...	9	11/19/2018	
5	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray Hill	40.74767	-73.97500	Entire home/apt	200	...	74	6/22/2019	

5 rows × 21 columns

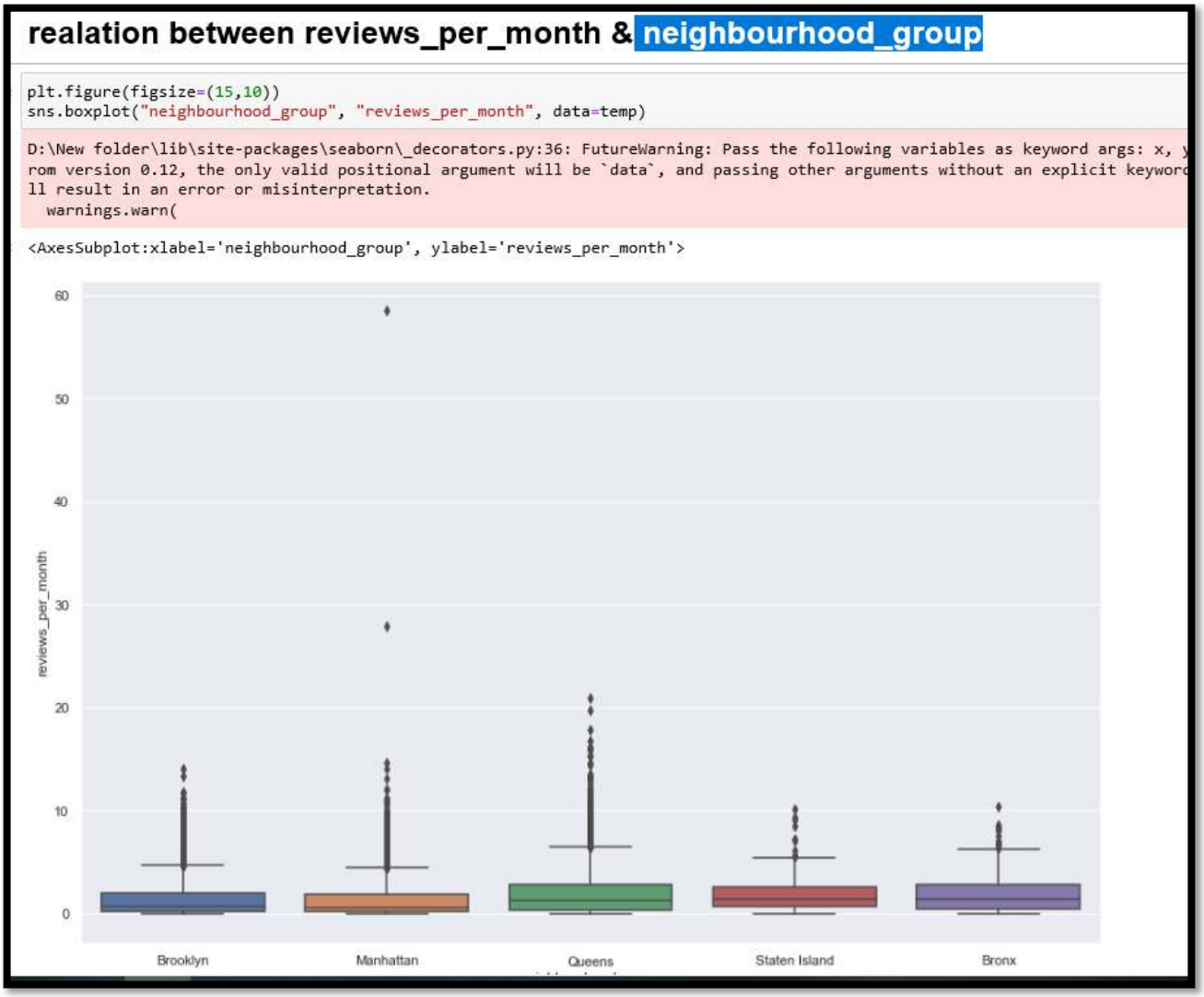
آیا رابطه ای بین reviews_per_month و neighbourhood_group هست ؟

realation between reviews_per_month & neighbourhood_group

```
plt.figure(figsize=(15,10))
sns.boxplot("neighbourhood_group", "reviews_per_month", data=temp)
```

```
D:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y
rom version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword
ll result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='neighbourhood_group', ylabel='reviews_per_month'>
```



```
fstat, pval = stats.f_oneway(*[temp.reviews_per_month[temp.neighbourhood_group == s]
for s in temp.neighbourhood_group.unique()])
print("Oneway Anova reviews_per_month ~ neighbourhood_group F=%.2f, p-value=%E" % (fstat, pval))
```

```
Oneway Anova reviews_per_month ~ neighbourhood_group F=182.84, p-value=1.674033E-155
```

ما تا اینجا متوجه شدیم که manhatan و brooklin بالاترین لیست با قیمت بالای ۱۵۰ دلار را دارا هستند و همین طور قیمت های بالای ۱۰۰ دلار کل خانه ، اتاق خصوصی هست و کمترین مقدار اتاق مشترک است

حالا در اینجا می خواهیم ماکسیمم host _id را پیدا کنیم .

df2 = data1.groupby(["host_id"])

max(df2.size())

حالا پنج عنصر ابتدایی و انتهایی را فرا می خوانیم

```
## Here we can see that 32K host_ids are unique appearing only once whereas some host_ids appea
df2.size().value_counts().head()
```

```
1    32062
2     3310
3      945
4      357
5      167
dtype: int64
```

```
df2.size().value_counts().tail()
```

```
50    1
49    1
16    1
43    1
32    1
dtype: int64
```

```
### Finding the host_id with maximum listings
host_id_counts = data1["host_id"].value_counts()
max_host = host_id_counts.idxmax()
max_host
```

```
219517861
```

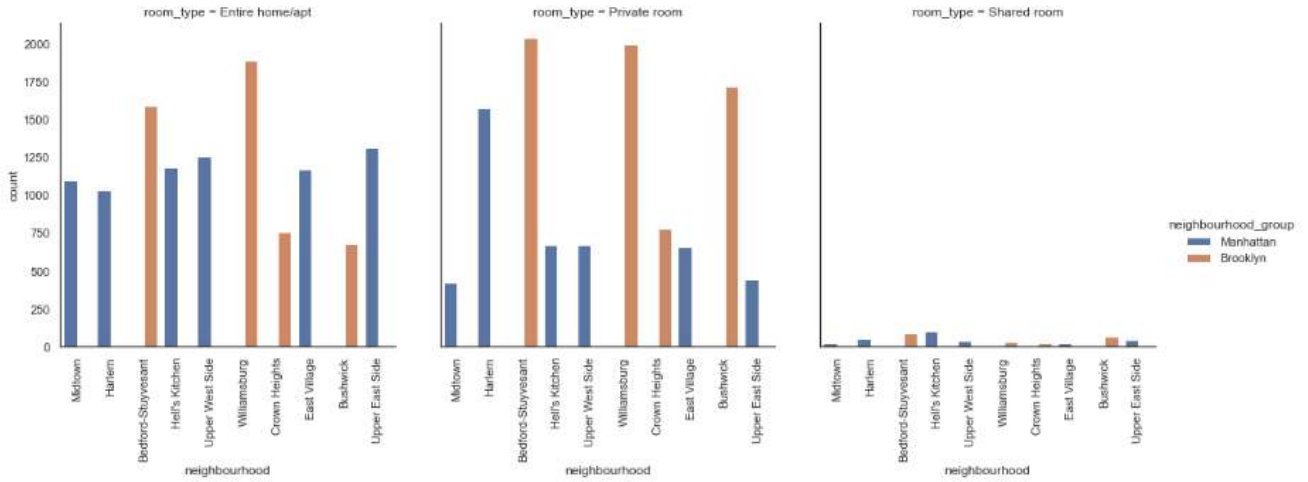
۱۰ محله برتر که بیشترین لیست را دارند کدامند ؟

```
#finding out top 10 neighbourhoods
df.neighbourhood.value_counts().head(10)
```

```
Williamsburg            3920
Bedford-Stuyvesant      3714
Harlem                  2658
Bushwick                2465
Upper West Side         1971
Hell's Kitchen          1958
East Village            1853
Upper East Side         1798
Crown Heights           1564
Midtown                 1545
Name: neighbourhood, dtype: int64
```

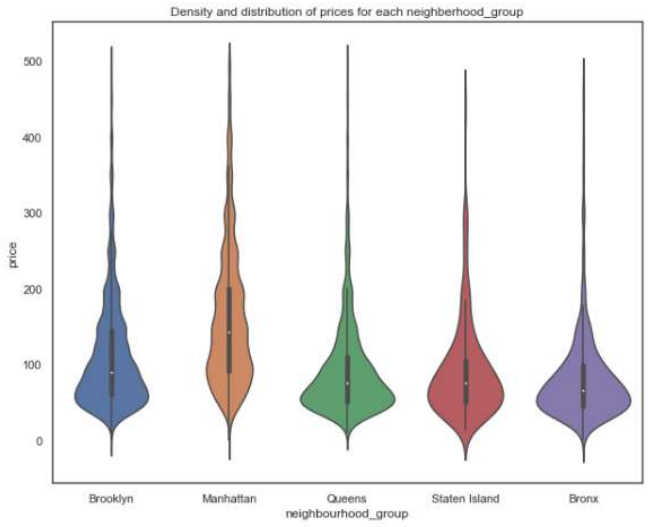
• Is there any noticeable difference of traffic among different areas and what could be the
reason for it?

```
In [137]: #let's now combine this with our boroughs and room type for a rich visualization we can make

          #grabbing top 10 neighbourhoods for sub-dataframe
          sub_7=df.loc[df['neighbourhood'].isin(['Williamsburg','Bedford-Stuyvesant','Harlem','Bushwick',
                         'Upper West Side','Hell\'s Kitchen','East Village','Upper East Side','Crown Heights','Midtown'])]
          #using catplot to represent multiple interesting attributes together and a count
          viz_3=sns.catplot(x='neighbourhood', hue='neighbourhood_group', col='room_type', data=sub_7, kind='count')
          viz_3.set_xticklabels(rotation=90)

Out[137]: <seaborn.axisgrid.FacetGrid at 0x162de5eb8b0>
```



مشاهداتی که قطعاً بیشترین تضاد را دارند این است که لیست Airbnb از نوع "Shared room" به سختی در میان ۱۰ محله پرجمعیت موجود است. سپس ، می توانیم ببینیم که برای این ۱۰ محله فقط ۲ بخش ارائه شده است: منهتن و بروکلین. تا حدودی انتظار می رفت چون منهتن و بروکلین یکی از پر مسافرت ترین مقصد هستند ، بنابراین بیشترین لیست را در دسترس خواهند داشت. همچنین می توانیم مشاهده کنیم که بدفورد-استویوزانت و ویلیامزبورگ محبوب ترین شهرها برای بخش منهتن و هارلم برای بروکلین هستند.

```
In [135]: #we can see from our statistical table that we have some extreme values, therefore we need to remove them for the sake of a bette

          #creating a sub-dataframe with no extreme values / less than 500
          sub_6=df[df.price < 500]
          #using violinplot to showcase density and distribtuion of prices
          viz_2=sns.violinplot(data=sub_6, x='neighbourhood_group', y='price')
          viz_2.set_title('Density and distribution of prices for each neighberhood_group')

Out[135]: Text(0.5, 1.0, 'Density and distribution of prices for each neighberhood_group')
```



با داشتن یک جدول آماری و یک طرح ویولن ، ما قطعاً می توانیم چند مورد را در مورد توزیع قیمت Airbnb در بخش های NYC مشاهده کنیم. اول ، می توانیم بگوییم که منهتن بالاترین طیف قیمت را برای لیست ها با قیمت ۱۵۰ دلار به عنوان مشاهده متوسط دارد و پس از آن بروکلین با ۹۰ دلار برای هر شب. کوینز و استاتن آیلند توزیع های بسیار مشابهی دارند ، برونکس ارزانترین از همه است. این توزیع و تراکم قیمت ها کاملاً انتظار می رفت. به عنوان مثال ، هیچ کس راز ندارد که منهتن یکی از گرانترین مکانهای زندگی در جهان است ، از طرف دیگر به نظر می رسد که برانکس دارای استاندارد زندگی پایین تر است

آیا رابطه ای میان host و مناطق پنج گانه هست ؟

• What can we learn about different hosts and areas?

```
In [146]: df[['neighbourhood','neighbourhood_group','host_id']].sort_values('neighbourhood',ascending =False).head(10)
```

Out[146]:

	neighbourhood	neighbourhood_group	host_id
37574	Woodside	Queens	192773562
33857	Woodside	Queens	19303369
44700	Woodside	Queens	259229690
44697	Woodside	Queens	259229690
35184	Woodside	Queens	137358866
44693	Woodside	Queens	259229690
40673	Woodside	Queens	137358866
1556	Woodside	Queens	3621183
35157	Woodside	Queens	210687004
24112	Woodside	Queens	136211138

```
In [147]: sns.boxplot("neighbourhood_group", "host_id", data=data1)
```

```
D:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. F
rom version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword wi
ll result in an error or misinterpretation.
  warnings.warn(
```

Out[147]: <AxesSubplot:xlabel='neighbourhood_group', ylabel='host_id'>