Airbnb price prediction

In [164]:
```
import pandas as pd
import numpy as np
data= pd.read_csv('train.csv')
data.head(5)
```

Out[164]:

	id	log_price	property_type	room_type	amenities	accommodates	bathrooms	bed_type
0	6901257	5.010635	Apartment	Entire home/apt	{"Wireless Internet","Air conditioning",Kitche...	3	1.0	Real Bed
1	6304928	5.129899	Apartment	Entire home/apt	{"Wireless Internet","Air conditioning",Kitche...	7	1.0	Real Bed
2	7919400	4.976734	Apartment	Entire home/apt	{TV,"Cable TV","Wireless Internet","Air condit...	5	1.0	Real Bed
3	13418779	6.620073	House	Entire home/apt	{TV,"Cable TV",Internet,"Wireless Internet",Ki...	4	1.0	Real Bed
4	3808709	4.744932	Apartment	Entire home/apt	{TV,Internet,"Wireless Internet","Air conditio...	2	1.0	Real Bed

5 rows × 29 columns

In [165]:
```
data.shape
```

Out[165]: (74111, 29)

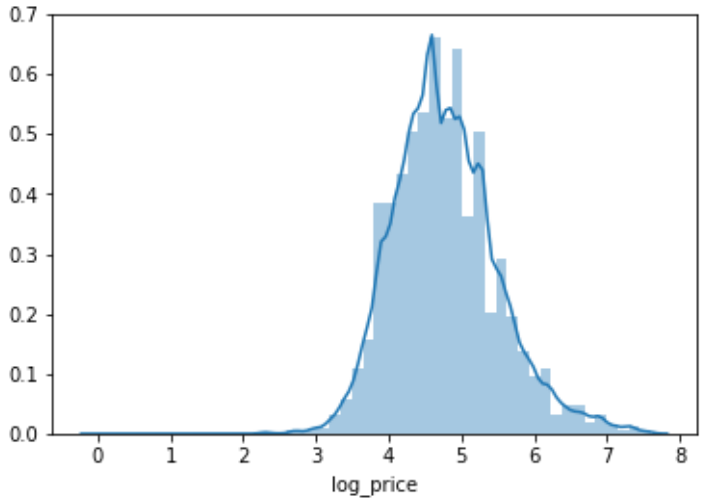
In [166]:
```
data.describe(include=['O'])
```

Out[166]:

	property_type	room_type	amenities	bed_type	cancellation_policy	city	description	first_review
count	74111	74111	74111	74111	74111	74111	74111	58247
unique	35	3	67122	5	5	6	73479	2554
top	Apartment	Entire home/apt	{}	Real Bed	strict	NYC	Hello, I've been running guest house for Korea...	2017-01-01
freq	49003	41310	586	72028	32374	32349	8	293

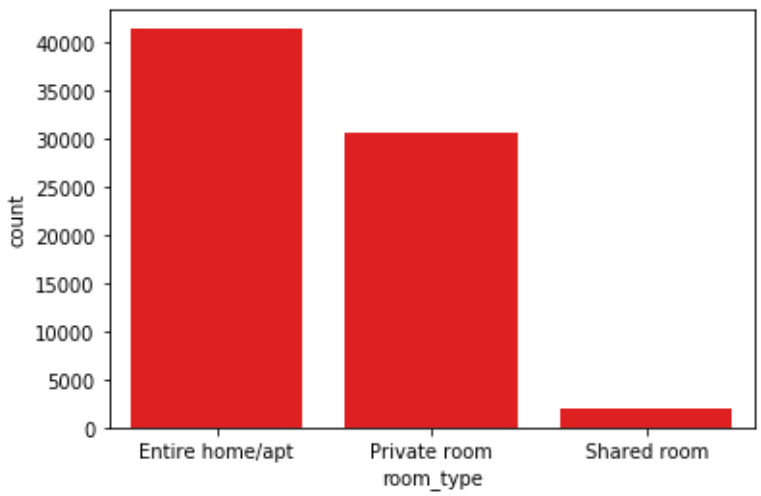
In [167]:
```
import seaborn as sns
import matplotlib.pyplot as plt
x= data.iloc[:,1]
sns.distplot(x);
```

```
C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Usi
ng a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(se
q)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `
arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



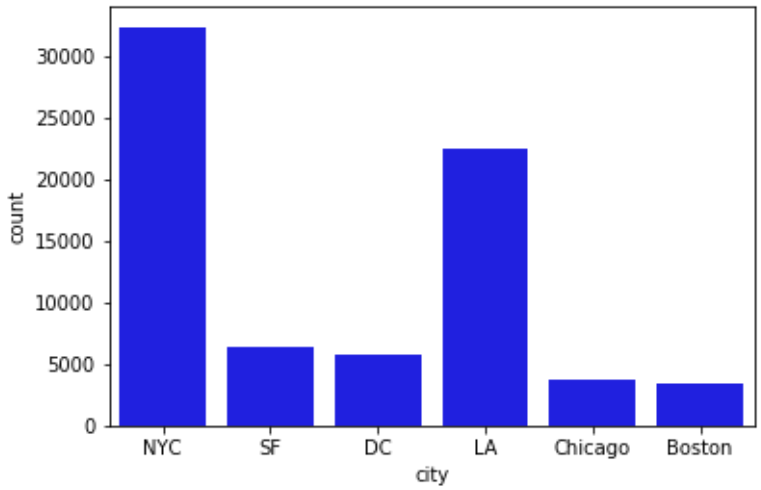
In [168]:
```
sns.countplot(x='room_type',color='red', data=data)
```

Out[168]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x202299b2f98>
```



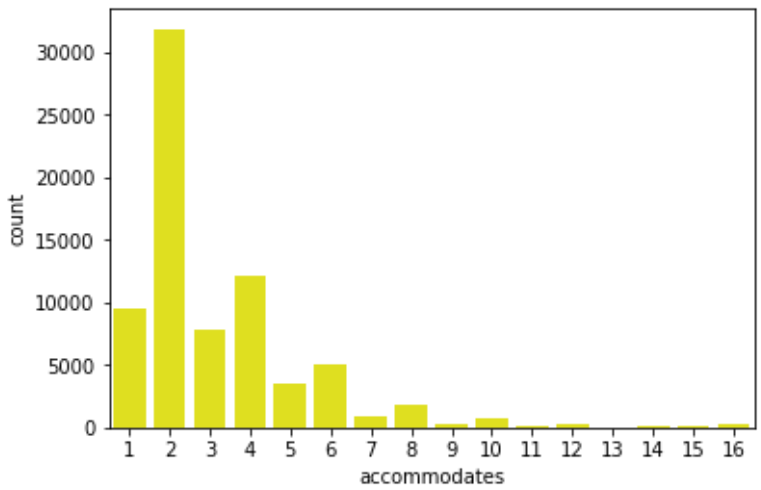
In [169]:
```
sns.countplot(x='city',color='blue', data=data)
```

Out[169]: <matplotlib.axes._subplots.AxesSubplot at 0x202299faeb8>



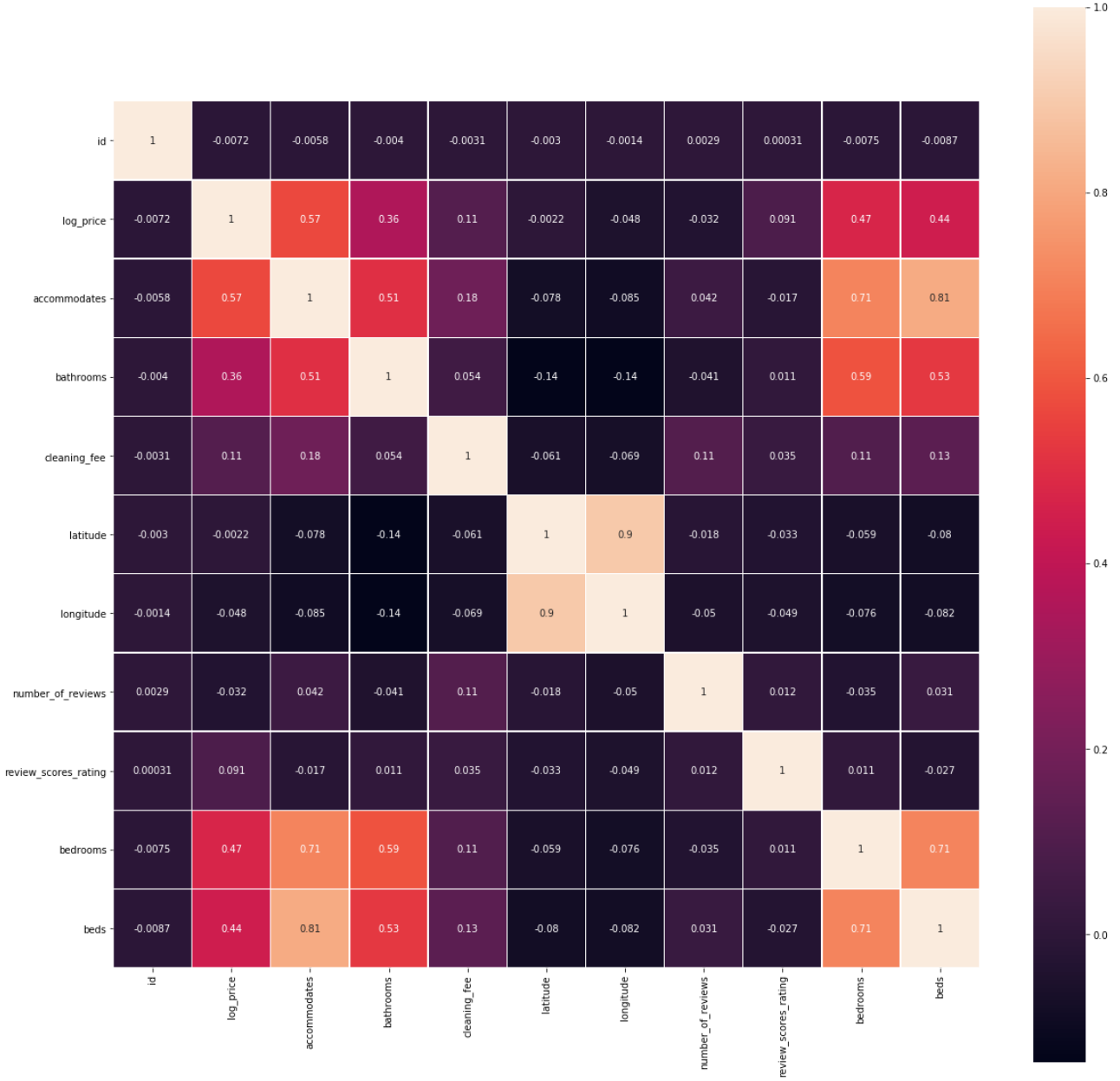
In [170]:
```
sns.countplot(x='accommodates',color='yellow', data=data)
```

Out[170]: <matplotlib.axes._subplots.AxesSubplot at 0x2022a21d0f0>



In [171]:
```
plt.figure(figsize=(20,20))
sns.heatmap(data.corr(),linewidths=0.25, square=True,
    linecolor='w',annot=True)
```

Out[171]: <matplotlib.axes._subplots.AxesSubplot at 0x2022a2292e8>



In [172]:
```
filedata=data
filedata=filedata.drop(['id','number_of_reviews','review_scores_rating','latitude',
       'longitude' ], axis=1)
```

In [173]:
```
filedata=filedata.drop(['bathrooms','bedrooms','beds'], axis=1)
```

In [174]:
```
for col in filedata:
    if (filedata[col].isnull().any()):
        print(col)
```

```
first_review
host_has_profile_pic
host_identity_verified
host_response_rate
host_since
last_review
neighbourhood
thumbnail_url
zipcode
```

In [175]:
```
for col in filedata:
    if (filedata[col].isnull().any()):
        filedata=filedata.drop([col],axis=1)
filedata
```

Out[175]:

	log_price	property_type	room_type	amenities	accommodates	bed_type	ca
0	5.010635	Apartment	Entire home/apt	{"Wireless Internet","Air conditioning",Kitche...	3	Real Bed	
1	5.129899	Apartment	Entire home/apt	{"Wireless Internet","Air conditioning",Kitche...	7	Real Bed	
2	4.976734	Apartment	Entire home/apt	{TV,"Cable TV","Wireless Internet","Air condit...	5	Real Bed	
3	6.620073	House	Entire home/apt	{TV,"Cable TV",Internet,"Wireless Internet",Ki...	4	Real Bed	
4	4.744932	Apartment	Entire home/apt	{TV,Internet,"Wireless Internet","Air conditio...	2	Real Bed	

In [176]:
```
filedata['bed_type'].value_counts()
```

Out[176]:
```
Real Bed         72028
Futon              753
Pull-out Sofa      585
Airbed             477
Couch              268
Name: bed_type, dtype: int64
```

In [177]:
```
filedata=filedata.drop(['property_type','amenities','description'], axis=1)
```

In [178]:
```
def replace_with_ohe(df, cat_columns):
    for col in cat_columns:
        df[col] = pd.Categorical(df[col])
        dum = pd.get_dummies(df[col], prefix=col)
        df=pd.concat([df, dum], axis=1)
    df = df.drop(cat_columns, axis=1)
    return df
```

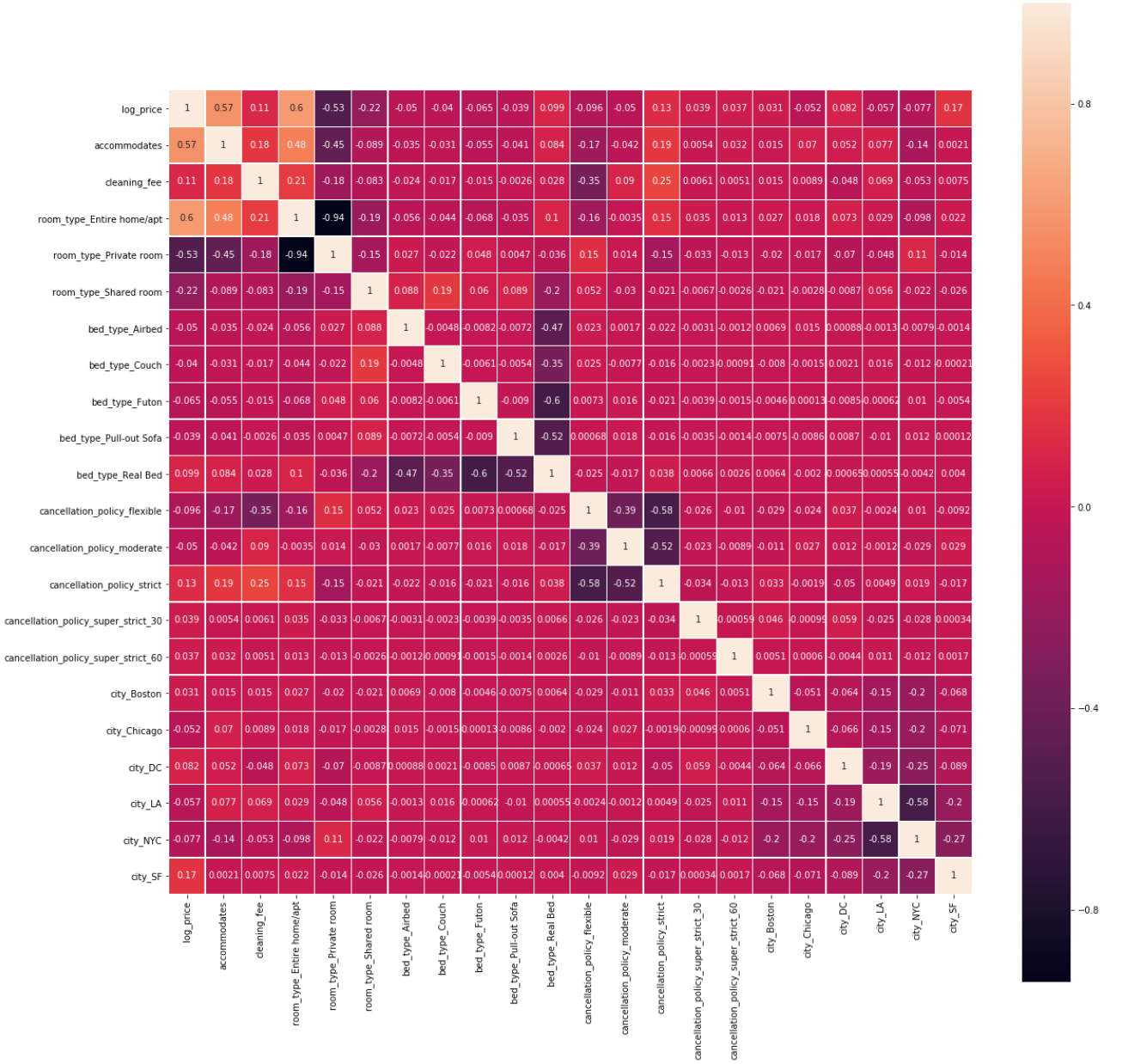
In [179]:
```
filedata = replace_with_ohe(filedata, ['room_type', 'bed_type', 'cancellation_policy', '
filedata
```

Out[179]:

	log_price	accommodates	cleaning_fee	instant_bookable	name	room_type_Entire home/apt	r
0	5.010635	3	True	f	Beautiful brownstone 1-bedroom	1	
1	5.129899	7	True	t	Superb 3BR Apt Located Near Times Square	1	
2	4.976734	5	True	t	The Garden Oasis	1	
3	6.620073	4	True	f	Beautiful Flat in the Heart of SF!	1	
4	4.744932	2	True	t	Great studio in midtown DC	1	
5	4.442651	2	True	t	Comfort Suite San Francisco	0	

In [180]:
```
plt.figure(figsize=(20,20))
sns.heatmap(filedata.corr(),linewidths=0.25, square=True,
    linecolor='w',annot=True)
```

Out[180]: <matplotlib.axes._subplots.AxesSubplot at 0x2022a2c94a8>



In [181]:
```
filedata['instant_bookable'] = (filedata['instant_bookable'] == 't').astype(int)
filedata['instant_bookable']
```

Out[181]:
```
0     0
1     1
2     1
3     0
4     1
5     1
6     1
7     0
8     0
9     1
10    0
11    0
12    0
13    1
14    0
15    0
16    0
17    0
18    0
```

In [182]:
```
X = filedata.iloc[:,1:].drop(['name'], axis=1)
```

In [183]:
```
Y = filedata['log_price']
```

In [184]:
```
X
```

Out[184]:

	accommodates	cleaning_fee	instant_bookable	room_type_Entire home/apt	room_type_Private room	room_type_
0	3	True	0	1	0	
1	7	True	1	1	0	
2	5	True	1	1	0	
3	4	True	0	1	0	
4	2	True	1	1	0	
5	2	True	1	0	1	
6	3	True	1	1	0	
7	2	True	0	1	0	
8	2	True	0	0	1	
9	2	True	1	0	1	

In [185]:
```
Y
```

Out[185]:
```
0        5.010635
1        5.129899
2        4.976734
3        6.620073
4        4.744932
5        4.442651
6        4.418841
7        4.787492
8        4.787492
9        3.583519
10       4.605170
11       5.010635
12       4.248495
13       5.298317
14       4.955827
15       4.094345
16       4.317488
17       4.595120
18       4.882802
```

In [186]:
```
from sklearn.model_selection import train_test_split
```

In [187]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y)
```

In [188]:
```
X_train
```

Out[188]:

	accommodates	cleaning_fee	instant_bookable	room_type_Entire home/apt	room_type_Private room	room_type_
57252	2	True	0	0	1	
18035	4	True	0	1	0	
24338	2	True	1	0	1	
44248	2	False	1	1	0	
51230	2	True	0	0	1	
60244	2	True	0	0	1	
36415	2	True	1	0	1	
15248	3	True	1	1	0	
31263	3	True	0	1	0	
52723	2	True	0	1	0	

Linear Regression

In [203]:
```
from sklearn.linear_model import LinearRegression
lg = LinearRegression()
lg.fit(X_train,Y_train)
print("accuracy= ",lg.score(X_test, Y_test))
```

```
accuracy=  0.5231020629751489
```

Random forest regression

In [204]:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(X_train, Y_train)
print("accuracy= ",rf.score(X_test, Y_test))
```

```
C:\Users\user\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246: FutureWarnin
g: The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.2
2.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)


accuracy=  0.532491983017879
```