```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df=pd.read_csv('/content/Housing.csv')
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwate
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	

```
df.tail()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwat
540	1820000	3000	2	1	1	yes	no	yes	
541	1767150	2400	3	1	1	no	no	no	
542	1750000	3620	2	1	1	yes	no	no	
543	1750000	2910	3	1	1	no	no	no	
544	1750000	3850	3	1	2	yes	no	no	

```
df.describe()
```

	price	area	bedrooms	bathrooms	stories	mainroad	gu
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000	545
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.858716	0
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.348635	0
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000	0
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	1.000000	0
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	1.000000	0
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000	0
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	1.000000	1

```
df.info()
```

```
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    int64
 6   guestroom         545 non-null    int64
 7   basement          545 non-null    int64
 8   hotwaterheating   545 non-null    int64
 9   airconditioning   545 non-null    int64
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    int64
 12  furnishingstatus  545 non-null    int64
dtypes: int64(13)
memory usage: 55.5 KB
```

```
df.isna().sum()
```

```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()

lst=['mainroad',  'guestroom',  'basement',   'hotwaterheating',  'airconditioning', 'prefa

for i in lst:
  df[i]=lb.fit_transform(df[i])
```

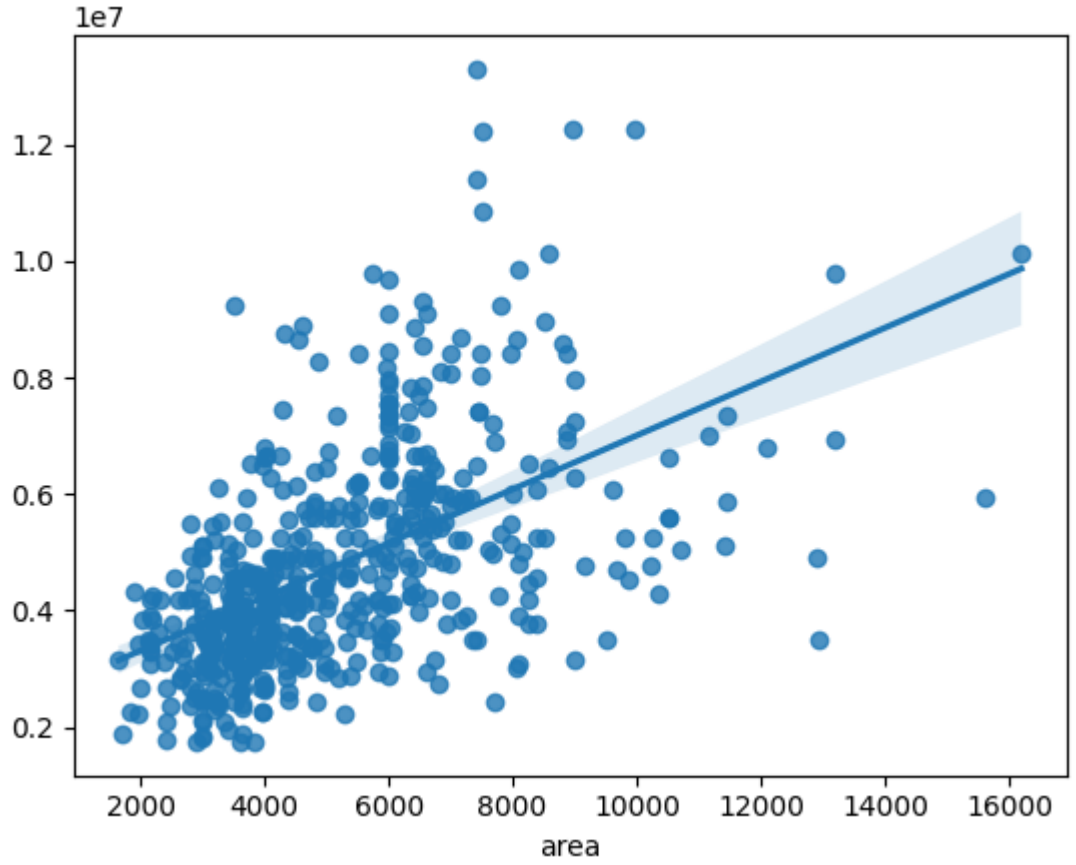
```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwate
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	

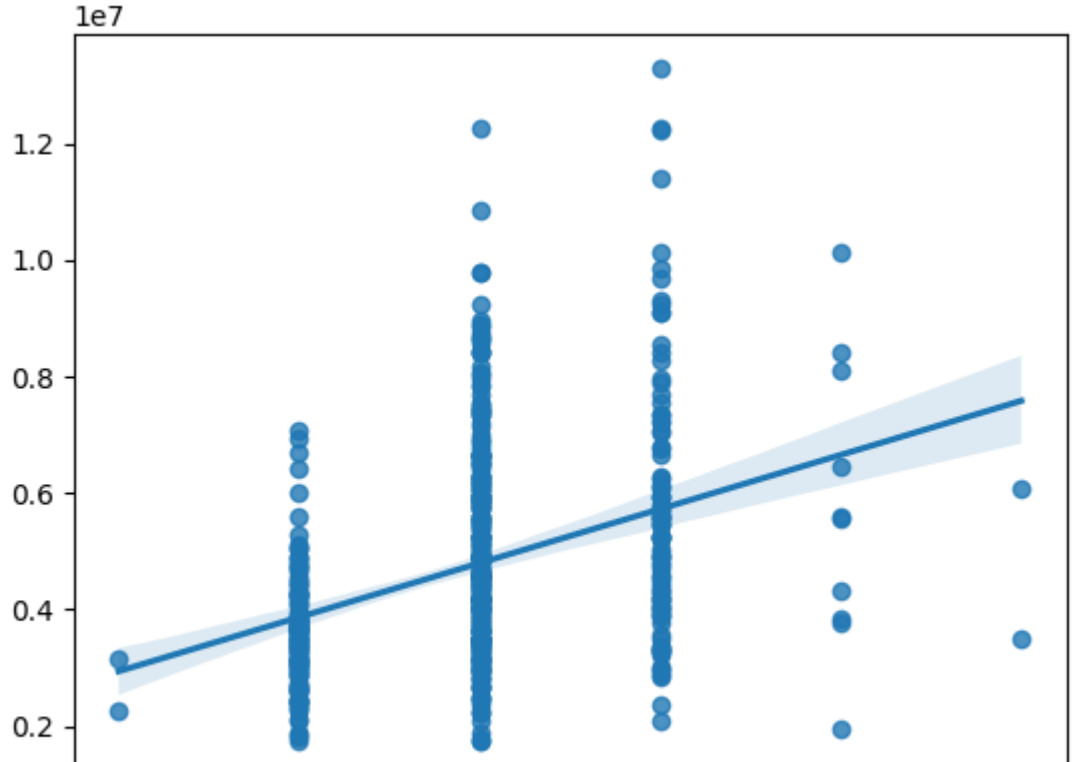
```
x=df.iloc[:,1:].values
y=df.iloc[:,0].values
```

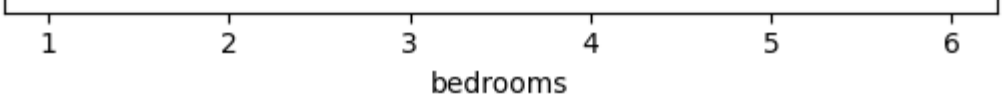
```
lt=['area',   'bedrooms',   'bathrooms',  'stories',  'mainroad',   'guestroom',  'basement
    'furnishingstatus']


for i in lt:
  print(sns.regplot(x=df[i],y=y))
  plt.show()
```

```
Axes(0.125,0.11;0.775x0.77)
```

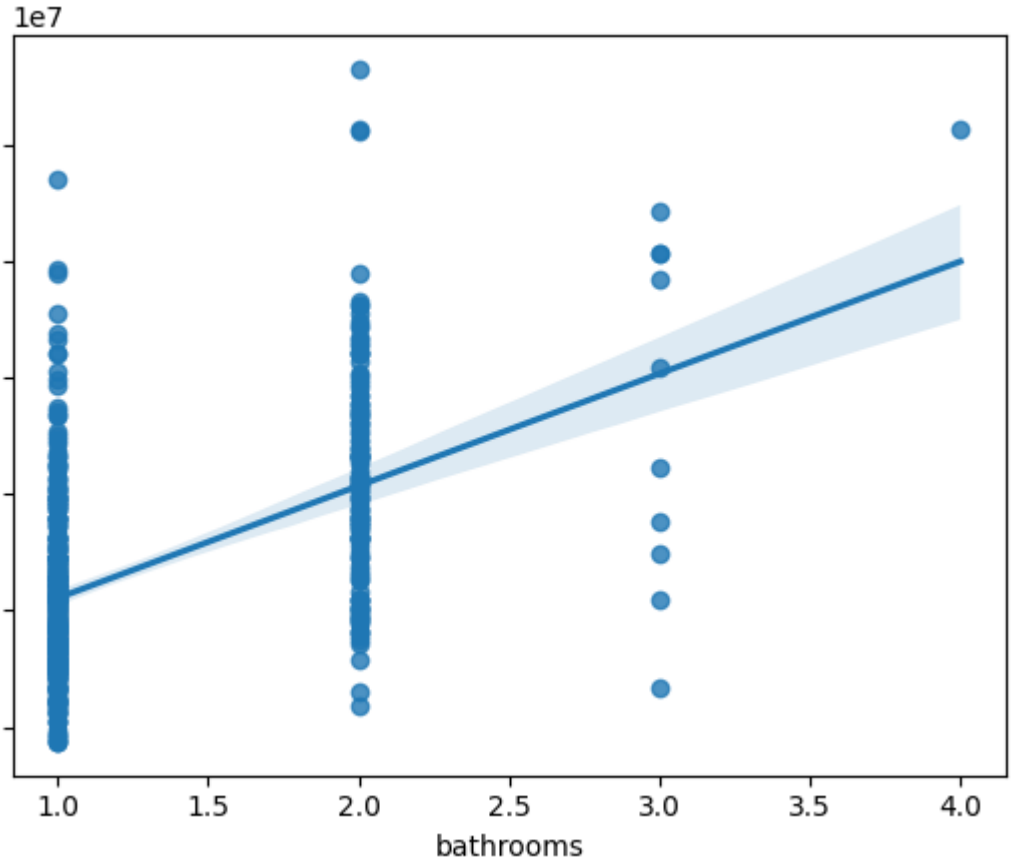



```
Axes(0.125,0.11;0.775x0.77)
```

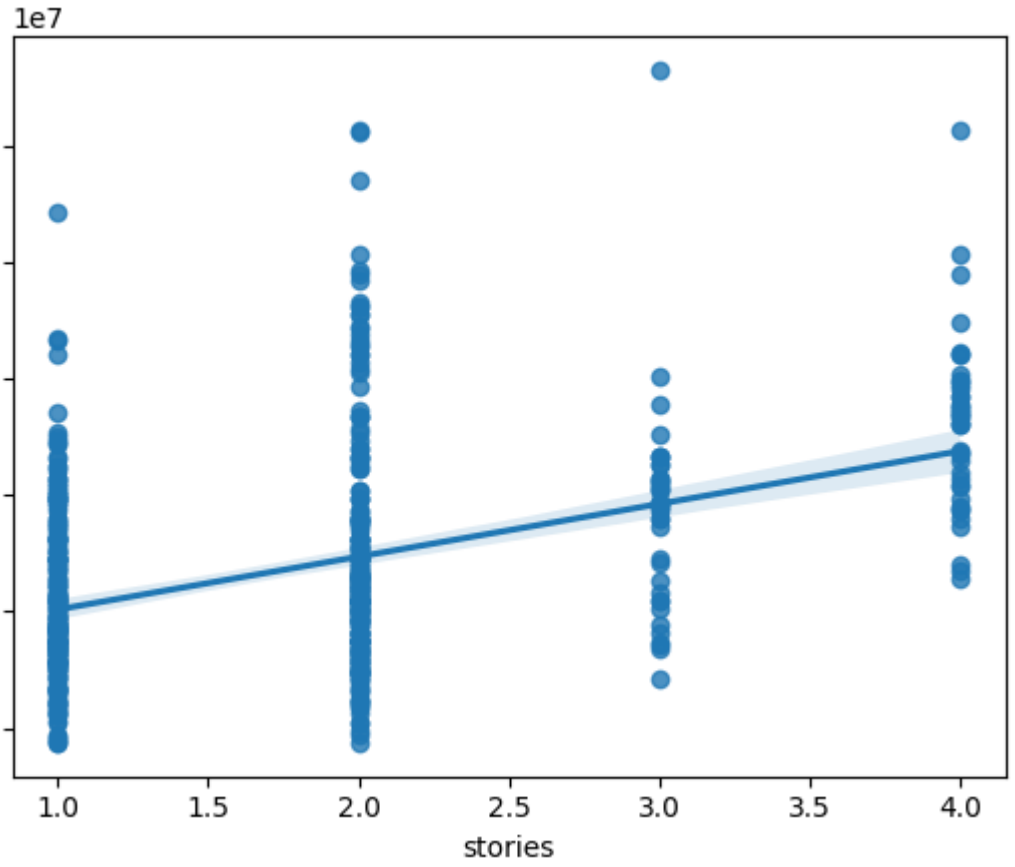




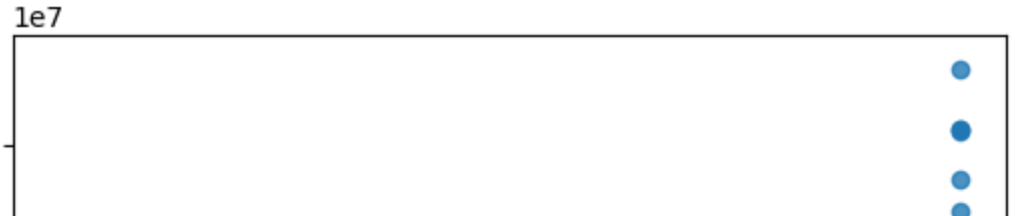

```
Axes(0.125,0.11;0.775x0.77)
```

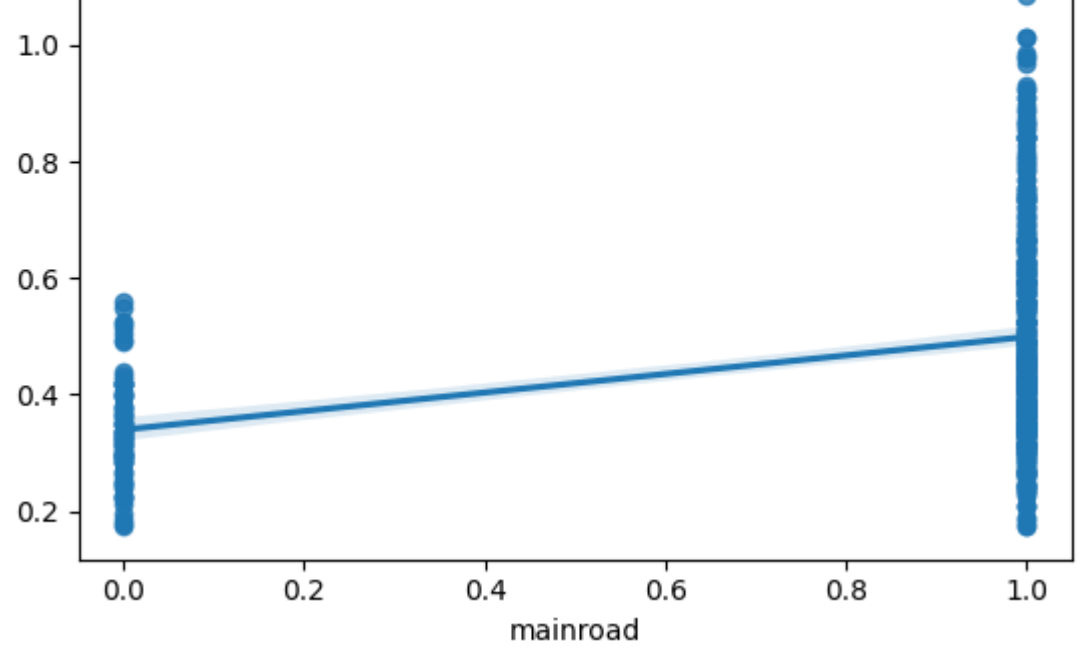



```
Axes(0.125,0.11;0.775x0.77)
```

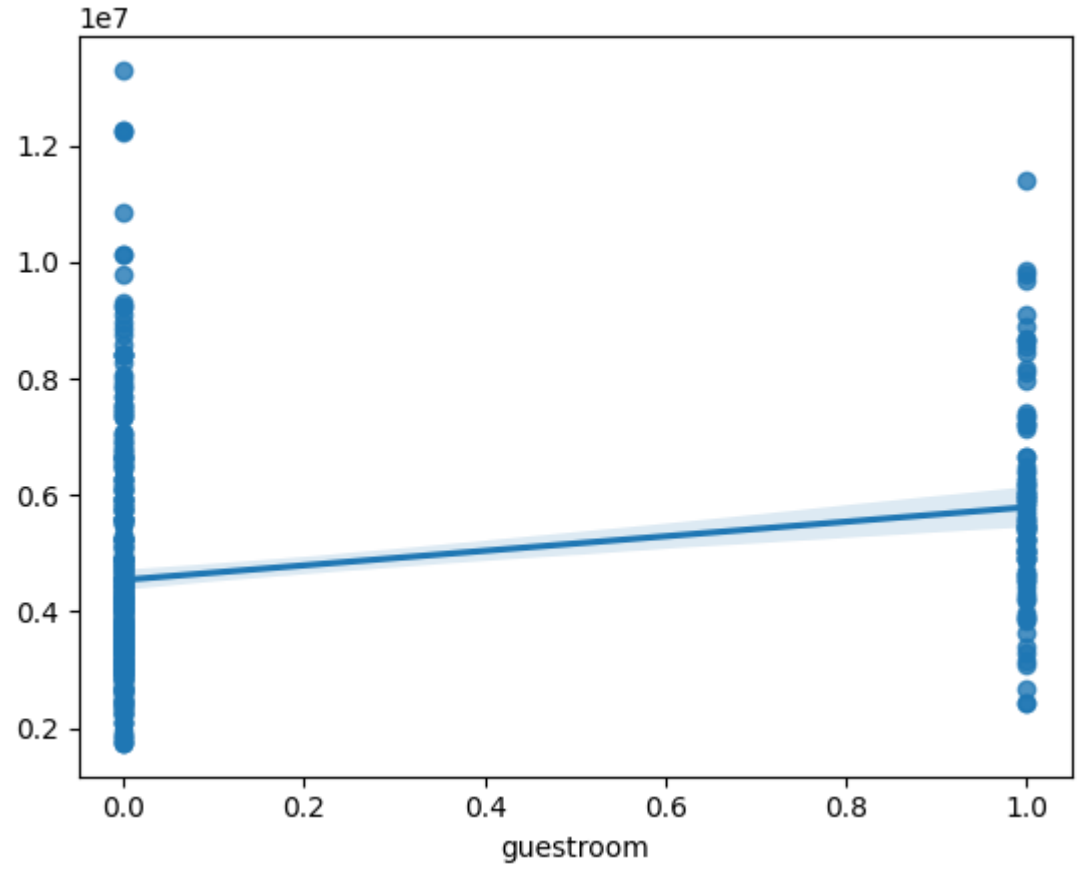



```
Axes(0.125,0.11;0.775x0.77)
```

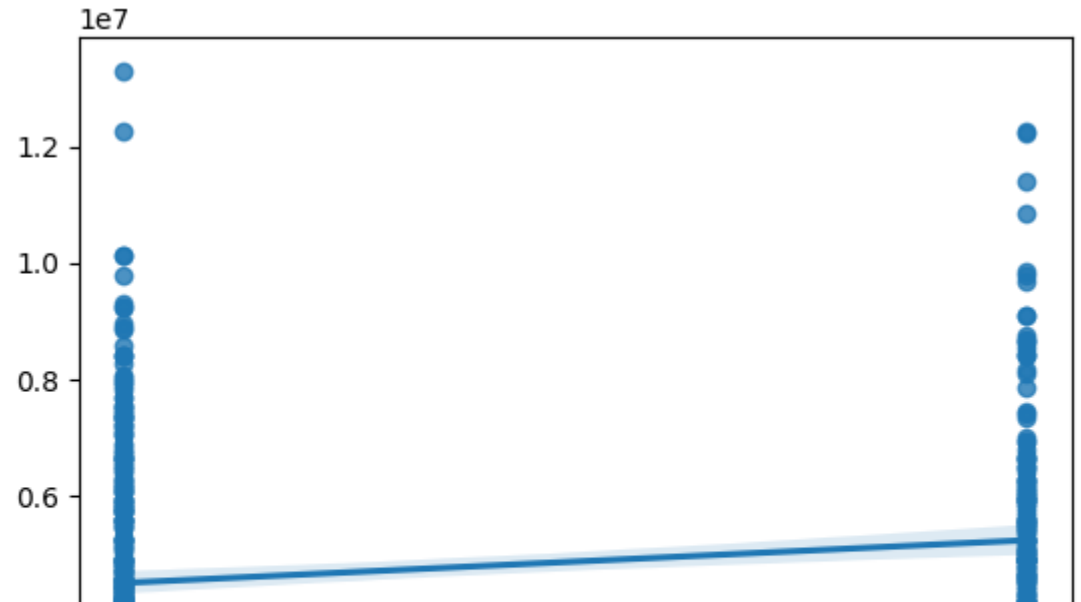


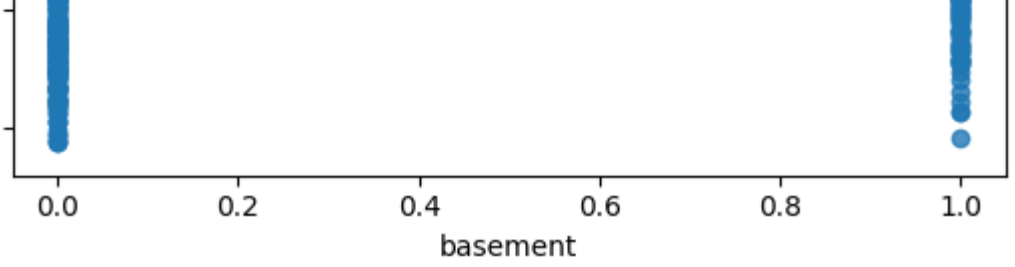



```
Axes(0.125,0.11;0.775x0.77)
```

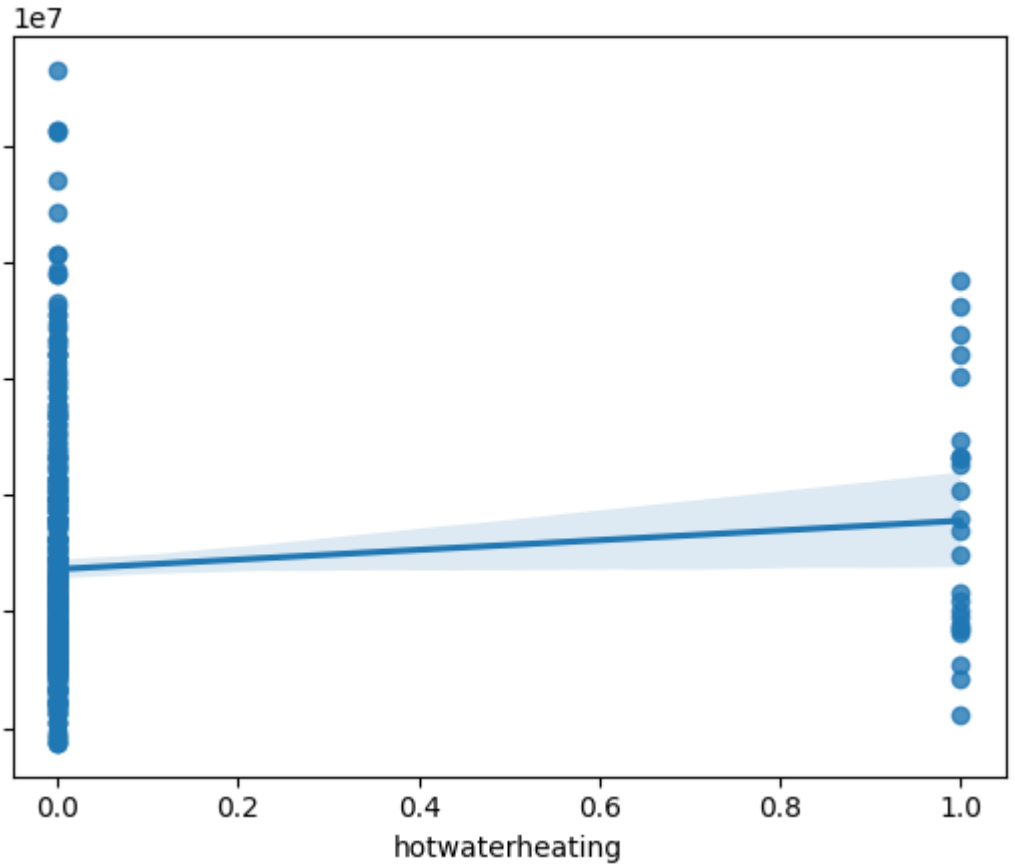



```
Axes(0.125,0.11;0.775x0.77)
```

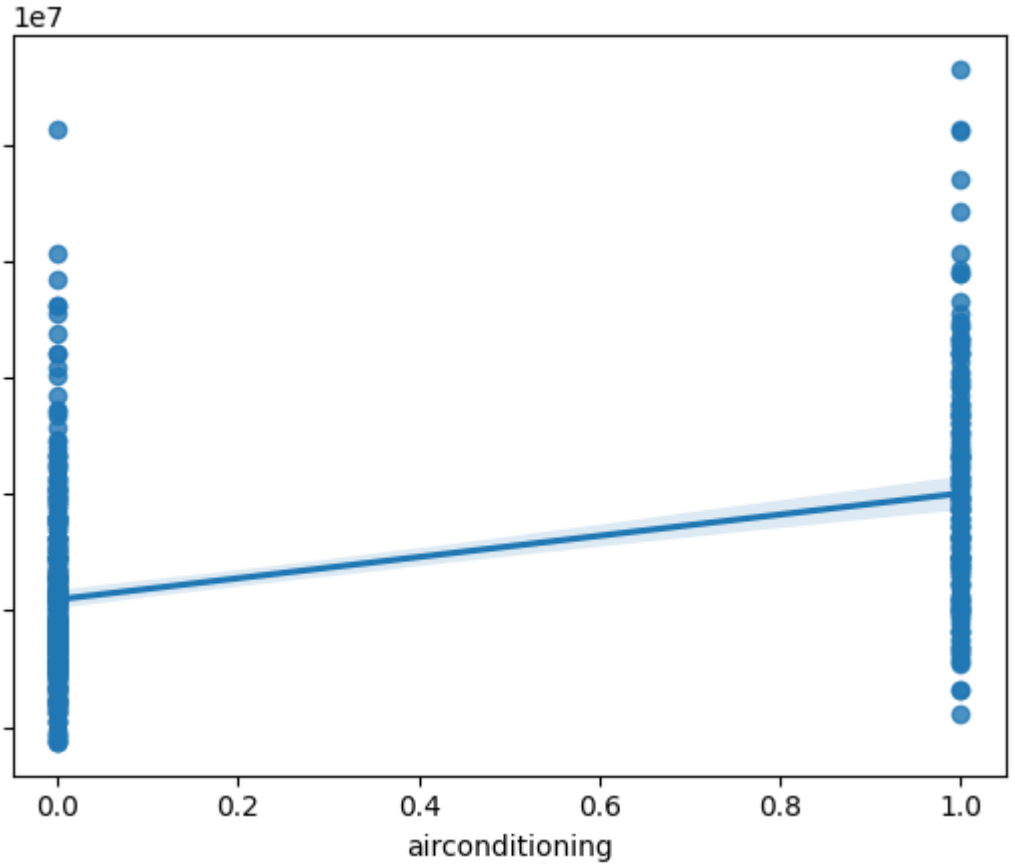




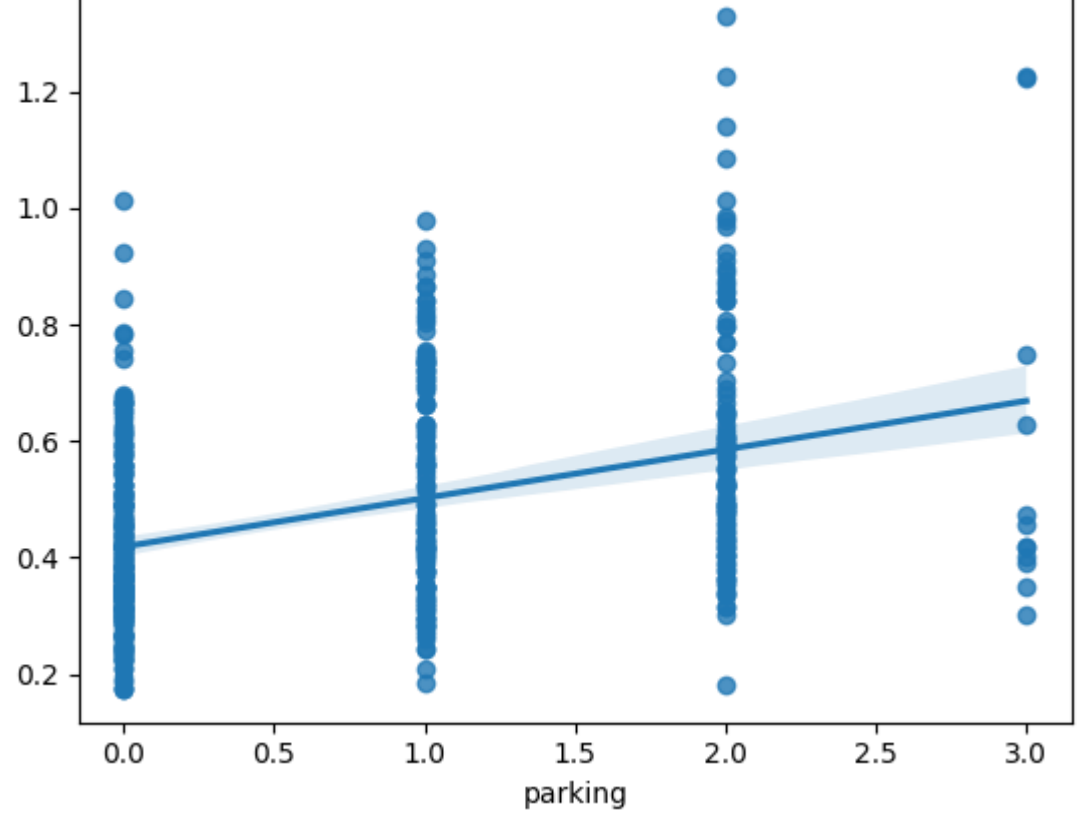

```
Axes(0.125,0.11;0.775x0.77)
```



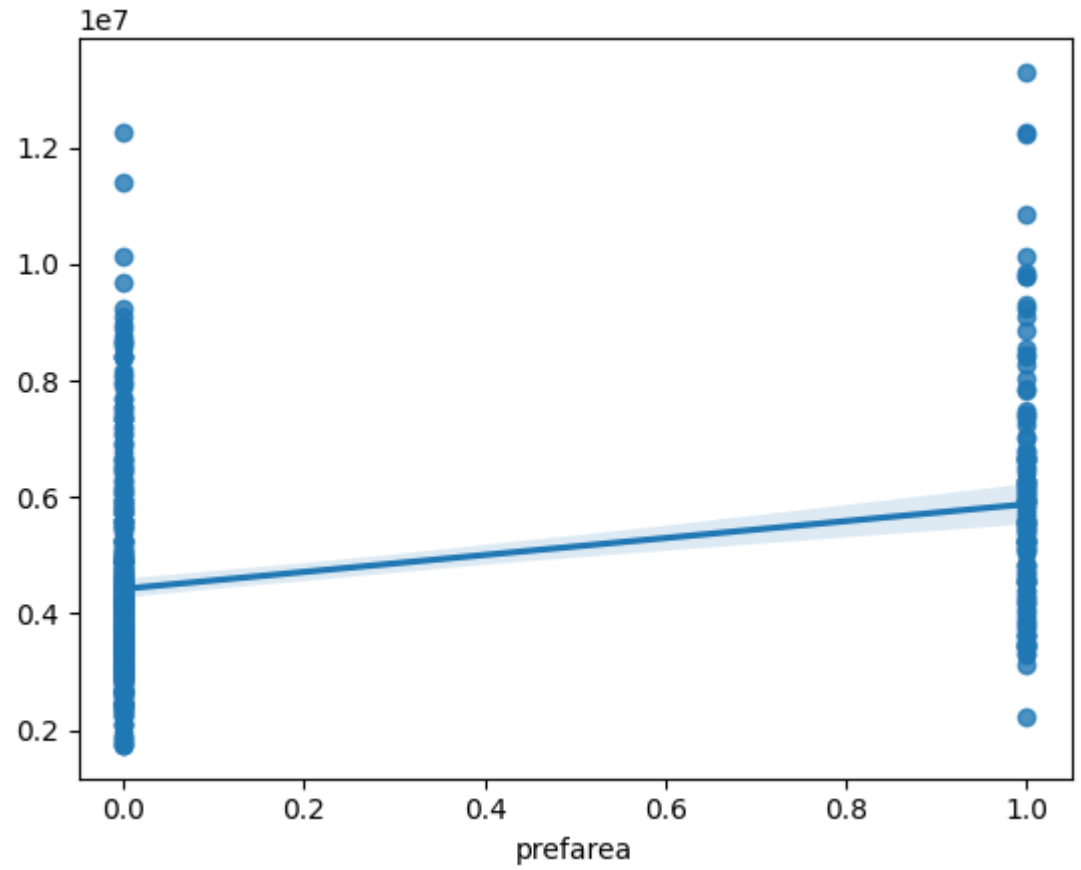

```
Axes(0.125,0.11;0.775x0.77)
```



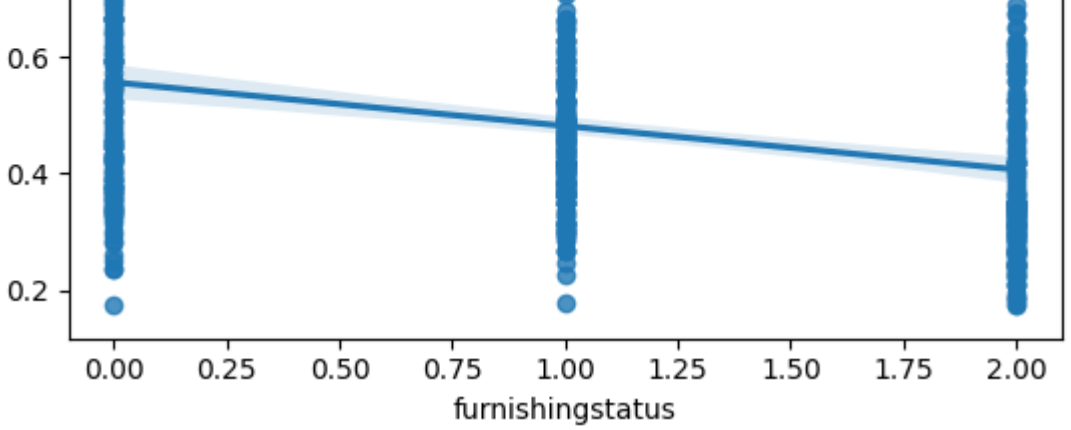

```
Axes(0.125,0.11;0.775x0.77)
```




```
Axes(0.125,0.11;0.775x0.77)
```




```
Axes(0.125,0.11;0.775x0.77)
```




```
from sklearn.model_selection import train_test_split
xtr,xts,ytr,yts=train_test_split(x,y,test_size=0.30,random_state=42)
```

```
from sklearn.linear_model import LinearRegression
linear=LinearRegression()
linear.fit(xtr,ytr)
ypr=linear.predict(xts)
```

```
df1=pd.DataFrame({'actual':yts,'predicted':ypr,'differance':yts-ypr})
df1
```

	actual	predicted	differance
0	4060000	5.407509e+06	-1.347509e+06
1	6650000	7.097185e+06	-4.471855e+05
2	3710000	3.055462e+06	6.545376e+05
3	6440000	4.476945e+06	1.963055e+06
4	2800000	3.315984e+06	-5.159837e+05
...	...	...	...
159	3500000	3.796903e+06	-2.969026e+05
160	3360000	4.130262e+06	-7.702624e+05
161	4970000	7.058249e+06	-2.088249e+06
162	3150000	5.417344e+06	-2.267344e+06
163	5530000	5.180834e+06	3.491663e+05

164 rows × 3 columns

```
print('slope is ')
print(list(zip(df.iloc[:,1:],linear.coef_)))
print('constant is ',linear.intercept_)
```

```
slope is
[('area', 253.06232837180667), ('bedrooms', 82734.87457030988), ('bathrooms', 1117372.
constant is  124157.75864096358
```

```
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared
```

```
# absolute error
print('mae',mean_absolute_error(yts,ypr))
# maep
print('maep',mean_absolute_percentage_error(yts,ypr))
#mse
print('mse',mean_squared_error(yts,ypr))
#rmse
print('rmse',np.sqrt(mean_squared_error(yts,ypr)))
```

```
mae 925543.5483156566
maep 0.21335770783051045
mse 1535047758428.0508
rmse 1238970.4429194632
```

```
#r2 score

from sklearn.metrics import r2_score
print('r2 ',r2_score(yts,ypr))
```

```
r2  0.6435419628959105
```