Linear Regression Model

In [16]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [17]:
```
df=pd.read_csv("House_Price.csv")
```

In [18]:
```
df.head()
```

Out[18]:

	price	crime_rate	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers
0	24.0	0.00632	32.31	0.538	6.575	65.2	4.35	3.81	4.18	4.01	24.7
1	21.6	0.02731	37.07	0.469	6.421	78.9	4.99	4.70	5.12	5.06	22.2
2	34.7	0.02729	37.07	0.469	7.185	61.1	5.03	4.86	5.01	4.97	22.2
3	33.4	0.03237	32.18	0.458	6.998	45.8	6.21	5.93	6.16	5.96	21.3
4	36.2	0.06905	32.18	0.458	7.147	54.2	6.16	5.86	6.37	5.86	21.3

In [19]:
```
df.shape
```

Out[19]:
```
(506, 20)
```

Training Model

In [20]:
```
from sklearn.linear_model import LinearRegression
```

In [21]:
```
from sklearn.model_selection import train_test_split
```

In [22]:
```
y=df[["price"]]
```

In [23]:
```
x=df[["room_num"]]
```

In [24]:
```
lm = LinearRegression()
```

Train Test Split to train the Model

In [25]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random
_state=101)
```

In [26]:
```
lm.fit(x_train,y_train)
```

Out[26]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [27]:
```
print(lm.intercept_,lm.coef_)
```

```
[-31.71878837] [[8.54740782]]
```

Predicting the Model Value After Training Phase

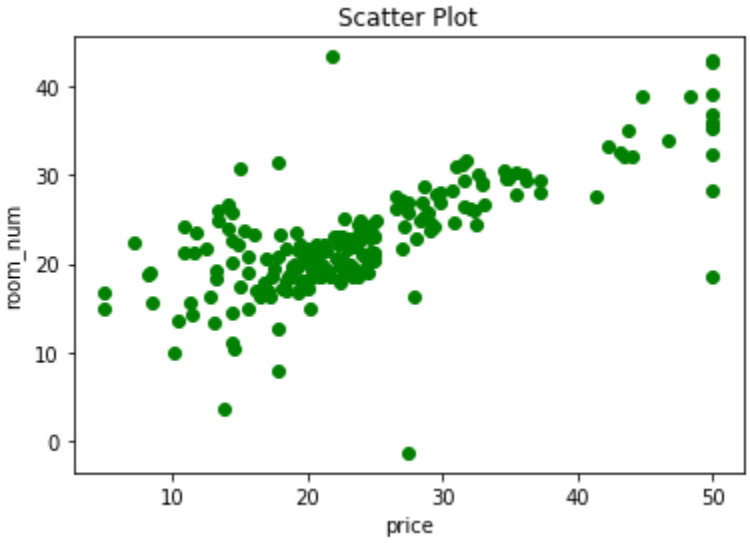
In [28]:
```
pred=lm.predict(x_test)
print(pred[0:10])
```

```
[[35.59204817]
 [29.36953528]
 [21.34351934]
 [22.13842827]
 [24.65136617]
 [30.13025458]
 [42.85734481]
 [19.13828813]
 [27.62586409]
 [15.55692425]]
```

Plotting Graph from the Dataset

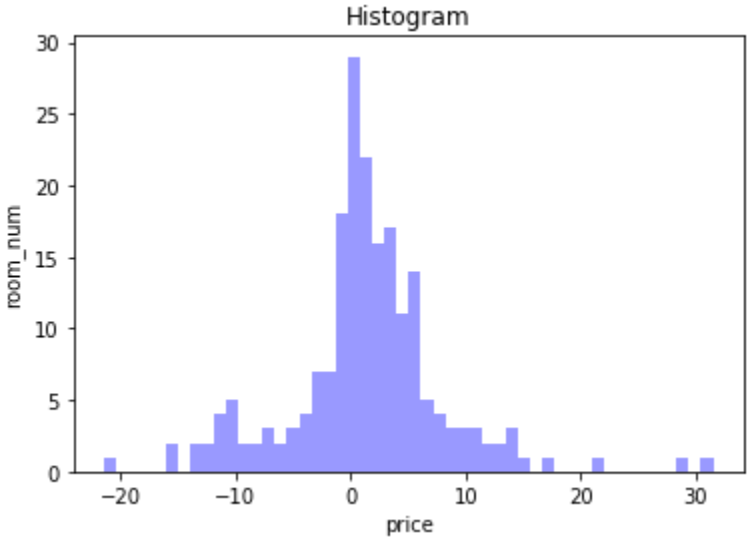
In [29]:
```
plt.scatter(y_test,pred,color="green")
plt.xlabel("price")
plt.ylabel("room_num")
plt.title("Scatter Plot")
```

Out[29]:
```
Text(0.5, 1.0, 'Scatter Plot')
```



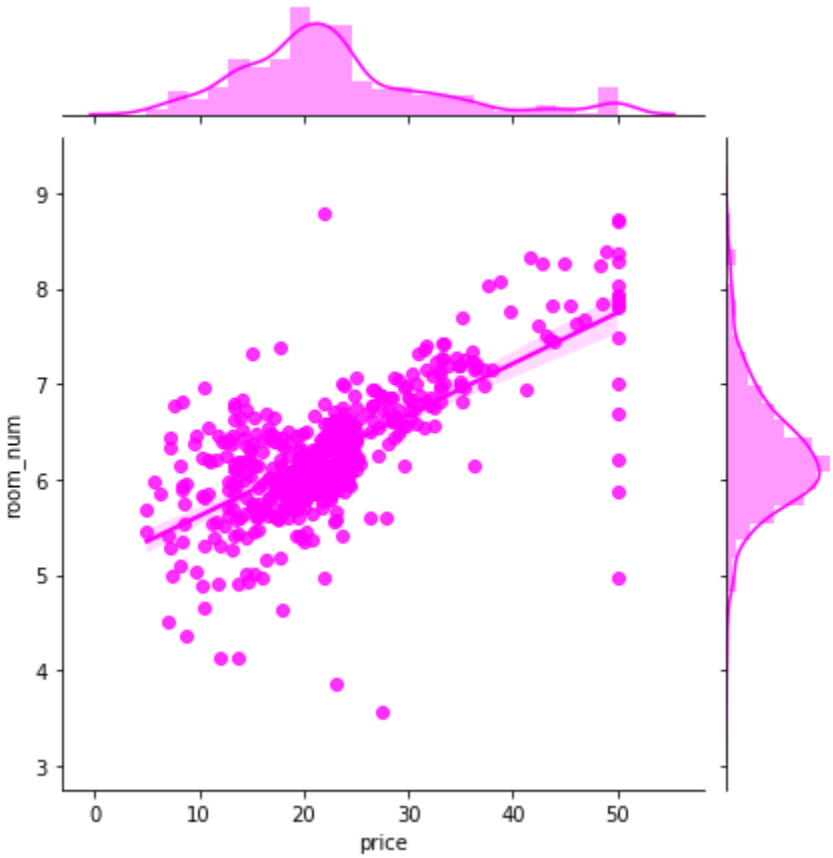
In [30]:
```
sns.distplot((y_test-pred),bins=50,color="blue",kde= False)
plt.xlabel("price")
plt.ylabel("room_num")
plt.title("Histogram")
```

Out[30]: Text(0.5, 1.0, 'Histogram')



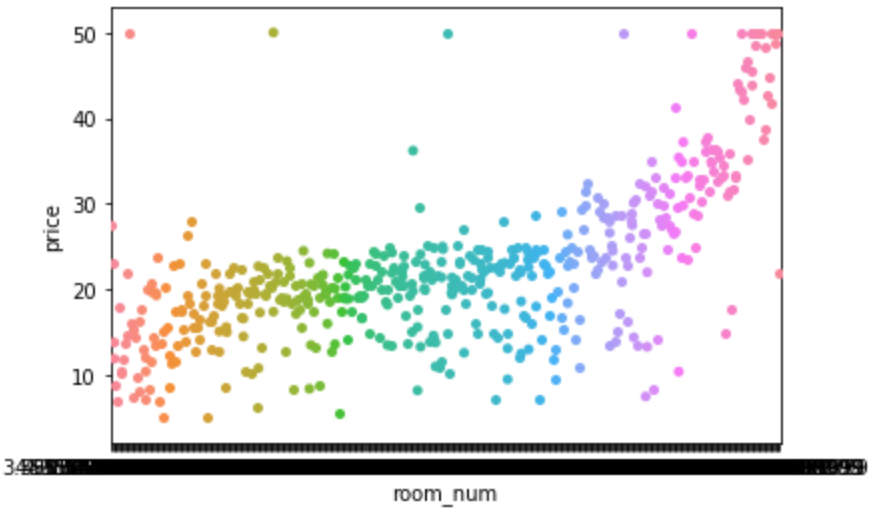
In [31]:
```
sns.jointplot(x=df[["price"]],y=df[["room_num"]],data=df,kind="reg",color="magenta")
plt.xlabel("price")
plt.ylabel("room_num")
```

Out[31]: Text(27.125, 0.5, 'room_num')



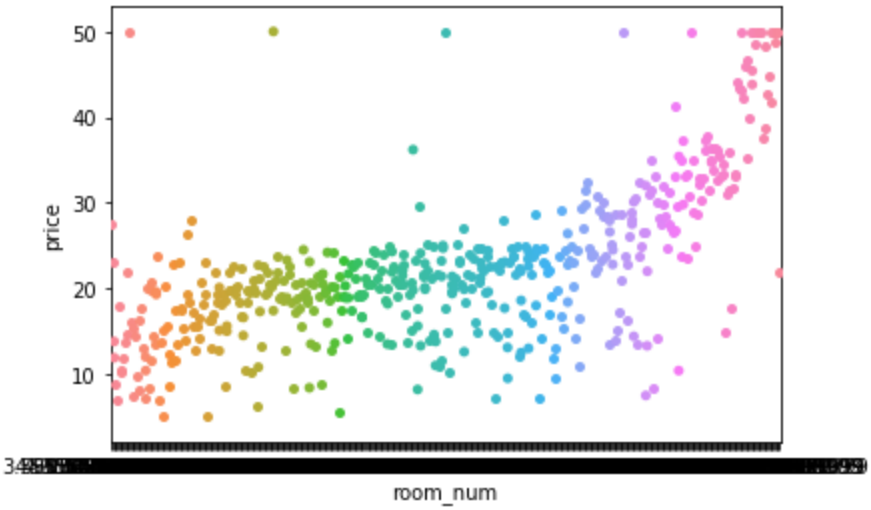
In [32]:
```
sns.swarmplot(x="room_num",y="price",data=df)
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x196d4f31ac8>

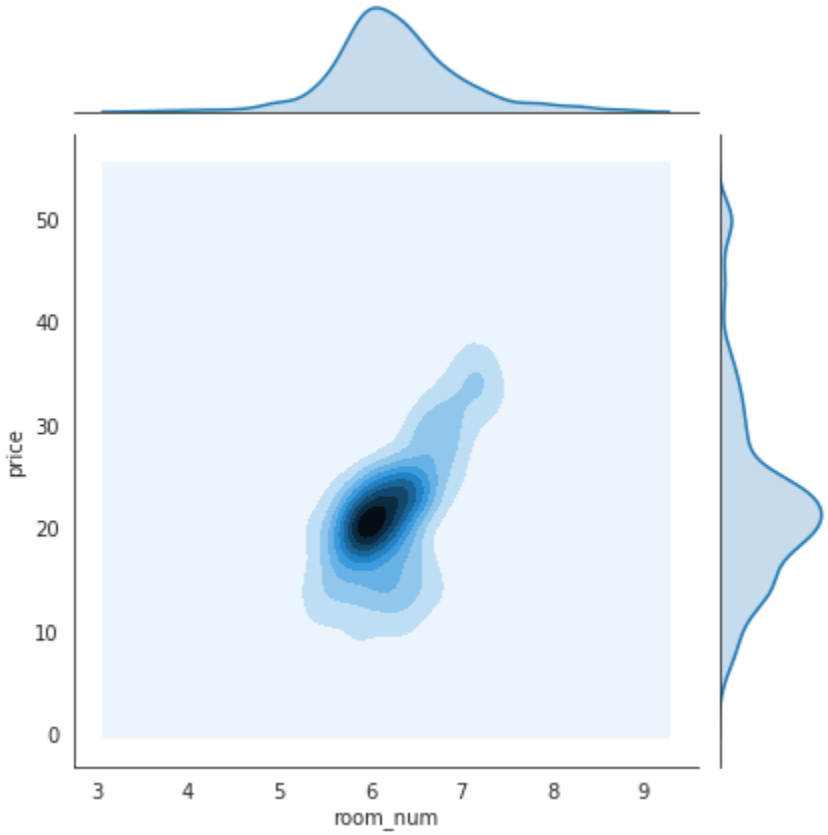


In [33]:
```
sns.stripplot(x="room_num",y="price",data=df)
```

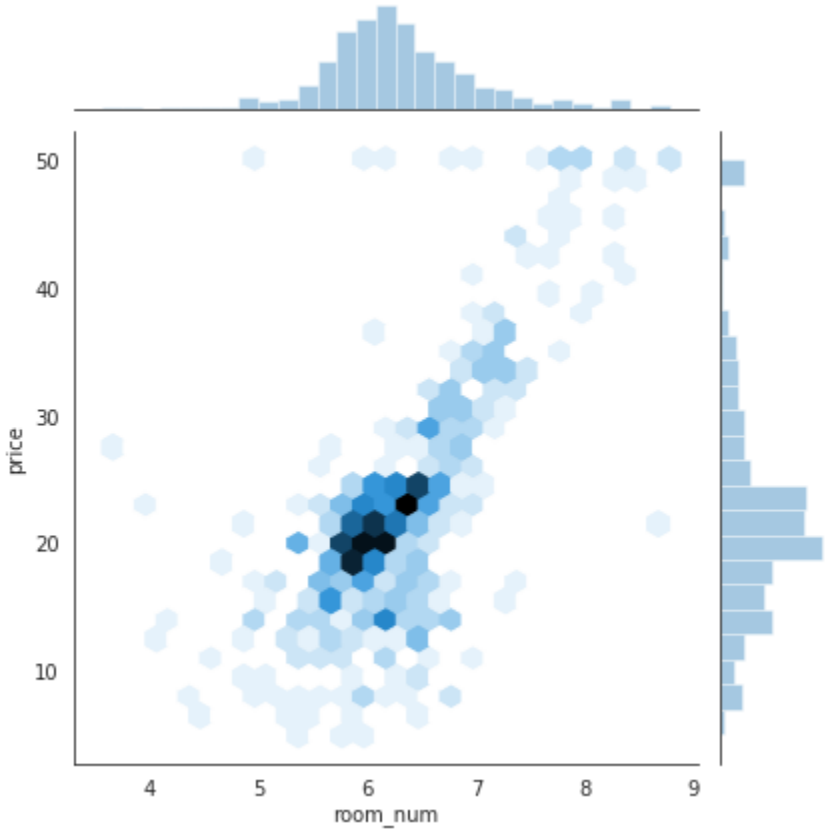
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x196d4e53488>



In [34]:
```
with sns.axes_style('white'):
    sns.jointplot(x="room_num",y="price",data=df,kind='kde')
```

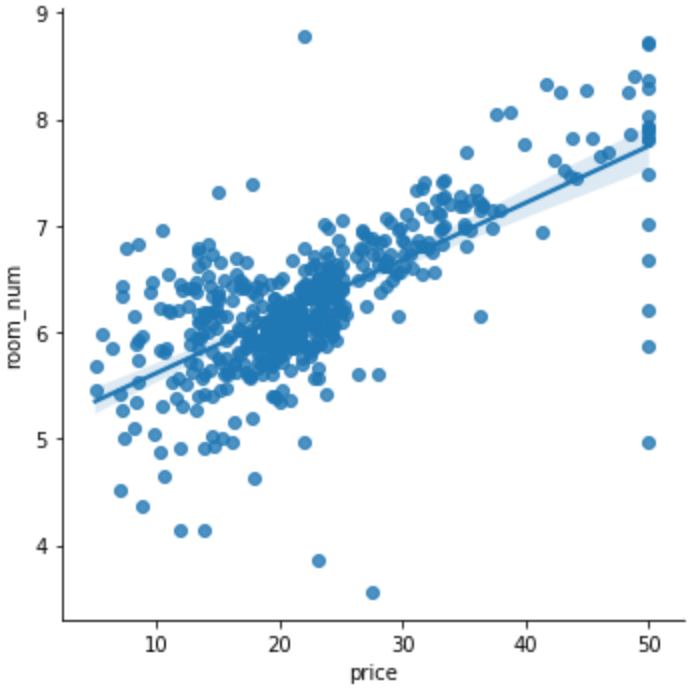


In [35]:
```
with sns.axes_style('white'):
    sns.jointplot(x="room_num",y="price",data=df,kind='hex')
```



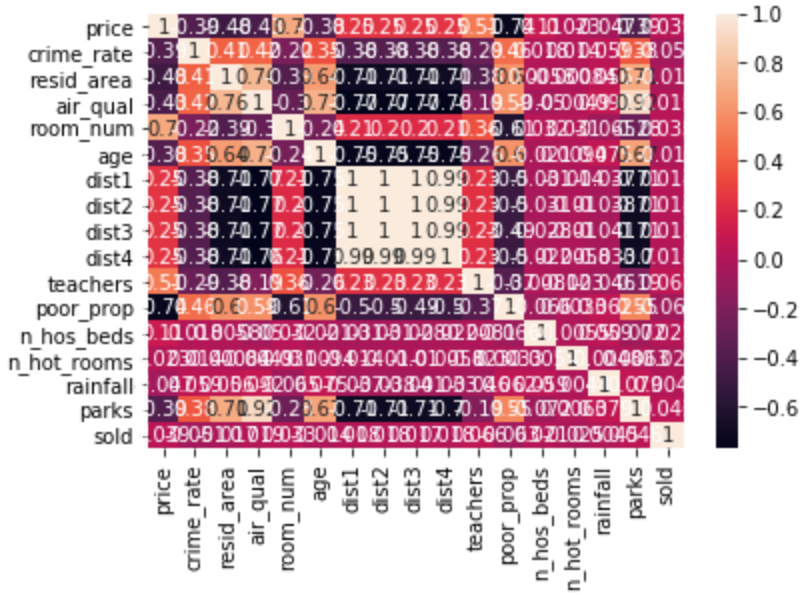
In [36]:
```
sns.lmplot(x="price",y="room_num",data=df)
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x196d7654dc8>



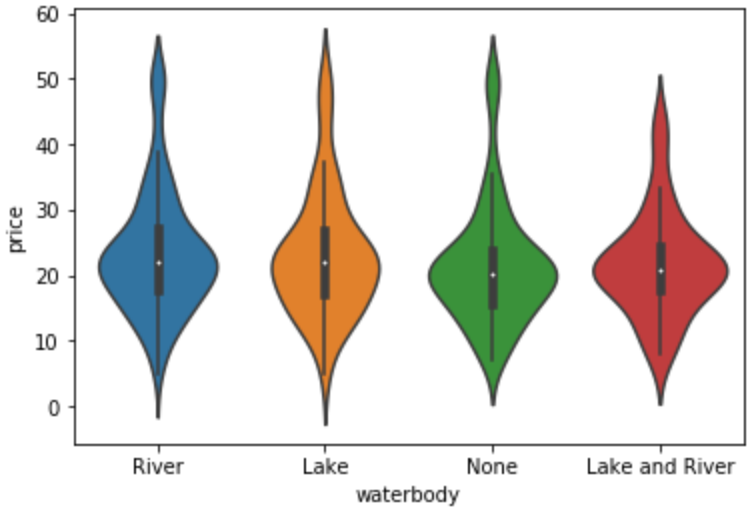
In [37]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x196d7649a08>



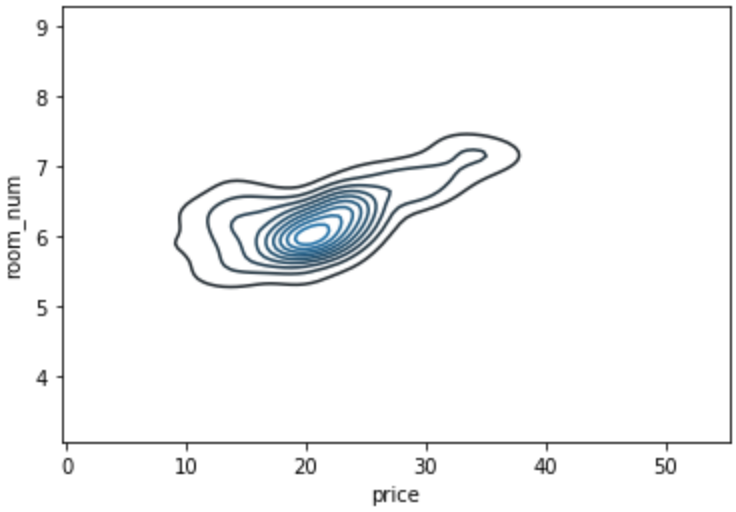
In [38]:
```
sns.violinplot(x="waterbody",y="price",data=df)
```

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x196d7a11548>

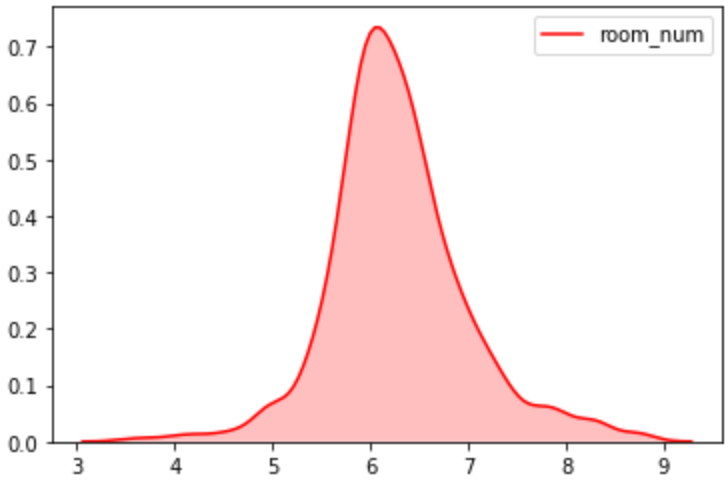


In [39]:
```
sns.kdeplot(df.price,df.room_num)
```

Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x196d7a23a48>



In [40]:
```
p1=sns.kdeplot(df["room_num"],shade=True,color="r")
```



Calculating Error after the Training Phase

In [41]:
```
from sklearn import metrics
```

In [42]:
```
metrics.mean_absolute_error(y_test,pred)
```

Out[42]:
```
4.774097035805513
```

In [43]:
```
metrics.mean_squared_error(y_test,pred)
```

Out[43]:
```
48.359147224858454
```

In [44]:
```
print(np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
6.954074145769403
```