In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

In [2]:
```
os.getcwd()
```

Out[2]:
```
'C:\\Users\\BrighterDays CodeLab'
```

In [3]:
```
data = pd.read_csv("housing.data")
```

In [4]:
```
data
```

Out[4]:

	0.00632 18.00 2.310 0 0.5380 6.5750 65.20 4.0900 1 296.0 15.30 396.90 4.98 24.00
0	0.02731 0.00 7.070 0 0.4690 6.4210 78...
1	0.02729 0.00 7.070 0 0.4690 7.1850 61...
2	0.03237 0.00 2.180 0 0.4580 6.9980 45...
3	0.06905 0.00 2.180 0 0.4580 7.1470 54...
4	0.02985 0.00 2.180 0 0.4580 6.4300 58...
...	...
500	0.06263 0.00 11.930 0 0.5730 6.5930 69...
501	0.04527 0.00 11.930 0 0.5730 6.1200 76...
502	0.06076 0.00 11.930 0 0.5730 6.9760 91...
503	0.10959 0.00 11.930 0 0.5730 6.7940 89...
504	0.04741 0.00 11.930 0 0.5730 6.0300 80...

505 rows × 1 columns

In [5]:
```
data = pd.read_csv("housing.data", delim_whitespace = True, header = None)
data
```

Out[5]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

506 rows × 14 columns

In [8]:
```
from sklearn.datasets import load_boston
```

In [9]:
```
boston =load_boston()
```

In [ ]:
```

```

In [10]:
```
dir(boston)
```

Out[10]:
```
['DESCR', 'data', 'feature_names', 'filename', 'target']
```

In [12]:
```
boston.feature_names
```

Out[12]:
```
array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')
```

In [22]:
```
newHeader = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']
```

In [23]:
```
data.columns
```

Out[23]:
```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'PRICE'],
      dtype='object')
```

In [24]:
```
data.columns = newHeader()
```

In [25]:
```
data
```

Out[25]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LST
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.

506 rows × 14 columns

In [26]:
```
dff1 = pd.DataFrame(data = boston.data, columns = boston.feature_names)
```

In [27]:
```
dff1
```

Out[27]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LST
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99	9.
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90	9.
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90	5.
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45	6.
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90	7.

506 rows × 13 columns

In [30]:
```
dff1["Price"] = boston.target
```

In [31]:
```
boston.data.shape
```

Out[31]:
```
(506, 13)
```

In [47]:
```
dff1 = data
```

In [55]:
```
data["Price"] = data['MEDV']
```

In [56]:
```
data
```

Out[56]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LST
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.

506 rows × 15 columns

In [57]:
```
type(boston)
```

Out[57]:
```
sklearn.utils.Bunch
```

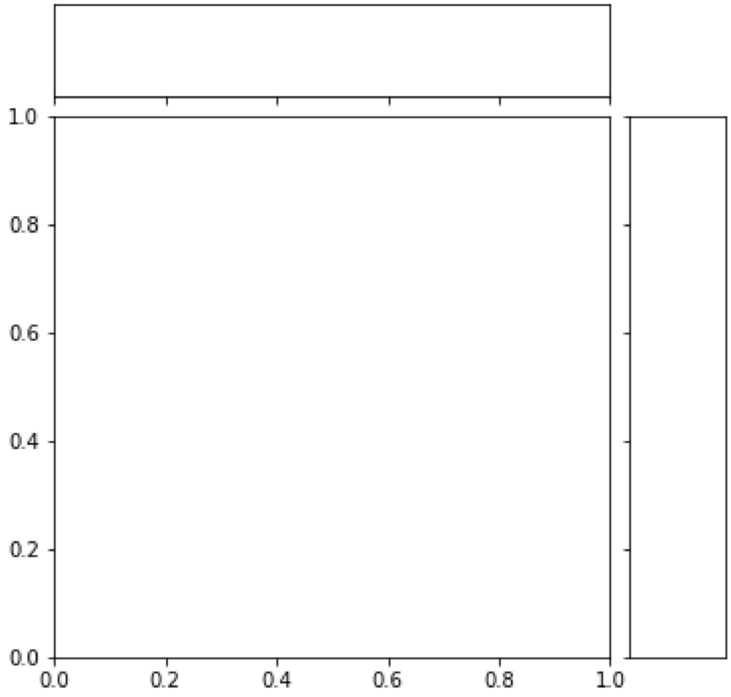
In [58]:
```
import seaborn as sns
```

In [59]:

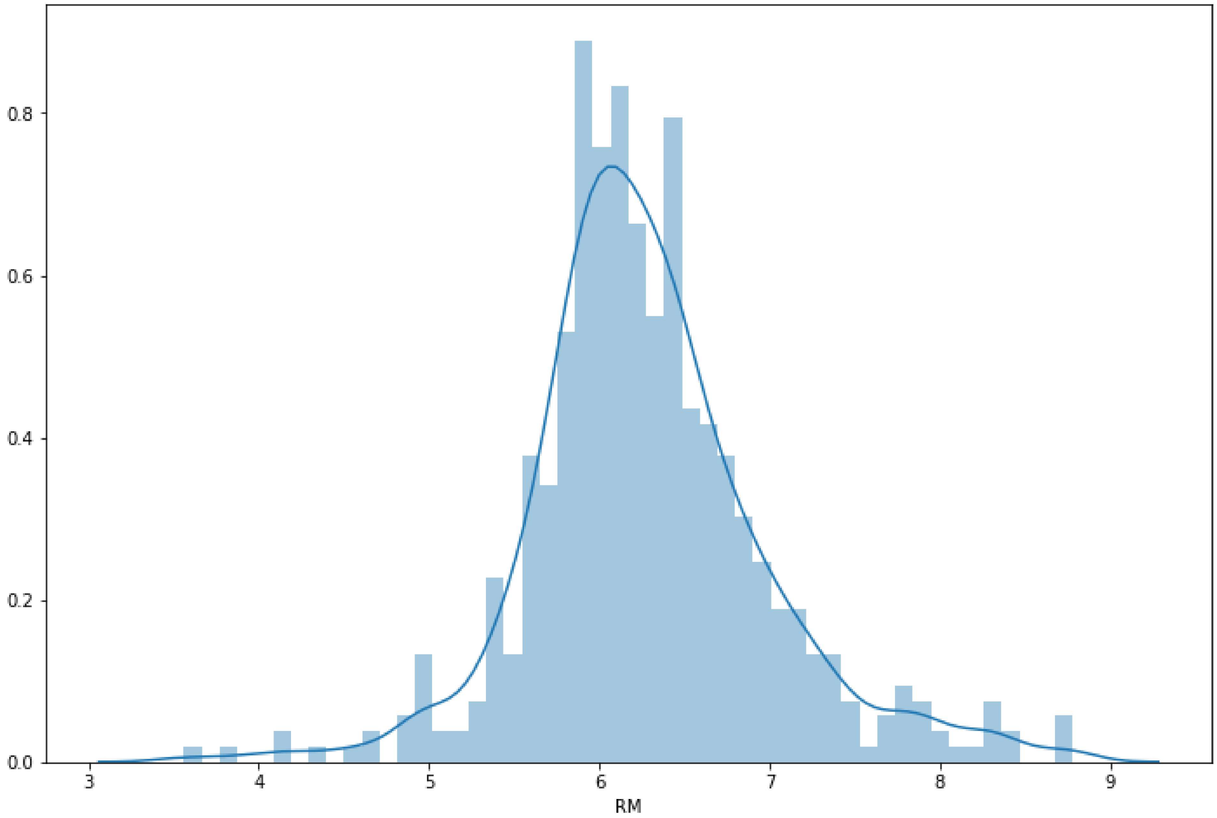
```
plt.figure(figsize = (12,8))
sns.jointplot(dff1 = data, x='RM', y='Price')
plt.grid()
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-59-74db496a38b7> in <module>
      1 plt.figure(figsize = (12,8))
----> 2 sns.jointplot(dff1 = data, x='RM', y='Price')
      3 plt.grid()
      4 plt.show()

~\Anaconda3\lib\site-packages\seaborn\axisgrid.py in jointplot(x, y, data, kind, stat_func, color, height, ratio, space, dropna, xlim, ylim, joint_kws, marginal_kws, annot_kws, **kwargs)
   2289     grid = JointGrid(x, y, data, dropna=dropna,
   2290                      height=height, ratio=ratio, space=space,
-> 2291                      xlim=xlim, ylim=ylim)
   2292
   2293     # Plot the data using the grid

~\Anaconda3\lib\site-packages\seaborn\axisgrid.py in __init__(self, x, y, data, height, ratio, space, dropna, xlim, ylim, size)
   1708             if isinstance(var, str):
   1709                 err = "Could not interpret input '{}'".format(var)
-> 1710                 raise ValueError(err)
   1711
   1712         # Find the names of the variables

ValueError: Could not interpret input 'RM'
```

```
<Figure size 864x576 with 0 Axes>
```



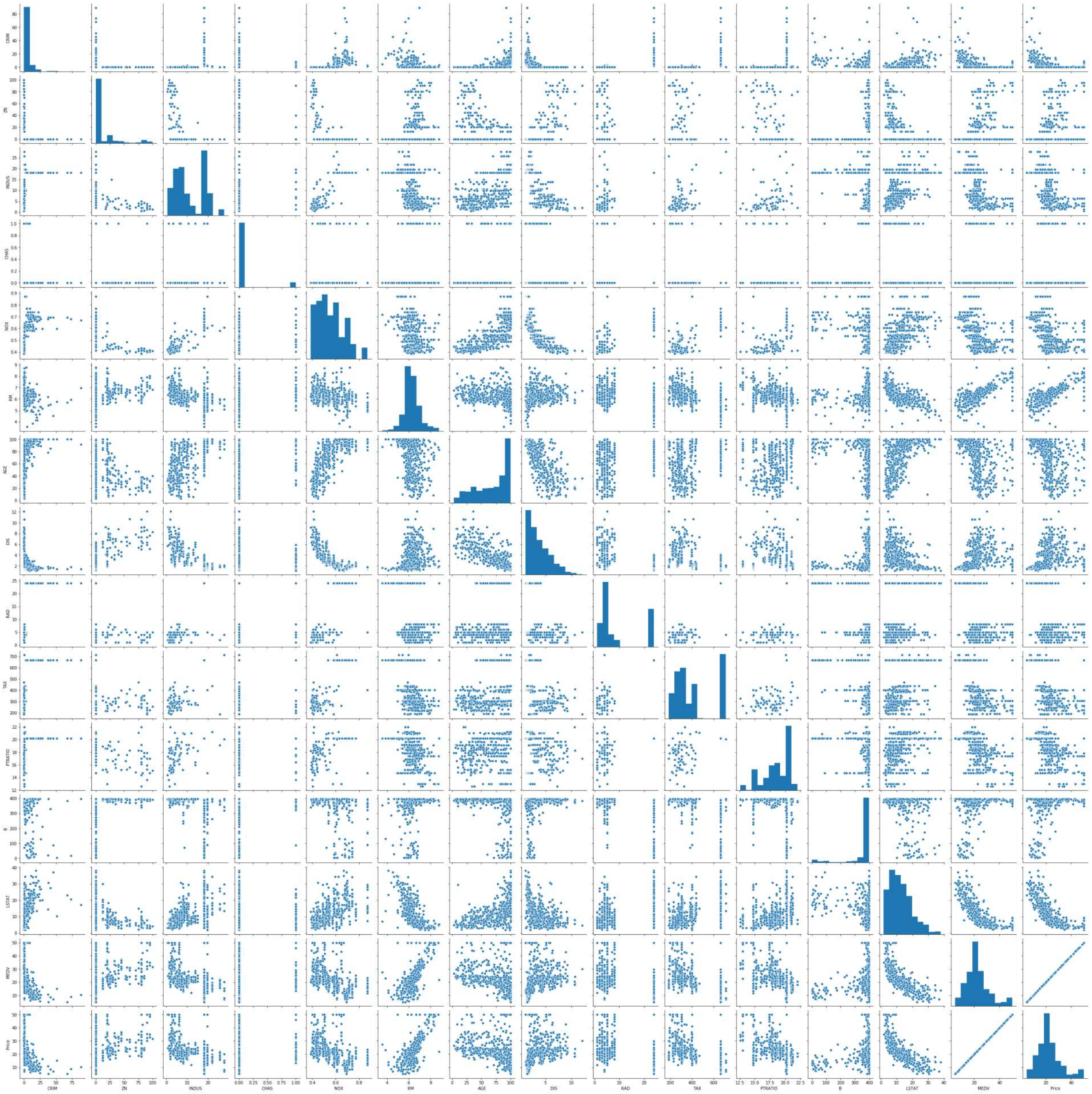
```
In [60]: plt.figure(figsize = (12,8))
         sns.distplot(data['RM'], bins = 50,kde =True,)
         plt.show()
```



```
In [ ]: data['R.M']
```

In [61]:
```
sns.pairplot(data)
```

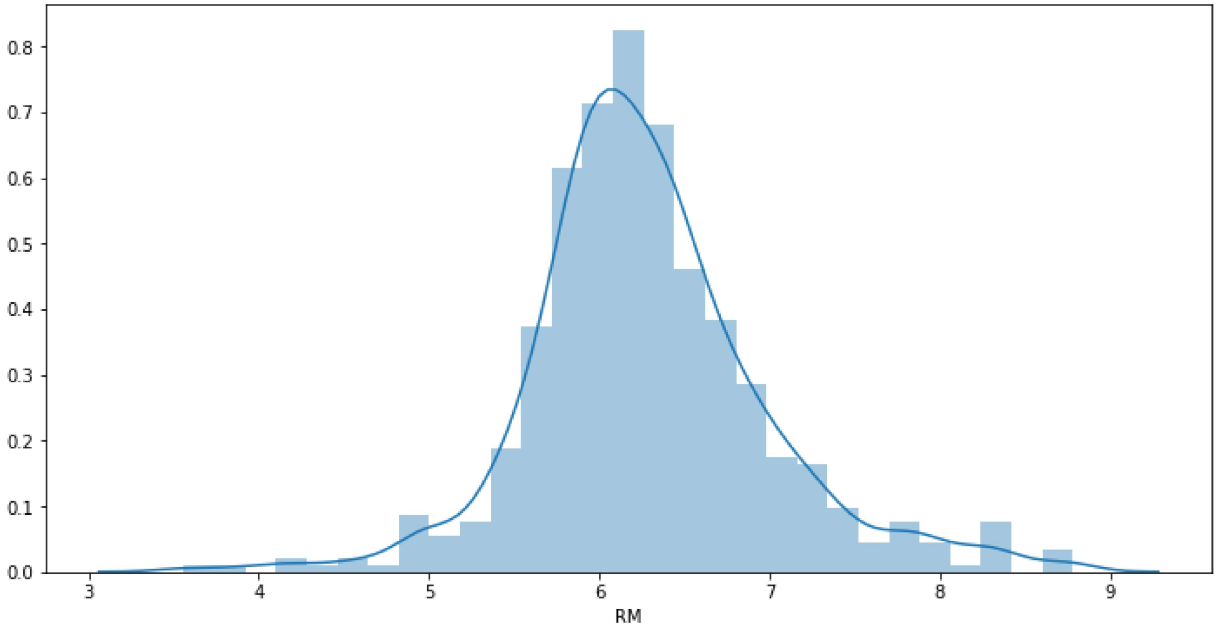
Out[61]:
```
<seaborn.axisgrid.PairGrid at 0xa2fa1d1508>
```



In [62]:
```
data['RM'].corr(data['Price'])
```

Out[62]:
```
0.6953599470715394
```

In [72]:
```
plt.figure(figsize = (12,6))
sns.distplot(data['RM'])
plt.show()
```



In [74]:
```
data['Price'].corr(data['CRIM'])
```

Out[74]:
```
-0.38830460858681165
```

In [75]:
```
data['Price'].corr(data['CHAS'])
```

Out[75]:
```
0.1752601771902985
```

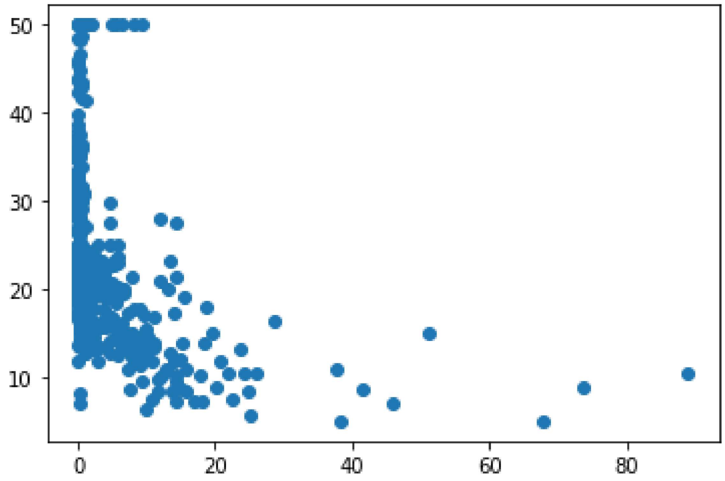
In [76]:
```
data
```

Out[76]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LST
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.

506 rows × 15 columns

In [80]:
```
plt.scatter(data['CRIM'],data['Price'])
plt.figure(figsize = (12,8))
plt.show()
```



```
<Figure size 864x576 with 0 Axes>
```

In [119]:
```
dff1.columns
```

Out[119]:
```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
       'PTRATIO', 'B', 'LSTAT', 'MEDV', 'Price'],
      dtype='object')
```

In [120]:
```
X = dff1[['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX',
```

In [121]:
```
X
```

Out[121]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LST
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.

506 rows × 13 columns

In [122]:
```
y = dff1['Price']
```

In [123]:
```
y
```

Out[123]:
```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ... 
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: Price, Length: 506, dtype: float64
```

In [124]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [126]:
```
model = LinearRegression()
```

In [127]:
```
model.fit(X, y)
```

Out[127]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [129]:
```
model.coef_
```

Out[129]:
```
array([-1.08011358e-01,  4.64204584e-02,  2.05586264e-02,  2.68673382e+00,
       -1.77666112e+01,  3.80986521e+00,  6.92224640e-04, -1.47556685e+00,
        3.06049479e-01, -1.23345939e-02, -9.52747232e-01,  9.31168327e-03,
       -5.24758378e-01])
```

In [131]:
```
model.intercept_
```

Out[131]:
```
36.45948838508991
```

In [134]:
```
pd.DataFrame(data = model.coef_, index = X.columns, columns =['COEF'])
```

Out[134]:

	COEF
CRIM	-0.108011
ZN	0.046420
INDUS	0.020559
CHAS	2.686734
NOX	-17.766611
RM	3.809865
AGE	0.000692
DIS	-1.475567
RAD	0.306049
TAX	-0.012335
PTRATIO	-0.952747
B	0.009312
LSTAT	-0.524758

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: