In [1]:

```
import pandas as pd
data = pd.read_csv("C:\\Users\\kmit\\Desktop\\housing.csv",",")
print(data.head(5))
```

```
   longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0    -122.23     37.88                41.0        880.0           129.0
1    -122.22     37.86                21.0       7099.0          1106.0
2    -122.24     37.85                52.0       1467.0           190.0
3    -122.25     37.85                52.0       1274.0           235.0
4    -122.25     37.85                52.0       1627.0           280.0

   population  households  median_income  median_house_value ocean_proximity

0       322.0       126.0         8.3252            452600.0        NEAR BAY

1      2401.0      1138.0         8.3014            358500.0        NEAR BAY

2       496.0       177.0         7.2574            352100.0        NEAR BAY

3       558.0       219.0         5.6431            341300.0        NEAR BAY

4       565.0       259.0         3.8462            342200.0        NEAR BAY
```

In [2]:

```
print("total samples....\n",data.size)
print("Null Values....\n", data.isnull().sum())
#print(data.isnull().count())
```

```
total samples....
 206400
Null Values....
 longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

In [3]:

```
d1 = data.dropna(subset=['total_bedrooms'])
print(data.shape,d1.shape)
```

```
(20640, 10) (20433, 10)
```

In [4]:

```
d1.cov()
```

Out[4]:

	longitude	latitude	housing_median_age	total_rooms	total_b
longitude	4.014324	-3.957670	-2.758919	1.991284e+02	5.876
latitude	-3.957670	4.563981	0.320091	-1.711788e+02	-6.029
housing_median_age	-2.758919	0.320091	158.553558	-9.923225e+03	-1.700
total_rooms	199.128445	-171.178818	-9923.224538	4.775403e+06	8.567
total_bedrooms	58.768508	-60.299623	-1700.312817	8.567306e+05	1.775
population	227.660858	-263.874646	-4220.630517	2.122942e+06	4.191
households	43.286878	-58.619704	-1457.475788	7.677502e+05	1.578
median_income	-0.059174	-0.323087	-2.828672	8.213000e+02	-6.180
median_house_value	-10499.897668	-35669.333210	154703.602850	3.362452e+07	2.416

In [5]:

```
d1.corr()
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	p
longitude	1.000000	-0.924616	-0.109357	0.045480	0.069608	
latitude	-0.924616	1.000000	0.011899	-0.036667	-0.066983	
housing_median_age	-0.109357	0.011899	1.000000	-0.360628	-0.320451	
total_rooms	0.045480	-0.036667	-0.360628	1.000000	0.930380	
total_bedrooms	0.069608	-0.066983	-0.320451	0.930380	1.000000	
population	0.100270	-0.108997	-0.295787	0.857281	0.877747	
households	0.056513	-0.071774	-0.302768	0.918992	0.979728	
median_income	-0.015550	-0.079626	-0.118278	0.197882	-0.007723	
median_house_value	-0.045398	-0.144638	0.106432	0.133294	0.049686	

In [6]:

```
d1.corr()['median_house_value']
```

Out[6]:

```
longitude            -0.045398
latitude             -0.144638
housing_median_age    0.106432
total_rooms           0.133294
total_bedrooms        0.049686
population           -0.025300
households            0.064894
median_income         0.688355
median_house_value    1.000000
Name: median_house_value, dtype: float64
```

In [7]:

```
d1.corr()['median_house_value'].sort_values()[::-1]
```

Out[7]:

```
median_house_value    1.000000
median_income         0.688355
total_rooms           0.133294
housing_median_age    0.106432
households            0.064894
total_bedrooms        0.049686
population           -0.025300
longitude            -0.045398
latitude             -0.144638
Name: median_house_value, dtype: float64
```

In [8]:

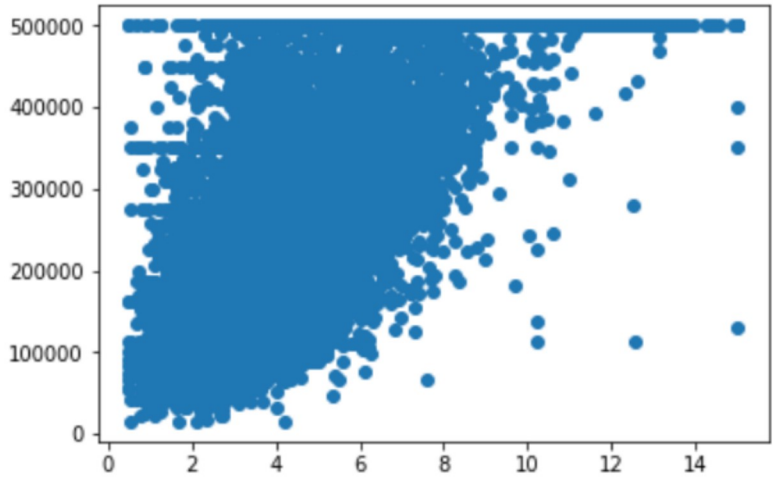
```
corr_cols=d1.corr()['median_house_value'].sort_values()[::-1]
#corr_cols[1:4]
corr_cols.index
```

Out[8]:

```
Index(['median_house_value', 'median_income', 'total_rooms',
       'housing_median_age', 'households', 'total_bedrooms', 'population',
       'longitude', 'latitude'],
      dtype='object')
```

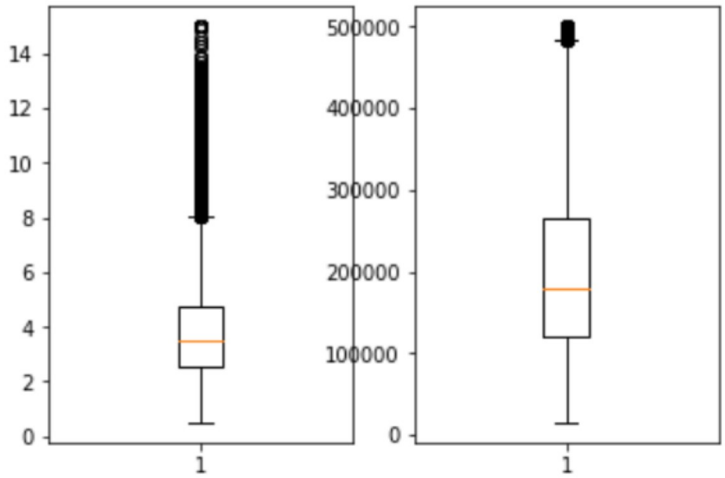
In [9]:

```
import matplotlib.pyplot as plt
plt.scatter(d1.median_income,d1.median_house_value)
plt.show()
```



In [10]:

```
plt.subplot(121)
plt.boxplot(d1.median_income)
plt.subplot(122)
plt.boxplot(d1.median_house_value)
plt.show()
```



In [11]:

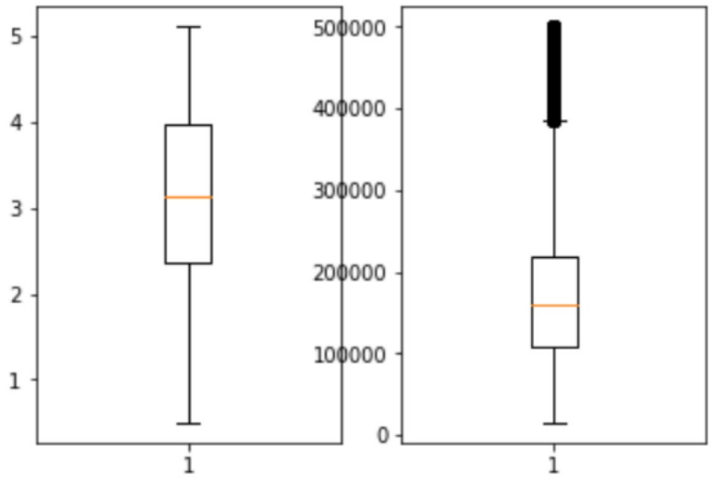
```
d2=d1[d1.median_income<d1.median_income.quantile(0.8)]
d2.shape
```

Out[11]:

(16346, 10)

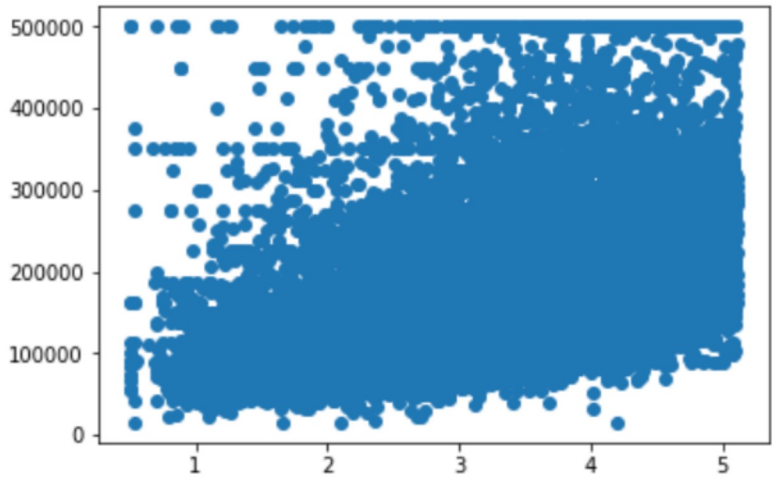
In [12]:

```
plt.subplot(121)
plt.boxplot(d2.median_income)
plt.subplot(122)
plt.boxplot(d2.median_house_value)
plt.show()
```



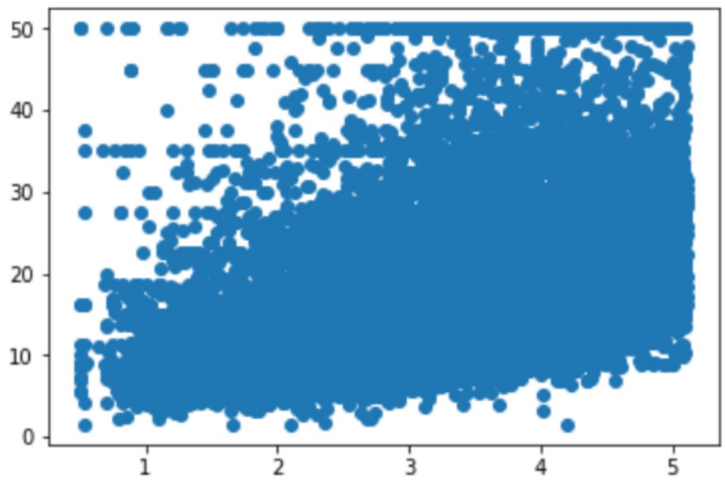
In [13]:

```
import matplotlib.pyplot as plt
plt.scatter(d2.median_income,d2.median_house_value)
plt.show()
```



In [14]:

```
import matplotlib.pyplot as plt
plt.scatter(d2.median_income,d2.median_house_value/10000)
plt.show()
```



In [35]:

```
d3=d2.drop(corr_cols.index[2:],axis=1)
d3=d3.drop(['ocean_proximity'],axis=1)
print(d3.shape)
d3
d3.median_house_value = d3.median_house_value/100000
print(d3)
```

```
(16346, 2)
       median_income  median_house_value
4             3.8462               3.422
5             4.0368               2.697
6             3.6591               2.992
7             3.1200               2.414
8             2.0804               2.267
9             3.6912               2.611
10            3.2031               2.815
11            3.2705               2.418
12            3.0750               2.135
13            2.6736               1.913
14            1.9167               1.592
15            2.1250               1.400
16            2.7750               1.525
17            2.1202               1.555
18            1.9911               1.587
19            2.6033               1.629
20            1.3578               1.475
21            1.7135               1.598
22            1.7250               1.139
23            2.1806               0.997
24            2.6000               1.326
25            2.4038               1.075
26            2.4597               0.938
27            1.8080               1.055
28            1.6424               1.089
29            1.6875               1.320
30            1.9274               1.223
31            1.9615               1.152
32            1.7969               1.104
33            1.3750               1.049
...              ...                 ...
20610         1.3631               0.455
20611         1.2857               0.470
20612         1.4934               0.483
20613         1.4958               0.534
20614         2.4695               0.580
20615         2.3598               0.575
20616         2.0469               0.551
20617         3.3021               0.708
20618         2.2500               0.634
20619         2.7303               0.991
20620         4.5625               1.000
20621         2.3661               0.775
20622         2.4167               0.670
20623         2.8235               0.655
20624         3.0739               0.872
20625         4.1250               0.720
20626         2.1667               0.938
```

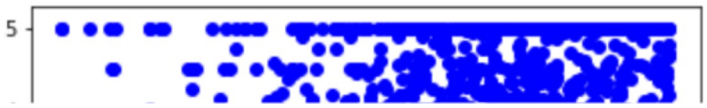
```
20628         2.5952              0.924
20629         2.0943              1.083
20630         3.5673              1.120
20631         3.5179              1.072
20632         3.1250              1.156
20633         2.5495              0.983
20634         3.7125              1.168
20635         1.5603              0.781
20636         2.5568              0.771
20637         1.7000              0.923
20638         1.8672              0.847
20639         2.3886              0.894

[16346 rows x 2 columns]
```

In [51]:

```
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error

plt.scatter(d3["median_income"],d3["median_house_value"],color='b')



#print(sample1.head(1))
testsize=(int)(d3.shape[0]*0.30)

train=d3[:-testsize]
    #print(train.shape)
test=d3[-testsize:]
    #print(test.shape)

train_x=train['median_income']
train_x=train_x[:,np.newaxis]
train_y=train['median_house_value']
train_y=train_y[:,np.newaxis]
test_x = test['median_income']
test_x = test_x[:,np.newaxis]
test_y=test['median_house_value']
test_y=test_y[:,np.newaxis]
train_x.shape

from sklearn.preprocessing import PolynomialFeatures
poly_features = PolynomialFeatures(degree=3, include_bias=True)
X_train_poly = poly_features.fit_transform(train_x)
X_test_poly = poly_features.fit_transform(test_x)
print(train_x,X_poly)

print(".......",X_train_poly.shape,train_y.shape,X_test_poly.shape,test_y.shape)
lm1=linear_model.LinearRegression()
lm1.fit(X_train_poly,train_y)
train_pred=lm1.predict(X_train_poly)
#train_y-train_pred

#plt.scatter(train.median_income,train.median_house_value)
test_pred=lm1.predict(X_test_poly)
    #plt.scatter(test_x,test_y)
plt.plot(test_x,test_pred,'r+')

print("MSE : ",mean_squared_error(test_y,test_pred), "\tRMSE:",np.sqrt(mean_squared_error(t
    #print(sample1.median_income.size,train.median_income.size)
plt.show()
```

```
[[ 3.8462]
 [ 4.0368]
 [ 3.6591]
 ...,
 [ 1.75  ]
 [ 1.9583]
 [ 1.7   ]] [[  3.8462      14.79325444  56.89781523]
 [  4.0368      16.29575424  65.78270072]
 [  3.6591      13.38901281  48.99173677]
```

```
 [  1.9583       3.83493889   7.50996083]
 [  1.7          2.89         4.913     ]]
....... (11443, 4) (11443, 1) (4903, 4) (4903, 1)
MSE :  0.844777273557   RMSE: 0.919117660344
```



In [26]:

```
import matplotlib.pyplot as plt
plt.boxplot(d3.total_bedrooms)
plt.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-26-e1b635c37c1d> in <module>()
      1 import matplotlib.pyplot as plt
----> 2 plt.boxplot(d3.total_bedrooms)
      3 plt.show()

~\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, na
me)
   3079             if name in self._info_axis:
   3080                 return self[name]
-> 3081             return object.__getattribute__(self, name)
   3082
   3083     def __setattr__(self, name, value):

AttributeError: 'DataFrame' object has no attribute 'total_bedrooms'
```

In [ ]: