In [2]:

```
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn import preprocessing, svm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
df = pd.read_csv('C:/Users/Deep/Desktop/Regression/USA_Housing.csv')
```

In [3]:

```
#Make a new Dataset considering only ‘Avg. Area House Age’ and ‘Price’ attributes

df_binary = df[['Avg. Area House Age', 'Price']].copy()
df_binary_columns = ['Avg. Area House Age', 'Price']
df_binary.head()
```

Out[3]:

	Avg. Area House Age	Price
0	5.682861	1.059034e+06
1	6.002900	1.505891e+06
2	5.865890	1.058988e+06
3	7.188236	1.260617e+06
4	5.040555	6.309435e+05

In [4]:

```
#Set a variable ‘x’ equal to the ‘Avg. Area House Age’ feature of the given dataset and a variable ‘y’ equal to t
he ‘Price’ column

x = df['Avg. Area House Age']
y = df['Price']
```

In [5]:

```
x
```

Out[5]:

```
0       5.682861
1       6.002900
2       5.865890
3       7.188236
4       5.040555
          ...
4995    7.830362
4996    6.999135
4997    7.250591
4998    5.534388
4999    5.992305
Name: Avg. Area House Age, Length: 5000, dtype: float64
```

In [6]:

```
y
```

Out[6]:

```
0       1.059034e+06
1       1.505891e+06
2       1.058988e+06
3       1.260617e+06
4       6.309435e+05
            ...
4995    1.060194e+06
4996    1.482618e+06
4997    1.030730e+06
4998    1.198657e+06
4999    1.298950e+06
Name: Price, Length: 5000, dtype: float64
```

In [7]:

```
#Split the new dataset into the Training set and Test set such that Test set contains 1/3 of total records using
model_selection.train_test_split from sklearn.

X = np.array(df_binary['Avg. Area House Age']).reshape(-1,1)
Y = np.array(df_binary['Price']).reshape(-1,1)
#df_binary.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.33)
```

In [8]:

```
#Train the Simple Linear Regression model based on the Training set after importing LinearRegression from sklearn
.linear_model.

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[8]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [11]:

```
#Display the model's coefficients after train the model and hence write the final regression model.

print(lr.score(x_test,y_test))
lr.coef_
lr.intercept_

m=lr.intercept_
b=lr.coef_
print('y=',m,'*X+',b)
```

```
0.20788514316997497
y= [275711.39245635] *X+ [[160459.41126494]]
```

In [13]:

```
#Predict the house price of the Test set data and display them

pred = lr.predict(x_test)
print(pred)
```

```
[[1281934.66709392]
 [1102774.521869  ]
 [1515907.23603551]
 ...
 [1165954.4454663 ]
 [1174112.92060321]
 [1555202.38917464]]
```

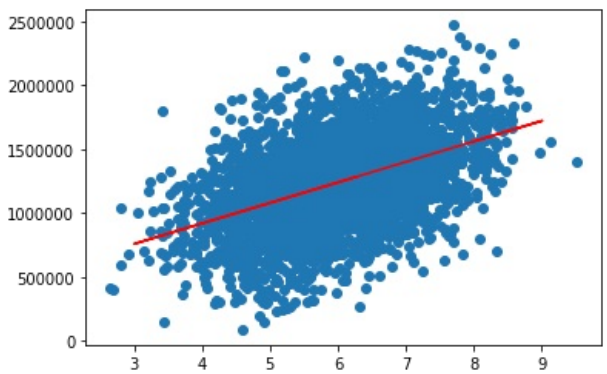
In [17]:

```
#Visualize the Training set results using scatter plot along with regression line.

plt.scatter(x_train,y_train)
plt.plot(x_test,pred,color='red')
```

Out[17]:

```
[<matplotlib.lines.Line2D at 0x17c9e2f4088>]
```



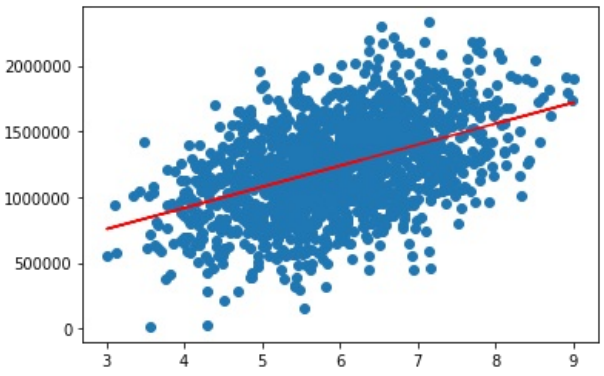
In [19]:

```
#Visualize the Test set results using scatter plot along with regression line

plt.scatter(x_test,y_test)
plt.plot(x_test,pred,color='red')
```

Out[19]:

```
[<matplotlib.lines.Line2D at 0x17c9ea47ec8>]
```



In [ ]: