In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
%matplotlib inline
```

In [4]:

```
USAHousing = pd.read_csv(r'C:\Users\DELL\Downloads\3PythonCourse\Refactored_Py_DS_ML_Bootcamp-ma
```

In [5]:

```
USAHousing.head()
```

Out[5]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addr
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry 674\nLaurabury, 37(
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Vi Suite 079\nL Kathleen, C
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizal Stravenue\nDanielto WI 064
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPC 44
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nF AE 09

In [6]:

```
USAHousing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
Avg. Area Income              5000 non-null float64
Avg. Area House Age           5000 non-null float64
Avg. Area Number of Rooms     5000 non-null float64
Avg. Area Number of Bedrooms  5000 non-null float64
Area Population               5000 non-null float64
Price                         5000 non-null float64
Address                       5000 non-null object
dtypes: float64(6), object(1)
memory usage: 273.5+ KB
```

In [7]:

```
USAHousing.describe()
```

Out[7]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

In [9]:

```
USAHousing.isnull().sum()
```

Out[9]:

```
Avg. Area Income              0
Avg. Area House Age           0
Avg. Area Number of Rooms     0
Avg. Area Number of Bedrooms  0
Area Population               0
Price                         0
Address                       0
dtype: int64
```

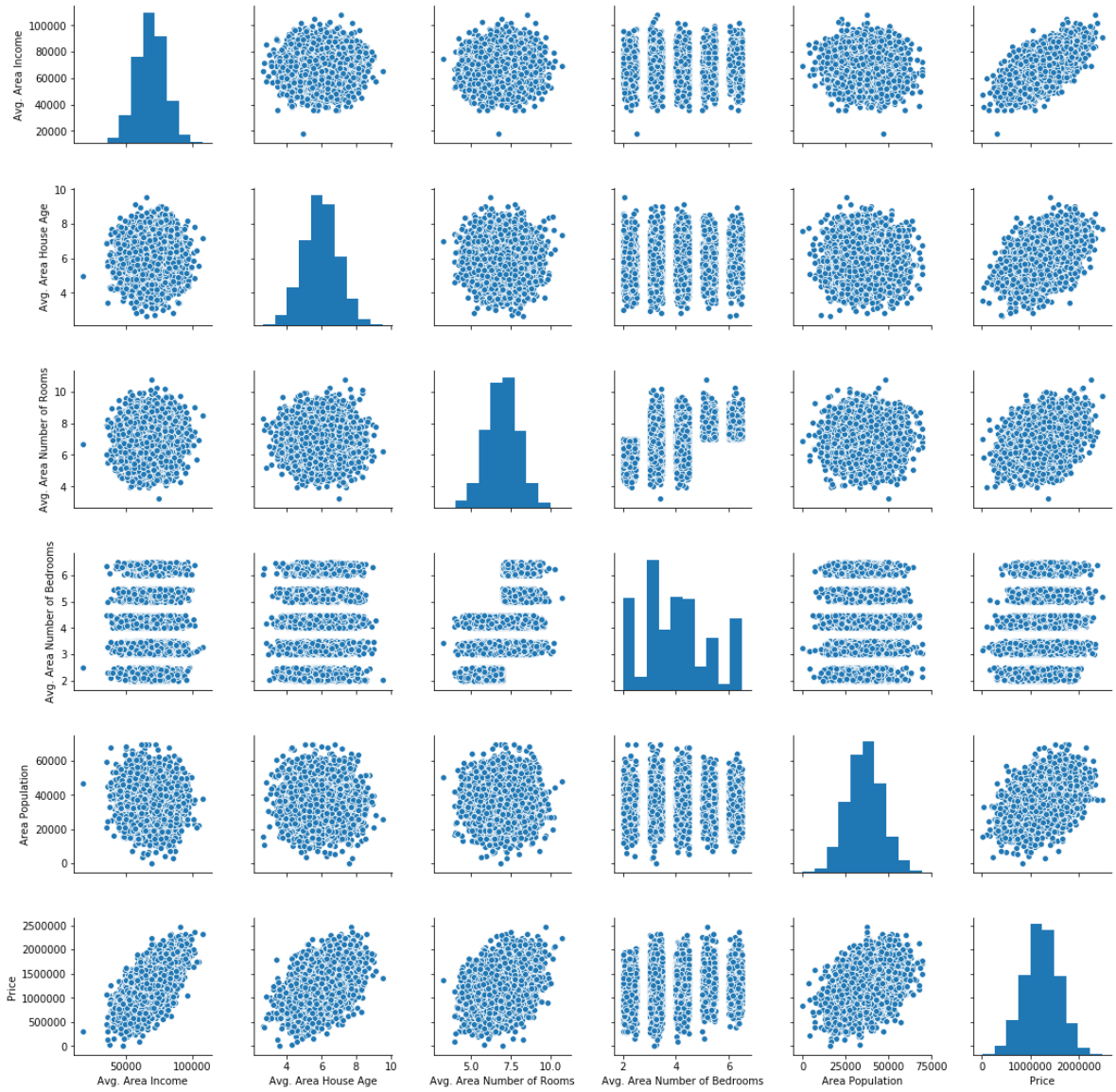
EDA

In [10]:

```
sns.pairplot(USAHousing)
```

Out[10]:

<seaborn.axisgrid.PairGrid at 0x20c57b4fc88>



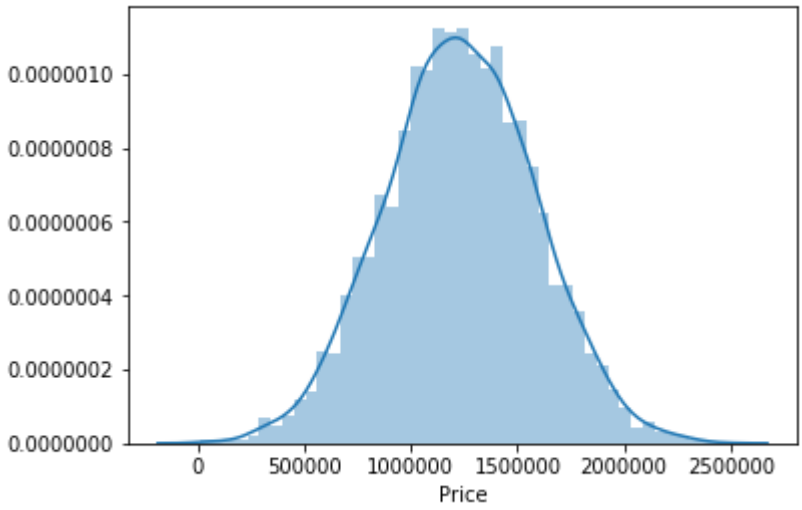
In [11]:

```
sns.distplot(USAHousing['Price'])
```

C:\Users\DELL\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Out[11]:

<matplotlib.axes._subplots.AxesSubplot at 0x20c59d3b8d0>

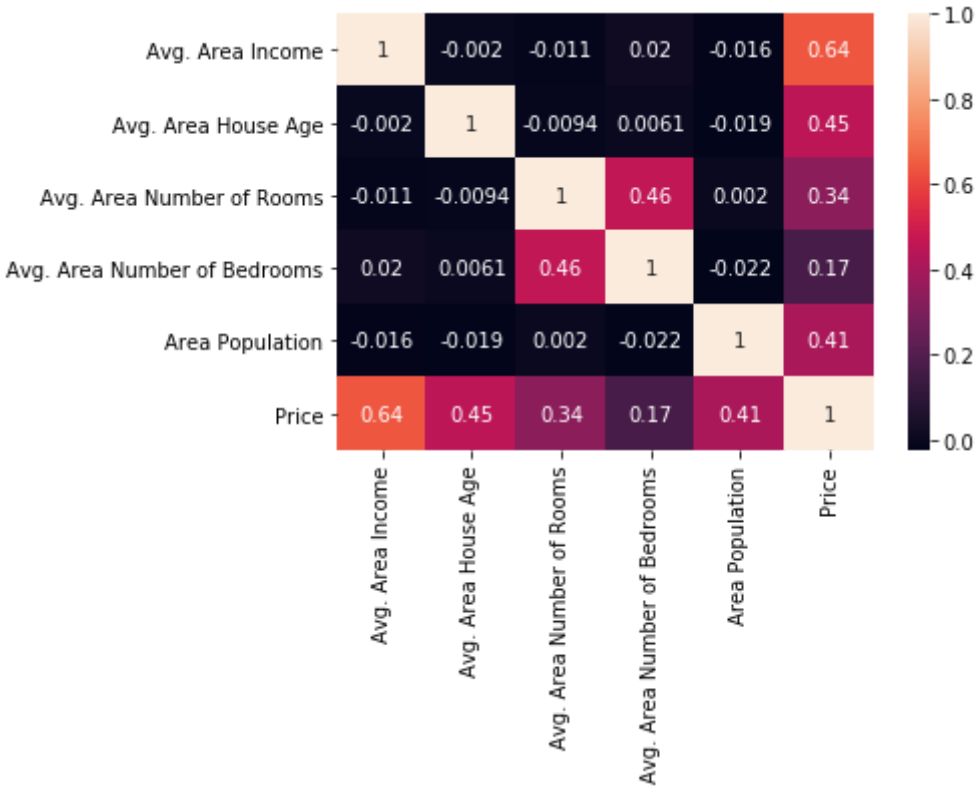


In [15]:

```
sns.heatmap(USAHousing.corr(), annot = True)
```

Out[15]:

<matplotlib.axes._subplots.AxesSubplot at 0x20c5ab4cef0>



Splitting the dataset

In [16]:

```
USAHousing.columns
```

Out[16]:

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [17]:

```
X = USAHousing[['Avg. Area Income','Avg. Area House Age','Avg. Area Number of Rooms','Avg. Area Number of
```

In [18]:

```
y = USAHousing['Price']
```

In [19]:

```
from sklearn.model_selection import train_test_split
```

In [20]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 101)
```

Training the model(Test data)

In [23]:

```
from sklearn.linear_model import LinearRegression
```

In [24]:

```
lm = LinearRegression()
```

In [25]:

```
lm.fit(X_train, y_train)
```

Out[25]:

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

Model Evaluation

In [30]:

```
lm.intercept_
```

Out[30]:

```
-2640441.3997810436
```

In [32]:

```
df = pd.DataFrame(lm.coef_, X.columns)
df.columns = ['Coefficients']
df
```

Out[32]:

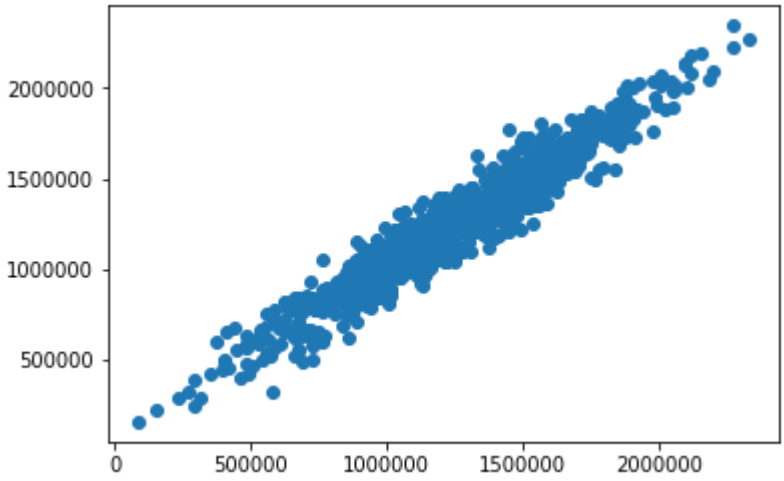
	Coefficients
Avg. Area Income	21.566696
Avg. Area House Age	165453.042478
Avg. Area Number of Rooms	120499.839093
Avg. Area Number of Bedrooms	1999.785336
Area Population	15.340604

In [33]:

```
prediction = lm.predict(X_test)
```

In [34]:

```
plt.scatter(y_test, prediction)
plt.show()
```



In [37]:

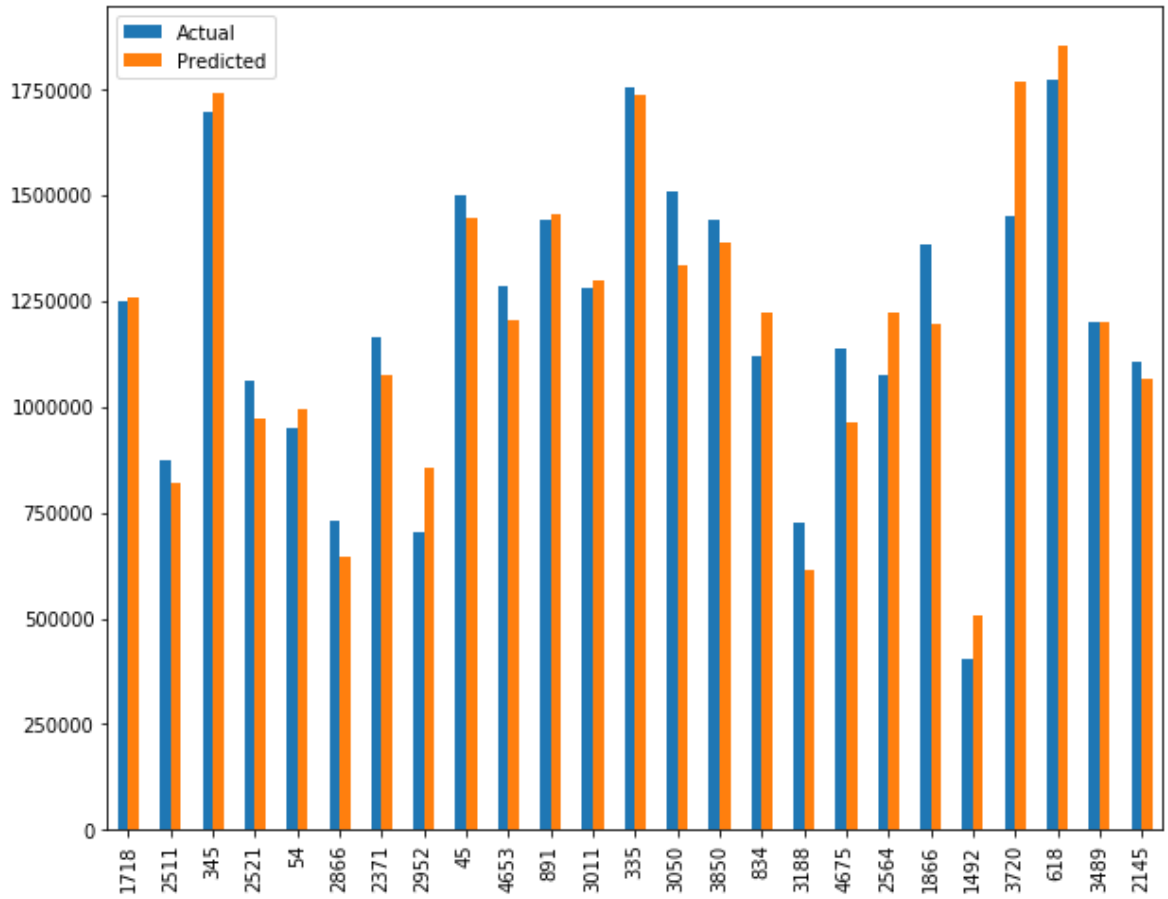
```
df1 = pd.DataFrame({'Actual': y_test, 'Predicted': prediction})
df2 = df1.head(25)
df2
```

Out[37]:

	Actual	Predicted
1718	1.251689e+06	1.257920e+06
2511	8.730483e+05	8.221124e+05
345	1.696978e+06	1.740669e+06
2521	1.063964e+06	9.724521e+05
54	9.487883e+05	9.934223e+05
2866	7.300436e+05	6.441261e+05
2371	1.166925e+06	1.073912e+06
2952	7.054441e+05	8.565840e+05
45	1.499989e+06	1.445318e+06
4653	1.288199e+06	1.204342e+06
891	1.441737e+06	1.455792e+06
3011	1.279681e+06	1.298557e+06
335	1.754969e+06	1.735924e+06
3050	1.511653e+06	1.336926e+06
3850	1.441956e+06	1.387637e+06
834	1.119993e+06	1.222404e+06
3188	7.278665e+05	6.137863e+05
4675	1.138885e+06	9.639335e+05
2564	1.074263e+06	1.221197e+06
1866	1.386473e+06	1.198072e+06
1492	4.046436e+05	5.058619e+05
3720	1.449829e+06	1.769107e+06
618	1.775875e+06	1.853881e+06
3489	1.202051e+06	1.200370e+06
2145	1.105737e+06	1.065129e+06

In [38]:

```
df2.plot(kind = 'bar', figsize = (10,8))
plt.show()
```



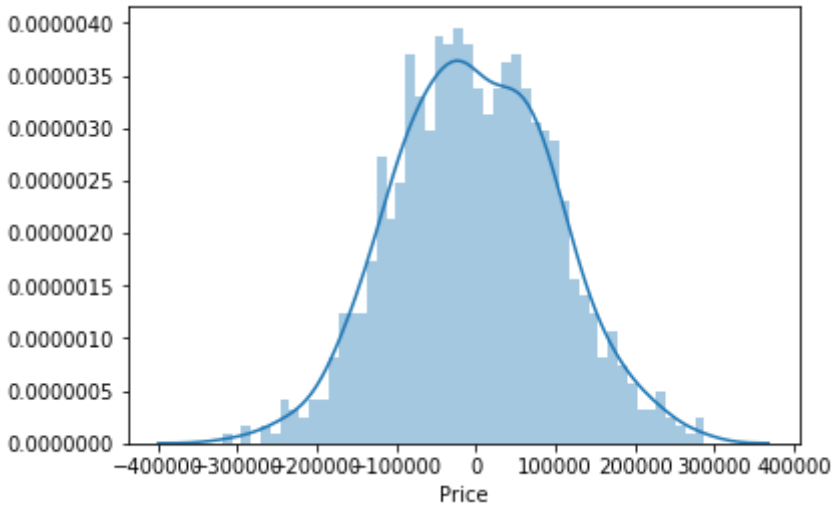
In [36]:

```
sns.distplot((y_test-prediction), bins = 50)
```

```
C:\Users\DELL\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a
non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of
`arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will r
esult either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20c5c235978>
```



Evaluating the Model

In [39]:

```
from sklearn import metrics
print('MAE',metrics.mean_absolute_error(y_test, prediction))
print('MSE', metrics.mean_squared_error(y_test, prediction))
print('RMSE',np.sqrt(metrics.mean_squared_error(y_test, prediction)))
```

```
MAE 81305.23300085348
MSE 10100187858.863457
RMSE 100499.69083964117
```

In [ ]:

```

```