In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score
%matplotlib inline
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.linear_model import Lasso
from sklearn import metrics
```

In [2]:
```
df= pd.read_csv('https://raw.githubusercontent.com/amankharwal/Website-data/master/CarPrice.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fuel
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	...	130	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	

5 rows × 26 columns

In [4]:
```
df.shape
```

Out[4]:
```
(205, 26)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   car_ID            205 non-null    int64
 1   symboling         205 non-null    int64
 2   CarName           205 non-null    object
 3   fueltype          205 non-null    object
 4   aspiration        205 non-null    object
 5   doornumber        205 non-null    object
 6   carbody           205 non-null    object
 7   drivewheel        205 non-null    object
 8   enginelocation    205 non-null    object
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    int64
 14  enginetype        205 non-null    object
 15  cylindernumber    205 non-null    object
 16  enginesize        205 non-null    int64
 17  fuelsystem        205 non-null    object
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64
 22  peakrpm           205 non-null    int64
 23  citympg           205 non-null    int64
 24  highwaympg        205 non-null    int64
 25  price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

In [6]:
```
df.describe()
```

Out[6]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressionr
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.00
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329756	3.255415	10.14
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.270844	0.313597	3.97
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.00
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.60
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.00
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	3.580000	3.410000	9.40
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.00

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
car_ID              0
symboling           0
CarName             0
fueltype            0
aspiration          0
doornumber          0
carbody             0
drivewheel          0
enginelocation      0
wheelbase           0
carlength           0
carwidth            0
carheight           0
curbweight          0
enginetype          0
cylindernumber      0
enginesize          0
fuelsystem          0
boreratio           0
stroke              0
compressionratio    0
horsepower          0
peakrpm             0
citympg             0
highwaympg          0
price               0
dtype: int64
```

In [8]:
```
df.isnull().values.any()
```

Out[8]:
```
False
```

In [9]:
```
df.head()
```

Out[9]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fuel
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	...	130	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	

5 rows × 26 columns

In [10]:
```
CompanyName=df['CarName'].apply(lambda x: x.split(' ')[0])
df.insert(3,'CompanyName',CompanyName)
df.drop(['CarName'],axis=1,inplace=True)
df.head()
```

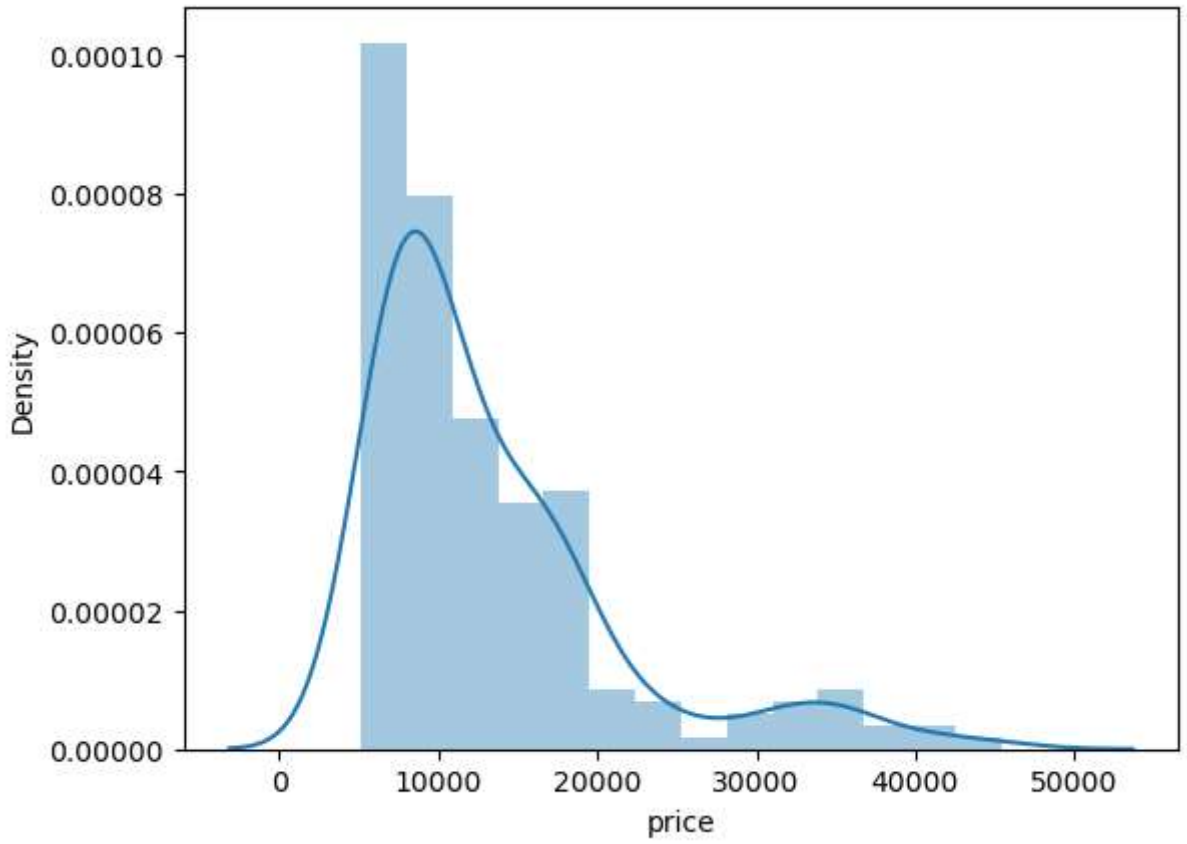
Out[10]:

	car_ID	symboling	CompanyName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	f
0	1	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	
1	2	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	
2	3	1	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	
3	4	2	audi	gas	std	four	sedan	fwd	front	99.8	...	109	
4	5	2	audi	gas	std	four	sedan	4wd	front	99.4	...	136	

5 rows × 26 columns

In [11]:
```
sns.distplot(df['price'])
plt.show()
```

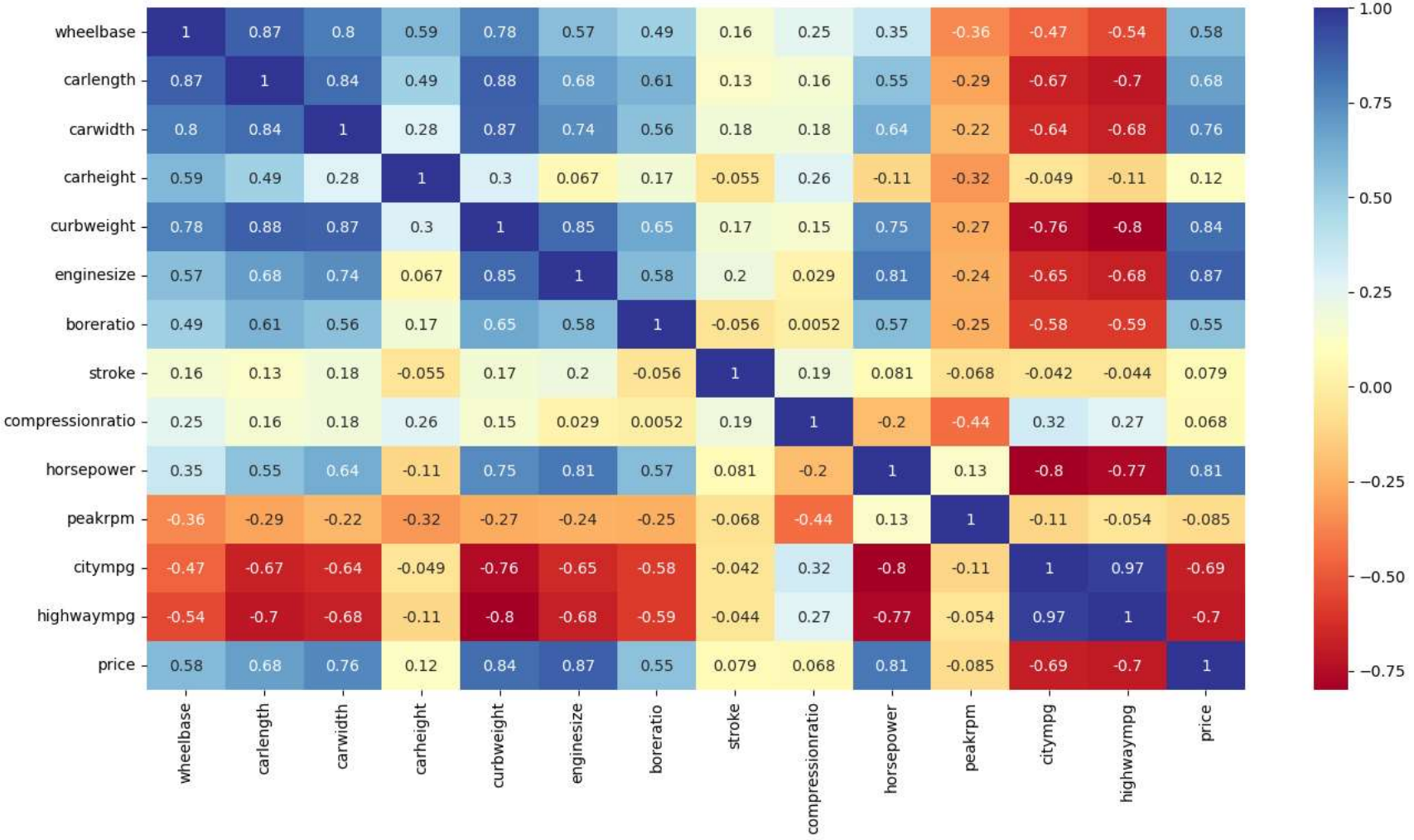
```
C:\Users\Hp\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function wit
h similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [12]:
```
plt.figure(figsize=(16,8))

data = df.select_dtypes(include=['float64', 'int'])
data = data.drop(['symboling', 'car_ID'], axis=1)
data = data.corr()

sns.heatmap(data, cmap="RdYlBu", annot=True)
plt.show()
```


wheelbase 1 0.87 0.8 0.59 0.78 0.57 0.49 0.16 0.25 0.35 -0.36 -0.47 -0.54 0.58
carlength 0.87 1 0.84 0.49 0.88 0.68 0.61 0.13 0.16 0.55 -0.29 -0.67 -0.7 0.68
carwidth 0.8 0.84 1 0.28 0.87 0.74 0.56 0.18 0.18 0.64 -0.22 -0.64 -0.68 0.76
carheight 0.59 0.49 0.28 1 0.3 0.067 0.17 -0.055 0.26 -0.11 -0.32 -0.049 -0.11 0.12
curbweight 0.78 0.88 0.87 0.3 1 0.85 0.65 0.17 0.15 0.75 -0.27 -0.76 -0.8 0.84
enginesize 0.57 0.68 0.74 0.067 0.85 1 0.58 0.2 0.029 0.81 -0.24 -0.65 -0.68 0.87
boreratio 0.49 0.61 0.56 0.17 0.65 0.58 1 -0.056 0.0052 0.57 -0.25 -0.58 -0.59 0.55
stroke 0.16 0.13 0.18 -0.055 0.17 0.2 -0.056 1 0.19 0.081 -0.068 -0.042 -0.044 0.079
compressionratio 0.25 0.16 0.18 0.26 0.15 0.029 0.0052 0.19 1 -0.2 -0.44 0.32 0.27 0.068
horsepower 0.35 0.55 0.64 -0.11 0.75 0.81 0.57 0.081 -0.2 1 0.13 -0.8 -0.77 0.81
peakrpm -0.36 -0.29 -0.22 -0.32 -0.27 -0.24 -0.25 -0.068 -0.44 0.13 1 -0.11 -0.054 -0.085
citympg -0.47 -0.67 -0.64 -0.049 -0.76 -0.65 -0.58 -0.042 0.32 -0.8 -0.11 1 0.97 -0.69
highwaympg -0.54 -0.7 -0.68 -0.11 -0.8 -0.68 -0.59 -0.044 0.27 -0.77 -0.054 0.97 1 -0.7
price 0.58 0.68 0.76 0.12 0.84 0.87 0.55 0.079 0.068 0.81 -0.085 -0.69 -0.7 1
wheelbase carlength carwidth carheight curbweight enginesize boreratio stroke compressionratio horsepower peakrpm citympg highwaympg price
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75

In [13]:
```
df.head()
```

Out[13]:

	car_ID	symboling	CompanyName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize
0	1	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130
1	2	3	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130
2	3	1	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152
3	4	2	audi	gas	std	four	sedan	fwd	front	99.8	...	109
4	5	2	audi	gas	std	four	sedan	4wd	front	99.4	...	136

5 rows × 26 columns

In [14]:
```
df.columns
```

Out[14]:
```
Index(['car_ID', 'symboling', 'CompanyName', 'fueltype', 'aspiration',
       'doornumber', 'carbody', 'drivewheel', 'enginelocation', 'wheelbase',
       'carlength', 'carwidth', 'carheight', 'curbweight', 'enginetype',
       'cylindernumber', 'enginesize', 'fuelsystem', 'boreratio', 'stroke',
       'compressionratio', 'horsepower', 'peakrpm', 'citympg', 'highwaympg',
       'price'],
      dtype='object')
```

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
car_ID              0
symboling           0
CompanyName         0
fueltype            0
aspiration          0
doornumber          0
carbody             0
drivewheel          0
enginelocation      0
wheelbase           0
carlength           0
carwidth            0
carheight           0
curbweight          0
enginetype          0
cylindernumber      0
enginesize          0
fuelsystem          0
boreratio           0
stroke              0
compressionratio    0
horsepower          0
peakrpm             0
citympg             0
highwaympg          0
price               0
dtype: int64
```

In [16]:
```
# checking the distribution of categorical data
print(df.fueltype.value_counts())
print(df.aspiration.value_counts())
print(df.carbody.value_counts())
print(df.drivewheel.value_counts())
print(df.enginetype.value_counts())
print(df.cylindernumber.value_counts())
print(df.doornumber.value_counts())
print(df.fuelsystem.value_counts())
```

```
gas       185
diesel     20
Name: fueltype, dtype: int64
std      168
turbo     37
Name: aspiration, dtype: int64
sedan          96
hatchback      70
wagon          25
hardtop         8
convertible     6
Name: carbody, dtype: int64
fwd    120
rwd     76
4wd      9
Name: drivewheel, dtype: int64
ohc      148
ohcf      15
ohcv      13
dohc      12
l         12
rotor      4
dohcv      1
Name: enginetype, dtype: int64
four      159
six        24
five       11
eight       5
two         4
three       1
twelve      1
Name: cylindernumber, dtype: int64
four    115
two      90
Name: doornumber, dtype: int64
mpfi    94
2bbl    66
idi     20
1bbl    11
spdi     9
4bbl     3
mfi      1
spfi     1
Name: fuelsystem, dtype: int64
```

In [17]:
```
# encoding "FuelType" Column
df.replace({'fueltype':{'gas':0,'diesel':1}},inplace=True)
```

In [18]:
```
# encoding "aspiration" Column
df.replace({'aspiration':{'std':0,'turbo':1}},inplace=True)
```

In [19]:
```
# encoding "carbody" Column
df.replace({'carbody':{'sedan':0,'hatchback':1,'wagon':2,'hardtop':3,'convertible':4}},inplace=True)
```

In [20]:
```
# encoding "drivewheel" Column
df.replace({'drivewheel':{'fwd':0,'rwd':1,'4wd':2}},inplace=True)
```

In [21]:
```
# encoding "enginetype" Column
df.replace({'enginetype':{'ohc':0,'ohcf':1,'ohcv':2,'dohc':3,'l':4,'rotor':5,'dohcv':6}},inplace=True)
```

In [22]:
```
# encoding "cylindernumber" Column
df.replace({'cylindernumber':{'four':0,'six':1,'five':2,'eight':3,'two':4,'three':5,'twelve':6}},inplace=True)
```

In [23]:
```
# encoding "doornumber" Column
df.replace({'doornumber':{'four':0,'two':1}},inplace=True)
```

In [24]:
```
# encoding "fuelsystem" Column
df.replace({'fuelsystem':{'mpfi':0,'2bbl':1,'idi':2,'1bbl':3,'spdi':4,'4bbl':5,'mfi':6,'spfi':7}},inplace=True)
```

In [25]:
```
df.head()
```

Out[25]:

	car_ID	symboling	CompanyName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fue
0	1	3	alfa-romero	0	0	1	4	1	front	88.6	...	130	
1	2	3	alfa-romero	0	0	1	4	1	front	88.6	...	130	
2	3	1	alfa-romero	0	0	1	1	1	front	94.5	...	152	
3	4	2	audi	0	0	0	0	0	front	99.8	...	109	
4	5	2	audi	0	0	0	0	2	front	99.4	...	136	

5 rows × 26 columns

In [26]:
```
X = df.drop(['CompanyName','enginelocation','price'],axis=1)
Y = df['price']
```

In [27]:
```
print(X)
```

```
     car_ID  symboling  fueltype  aspiration  doornumber  carbody  drivewheel  \
0         1          3         0           0           1        4           1
1         2          3         0           0           1        4           1
2         3          1         0           0           1        1           1
3         4          2         0           0           0        0           0
4         5          2         0           0           0        0           2
..      ...        ...       ...         ...         ...      ...         ...
200     201         -1         0           0           0        0           1
201     202         -1         0           1           0        0           1
202     203         -1         0           0           0        0           1
203     204         -1         1           1           0        0           1
204     205         -1         0           1           0        0           1

     wheelbase  carlength  carwidth  ...  cylindernumber  enginesize  \
0         88.6      168.8      64.1  ...               0         130
1         88.6      168.8      64.1  ...               0         130
2         94.5      171.2      65.5  ...               1         152
3         99.8      176.6      66.2  ...               0         109
4         99.4      176.6      66.4  ...               2         136
..         ...        ...       ...  ...             ...         ...
200      109.1      188.8      68.9  ...               0         141
201      109.1      188.8      68.8  ...               0         141
202      109.1      188.8      68.9  ...               1         173
203      109.1      188.8      68.9  ...               1         145
204      109.1      188.8      68.9  ...               0         141

     fuelsystem  boreratio  stroke  compressionratio  horsepower  peakrpm  \
0             0       3.47    2.68               9.0         111     5000
1             0       3.47    2.68               9.0         111     5000
2             0       2.68    3.47               9.0         154     5000
3             0       3.19    3.40              10.0         102     5500
4             0       3.19    3.40               8.0         115     5500
..          ...        ...     ...               ...         ...      ...
200           0       3.78    3.15               9.5         114     5400
201           0       3.78    3.15               8.7         160     5300
202           0       3.58    2.87               8.8         134     5500
203           2       3.01    3.40              23.0         106     4800
204           0       3.78    3.15               9.5         114     5400

     citympg  highwaympg
0         21          27
1         21          27
2         19          26
3         24          30
4         18          22
```

```
..       ...         ...
200       23          28
201       19          25
202       18          23
203       26          27
204       19          25

[205 rows x 23 columns]
```

In [28]:
```
print(Y)
```

```
0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
```

In [29]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state=5)
```

In [30]:
```
# loading the linear regression model
lin_reg_model = LinearRegression()
```

In [31]:
```
lin_reg_model.fit(X_train,Y_train)
```

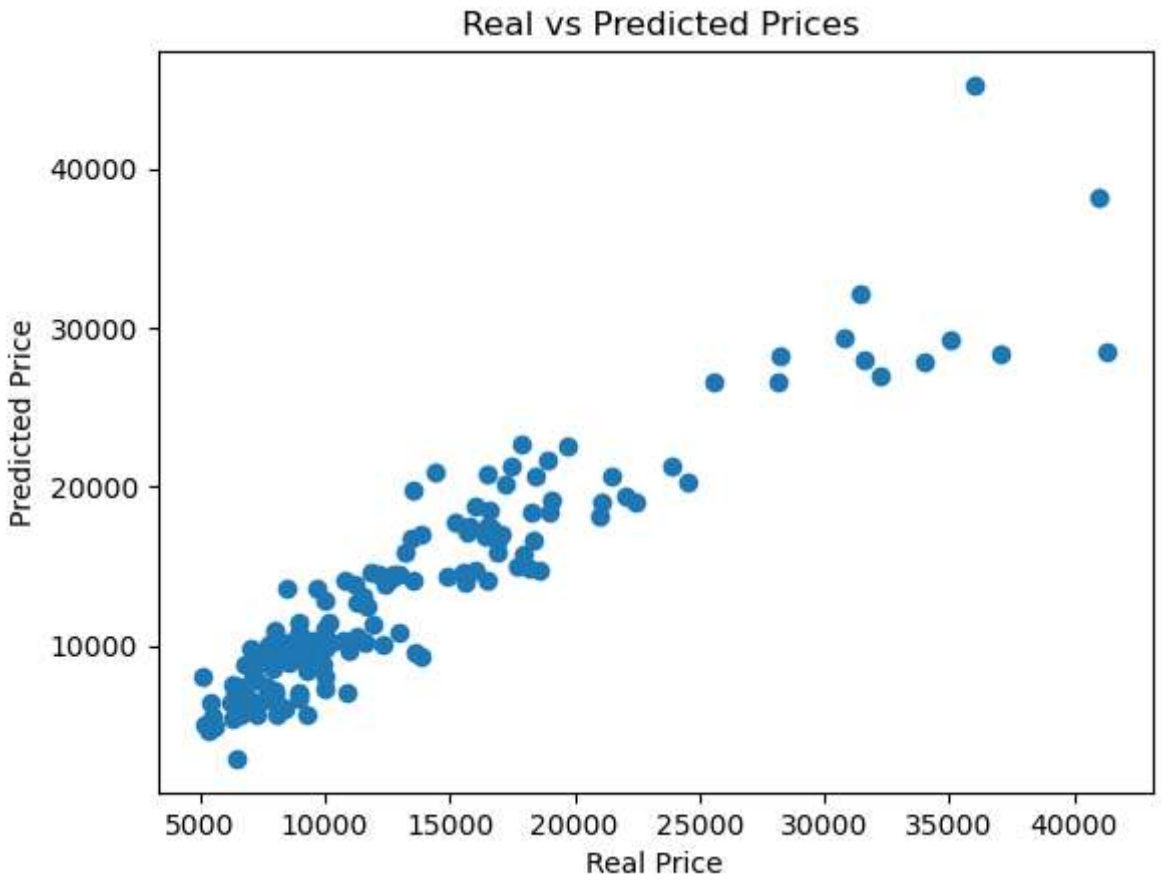
Out[31]:
```
LinearRegression()
```

In [32]:
```
# prediction on Training data
training_data_prediction = lin_reg_model.predict(X_train)
error_score = metrics.r2_score(Y_train, training_data_prediction)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.8760853857144774
```

In [33]:
```
plt.scatter(Y_train, training_data_prediction)
plt.xlabel("Real Price")
plt.ylabel("Predicted Price")
```

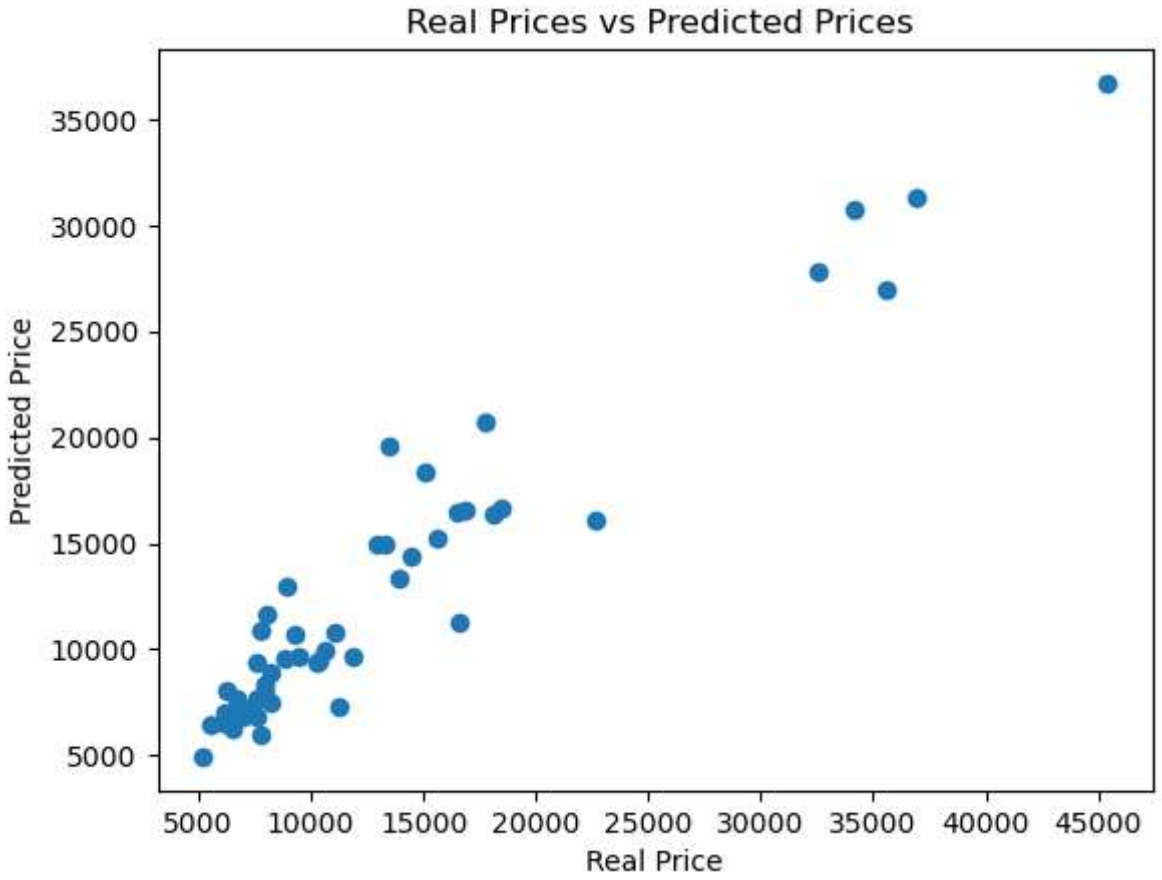
```
plt.title(" Real vs Predicted Prices")
plt.show()
```



In [34]:
```
# prediction on Training data
test_data_prediction = lin_reg_model.predict(X_test)
# R squared Error
error_score = metrics.r2_score(Y_test, test_data_prediction)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.8934204393960505
```

In [35]:
```
plt.scatter(Y_test, test_data_prediction)
plt.xlabel("Real Price")
plt.ylabel("Predicted Price")
plt.title(" Real Prices vs Predicted Prices")
plt.show()
```



In [36]:
```
# loading the linear regression model
lass_reg_model = Lasso()
lass_reg_model.fit(X_train,Y_train)
```

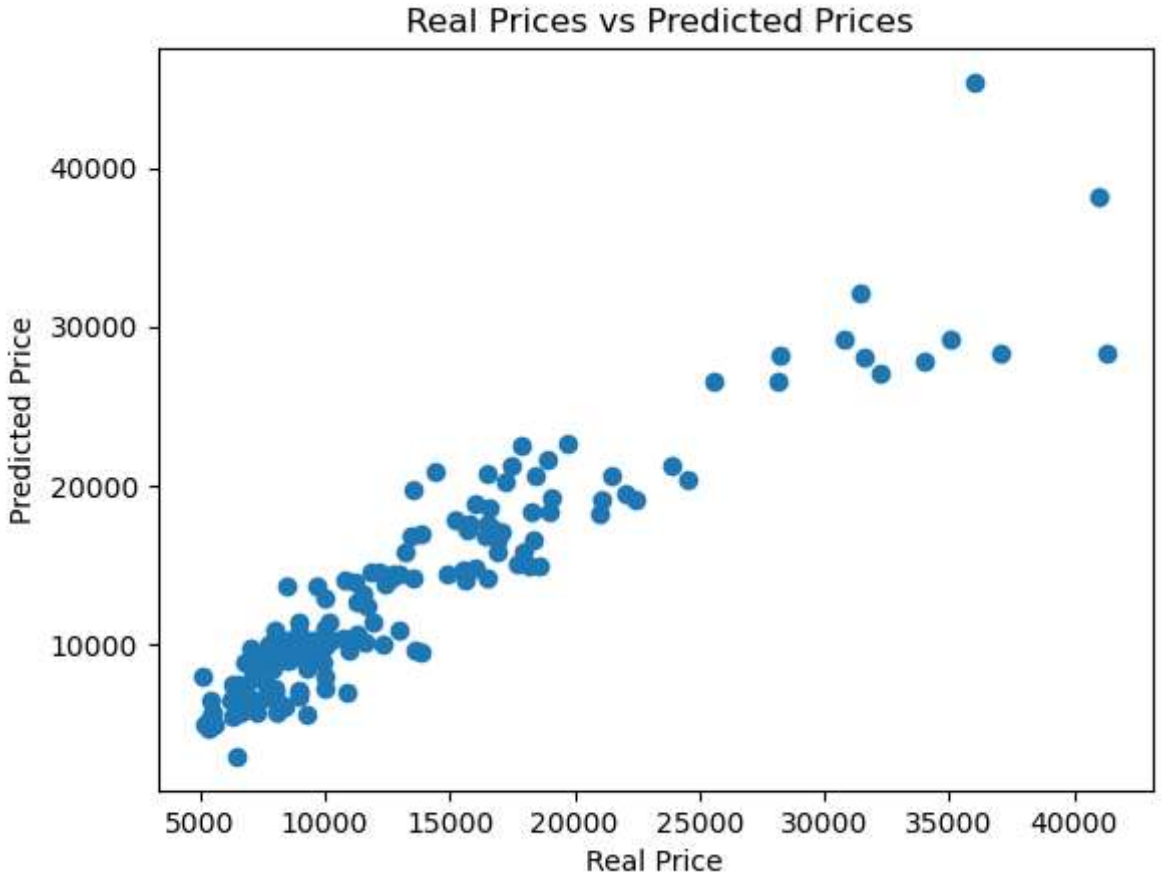
```
C:\Users\Hp\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:647: ConvergenceWarning: Objective
did not converge. You might want to increase the number of iterations, check the scale of the features or consider incr
easing regularisation. Duality gap: 4.237e+06, tolerance: 8.930e+05
  model = cd_fast.enet_coordinate_descent(
```

Out[36]:
```
Lasso()
```

In [37]:
```
# prediction on Training data
training_data_prediction = lass_reg_model.predict(X_train)
# R squared Error
error_score = metrics.r2_score(Y_train, training_data_prediction)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.8760556619393252
```

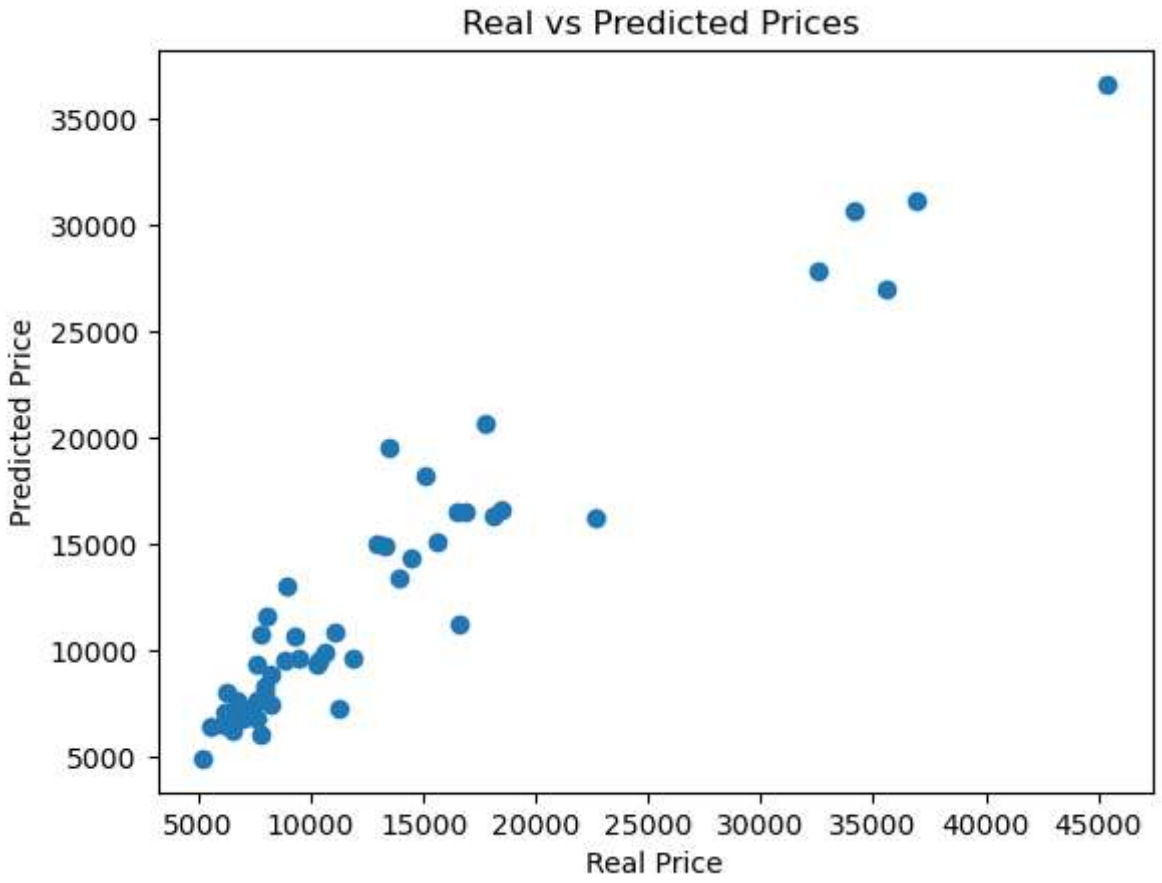
In [38]:
```
plt.scatter(Y_train, training_data_prediction)
plt.xlabel("Real Price")
plt.ylabel("Predicted Price")
plt.title(" Real Prices vs Predicted Prices")
plt.show()
```



In [39]:
```
# prediction on Training data
test_data_prediction = lass_reg_model.predict(X_test)
# R squared Error
error_score = metrics.r2_score(Y_test, test_data_prediction)
print("R squared Error : ", error_score)
```

```
R squared Error :  0.8937058905380512
```

In [40]:
```
plt.scatter(Y_test, test_data_prediction)
plt.xlabel("Real Price")
plt.ylabel("Predicted Price")
plt.title(" Real vs Predicted Prices")
plt.show()
```



In [41]:
```
#LinearRegression
reg=LinearRegression()
reg.fit(X_train,Y_train)
Y_pred=reg.predict(X_test)
print("R2 score: ",r2_score(Y_test,Y_pred))
```

```
R2 score:  0.8934204393960505
```

Thanks

In [ ]: