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

In [2]:
```
df=pd.read_csv("CarPrice_Assignment.csv")
```

In [3]:
```
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 [4]:
```
df.corr()
```

Out[4]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	bore
car_ID	1.000000	-0.151621	0.129729	0.170636	0.052387	0.255960	0.071962	-0.033930	0.26
symboling	-0.151621	1.000000	-0.531954	-0.357612	-0.232919	-0.541038	-0.227691	-0.105790	-0.13
wheelbase	0.129729	-0.531954	1.000000	0.874587	0.795144	0.589435	0.776386	0.569329	0.48
carlength	0.170636	-0.357612	0.874587	1.000000	0.841118	0.491029	0.877728	0.683360	0.60
carwidth	0.052387	-0.232919	0.795144	0.841118	1.000000	0.279210	0.867032	0.735433	0.55
carheight	0.255960	-0.541038	0.589435	0.491029	0.279210	1.000000	0.295572	0.067149	0.17
curbweight	0.071962	-0.227691	0.776386	0.877728	0.867032	0.295572	1.000000	0.850594	0.64
enginesize	-0.033930	-0.105790	0.569329	0.683360	0.735433	0.067149	0.850594	1.000000	0.58
boreratio	0.260064	-0.130051	0.488750	0.606454	0.559150	0.171071	0.648480	0.583774	1.00

stroke	-0.160824	-0.008735	0.160959	0.129533	0.182942	-0.055307	0.168790	0.203129	-0.05
compressionratio	0.150276	-0.178515	0.249786	0.158414	0.181129	0.261214	0.151362	0.028971	0.00
horsepower	-0.015006	0.070873	0.353294	0.552623	0.640732	-0.108802	0.750739	0.809769	0.57
peakrpm	-0.203789	0.273606	-0.360469	-0.287242	-0.220012	-0.320411	-0.266243	-0.244660	-0.25
citympg	0.015940	-0.035823	-0.470414	-0.670909	-0.642704	-0.048640	-0.757414	-0.653658	-0.58
highwaympg	0.011255	0.034606	-0.544082	-0.704662	-0.677218	-0.107358	-0.797465	-0.677470	-0.58
price	-0.109093	-0.079978	0.577816	0.682920	0.759325	0.119336	0.835305	0.874145	0.55

In [5]:
```
df.drop(["drivewheel","stroke","doornumber","carbody","car_ID","wheelbase","cylindernumb
```

In [6]:
```
df.drop(["CarName","aspiration","enginelocation","carheight"],axis=1,inplace=True)
```

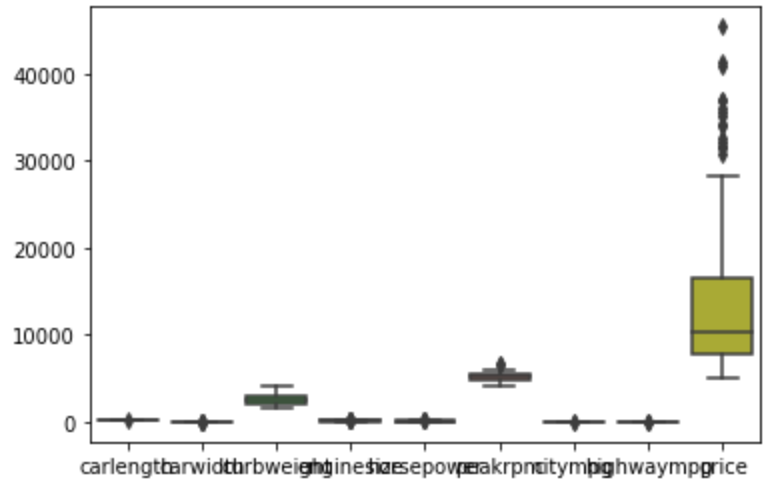
In [7]:
```
df
```

Out[7]:

	carlength	carwidth	curbweight	enginetype	enginesize	horsepower	peakrpm	citympg	highwaympg	
0	168.8	64.1	2548	dohc	130	111	5000	21	27	134
1	168.8	64.1	2548	dohc	130	111	5000	21	27	165
2	171.2	65.5	2823	ohcv	152	154	5000	19	26	165
3	176.6	66.2	2337	ohc	109	102	5500	24	30	139
4	176.6	66.4	2824	ohc	136	115	5500	18	22	174
...	...	...	...	...	...	...	...	...	...	
200	188.8	68.9	2952	ohc	141	114	5400	23	28	168
201	188.8	68.8	3049	ohc	141	160	5300	19	25	190
202	188.8	68.9	3012	ohcv	173	134	5500	18	23	214
203	188.8	68.9	3217	ohc	145	106	4800	26	27	224
204	188.8	68.9	3062	ohc	141	114	5400	19	25	226

205 rows × 10 columns

In [8]:
```
sns.boxplot(data=df);
```



In [9]:
```
from sklearn.preprocessing import OrdinalEncoder
ordi=OrdinalEncoder()
```

```
df["enginetype"]=ordi.fit_transform(df[["enginetype"]])
```

In [10]:
```
df.corr()
```

Out[10]:

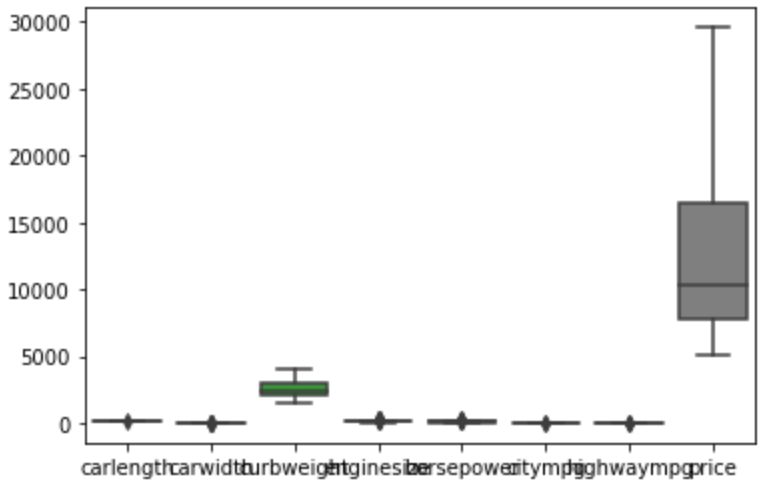
	carlength	carwidth	curbweight	enginetype	enginesize	horsepower	peakrpm	citympg	highwa
carlength	1.000000	0.841118	0.877728	-0.113291	0.683360	0.552623	-0.287242	-0.670909	-0.7
carwidth	0.841118	1.000000	0.867032	0.012298	0.735433	0.640732	-0.220012	-0.642704	-0.6
curbweight	0.877728	0.867032	1.000000	-0.055265	0.850594	0.750739	-0.266243	-0.757414	-0.7
enginetype	-0.113291	0.012298	-0.055265	1.000000	0.040766	0.010301	0.005599	-0.085004	-0.0
enginesize	0.683360	0.735433	0.850594	0.040766	1.000000	0.809769	-0.244660	-0.653658	-0.6
horsepower	0.552623	0.640732	0.750739	0.010301	0.809769	1.000000	0.131073	-0.801456	-0.7
peakrpm	-0.287242	-0.220012	-0.266243	0.005599	-0.244660	0.131073	1.000000	-0.113544	-0.0
citympg	-0.670909	-0.642704	-0.757414	-0.085004	-0.653658	-0.801456	-0.113544	1.000000	0.9
highwaympg	-0.704662	-0.677218	-0.797465	-0.078456	-0.677470	-0.770544	-0.054275	0.971337	1.0
price	0.682920	0.759325	0.835305	0.049171	0.874145	0.808139	-0.085267	-0.685751	-0.6

In [11]:
```
df.drop(["enginetype","peakrpm"],axis=1,inplace=True)
```

In [13]:
```
q1=np.quantile(df["price"],0.25)
q3=np.quantile(df["price"],0.75)
iqr=q3-q1
uw=q3+1.5*iqr
lw=q1-1.5*iqr
for i in df["price"]:
    if i>uw:
        df["price"]=df["price"].replace(i,uw)
```

In [14]:
```
sns.boxplot(data=df)
```

Out[14]:
```
<AxesSubplot:>
```



In [15]:
```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

In [16]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [17]:
```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
lr.fit(xtrain,ytrain)
ypred=lr.predict(xtest)
```

In [18]:
```
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(ytest,ypred)
rmse=np.sqrt(mse)
print(rmse)
```

```
2673.985276526457
```

In [19]:
```
lr.score(xtrain,ytrain)
```

Out[19]:
```
0.8271168025783082
```

In [20]:
```
lr.score(xtest,ytest)
```

Out[20]:
```
0.8283586405637202
```

In [ ]:
```

```