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("F:\\NEW_DATSET\\CarPrice_Assignment.csv")
df.shape
```

Out[2]:
```
(205, 26)
```

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.describe()
```

Out[4]:

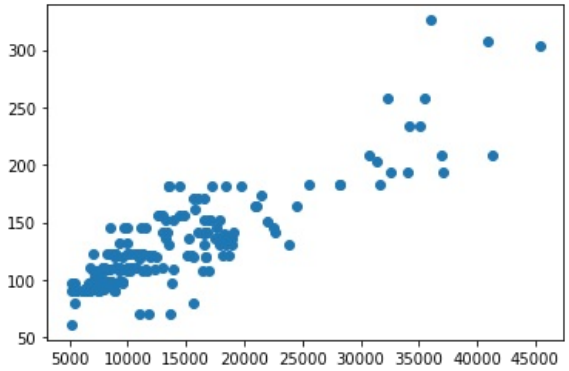
	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesize	boreratio	stroke	compressionr
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	3.329756	3.255415	10.142
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	0.270844	0.313597	3.972
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	3.150000	3.110000	8.600
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	3.580000	3.410000	9.400
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000

In [5]:
```
df.fuelsystem.unique()
```

Out[5]:
```
array(['mpfi', '2bbl', 'mfi', '1bbl', 'spfi', '4bbl', 'idi', 'spdi'],
      dtype=object)
```

In [6]:
```
plt.scatter(x=df.price,y=df.enginesize)
```

Out[6]: <matplotlib.collections.PathCollection at 0x24982789e20>

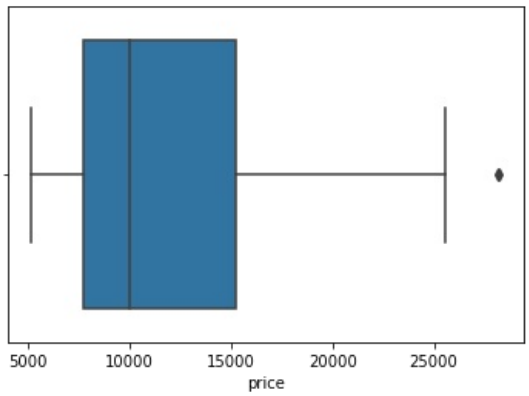


In [7]:
```
q1,q2=np.percentile(df['price'],25),np.percentile(df['price'],75)
cutt_off=(q2-q1)*1.5
lower,upper=1,q2+cutt_off
print(lower,upper)
df=df[(df['price']<upper)&(df['price']>lower)]
```

```
1 29575.5
```

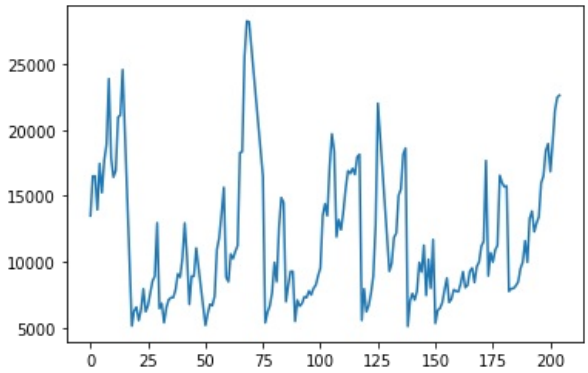
In [8]:
```
sns.boxplot(x='price',data=df)
```

Out[8]: <AxesSubplot:xlabel='price'>



In [9]:
```
df.price.plot()
```

Out[9]: <AxesSubplot:>



In [ ]:
```

```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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 [11]:
```
df=df.drop(['CarName'],axis=1)
```

In [12]:
```
df.fueltype.value_counts()
```

Out[12]:
```
gas       171
diesel     19
Name: fueltype, dtype: int64
```

In [13]:
```
fueltype=pd.get_dummies(df['fueltype'],drop_first=True)
fueltype.head()
```

Out[13]:

	gas
0	1
1	1
2	1
3	1
4	1

In [14]:
```
carbod=pd.get_dummies(df['carbody'],drop_first=True)
carbod.tail()
```

Out[14]:

	hardtop	hatchback	sedan	wagon
200	0	0	1	0
201	0	0	1	0
202	0	0	1	0
203	0	0	1	0
204	0	0	1	0

In [15]:
```
engineloc=pd.get_dummies(df['enginelocation'],drop_first=True)
drivewheels=pd.get_dummies(df['drivewheel'],drop_first=True)
drivewheels.head()
```

Out[15]:

	fwd	rwd
0	0	1

1	0	1
2	0	1
3	1	0
4	0	0

In [16]:
```
gassystem=pd.get_dummies(df['fuelsystem'],drop_first=True)
gassystem.head()
```

Out[16]:

	2bbl	4bbl	idi	mfi	mpfi	spdi	spfi
0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0
2	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0
4	0	0	0	0	1	0	0

In [17]:
```
doorno=pd.get_dummies(df['doornumber'],drop_first=True)
doorno.head()
```

Out[17]:

	two
0	1
1	1
2	1
3	0
4	0

In [18]:
```
df.head()
```

Out[18]:

	car_ID	symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	...	enginesize	fuelsystem
0	1	3	gas	std	two	convertible	rwd	front	88.6	168.8	...	130	mpfi
1	2	3	gas	std	two	convertible	rwd	front	88.6	168.8	...	130	mpfi
2	3	1	gas	std	two	hatchback	rwd	front	94.5	171.2	...	152	mpfi
3	4	2	gas	std	four	sedan	fwd	front	99.8	176.6	...	109	mpfi
4	5	2	gas	std	four	sedan	4wd	front	99.4	176.6	...	136	mpfi

5 rows × 25 columns

In [19]:
```
df=df.drop(['aspiration','carbody','enginelocation','drivewheel','fuelsystem','doornumber','fueltype'],axis=1)
df.head()
```

Out[19]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginetype	cylindernumber	enginesize	boreratio	stroke	compre
0	1	3	88.6	168.8	64.1	48.8	2548	dohc	four	130	3.47	2.68	
1	2	3	88.6	168.8	64.1	48.8	2548	dohc	four	130	3.47	2.68	
2	3	1	94.5	171.2	65.5	52.4	2823	ohcv	six	152	2.68	3.47	
3	4	2	99.8	176.6	66.2	54.3	2337	ohc	four	109	3.19	3.40	
4	5	2	99.4	176.6	66.4	54.3	2824	ohc	five	136	3.19	3.40	

In [20]:
```
cylinderno=pd.get_dummies(df['cylindernumber'],drop_first=True)
cylinderno.head()
```

Out[20]:

	four	six	three	two
0	1	0	0	0
1	1	0	0	0
2	0	1	0	0

3	1	0	0	0
4	0	0	0	0

In [21]:
```
df=pd.concat([cylinderno,df],axis=1)
df.isnull().sum()
```

Out[21]:
```
four               0
six                0
three              0
two                0
car_ID             0
symboling          0
wheelbase          0
carlength          0
carwidth           0
carheight          0
curbweight         0
enginetype         0
cylindernumber     0
enginesize         0
boreratio          0
stroke             0
compressionratio   0
horsepower         0
peakrpm            0
citympg            0
highwaympg         0
price              0
dtype: int64
```

In [22]:
```
df=df.drop(['cylindernumber'],axis=1)
df.head()
```

Out[22]:

	four	six	three	two	car_ID	symboling	wheelbase	carlength	carwidth	carheight	...	enginetype	enginesize	boreratio	stroke	compressi
0	1	0	0	0	1	3	88.6	168.8	64.1	48.8	...	dohc	130	3.47	2.68	
1	1	0	0	0	2	3	88.6	168.8	64.1	48.8	...	dohc	130	3.47	2.68	
2	0	1	0	0	3	1	94.5	171.2	65.5	52.4	...	ohcv	152	2.68	3.47	
3	1	0	0	0	4	2	99.8	176.6	66.2	54.3	...	ohc	109	3.19	3.40	
4	0	0	0	0	5	2	99.4	176.6	66.4	54.3	...	ohc	136	3.19	3.40	

5 rows × 21 columns

In [23]:
```
enginetype=pd.get_dummies(df['enginetype'],drop_first=True)
enginetype.head()
```

Out[23]:

	l	ohc	ohcf	ohcv	rotor
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	1	0
3	0	1	0	0	0
4	0	1	0	0	0

In [24]:
```
df=pd.concat([enginetype,df],axis=1)
df.isnull().sum()
```

Out[24]:
```
l                  0
ohc                0
ohcf               0
ohcv               0
rotor              0
four               0
six                0
three              0
two                0
```

```
car_ID              0
symboling           0
wheelbase           0
carlength           0
carwidth            0
carheight           0
curbweight          0
enginetype          0
enginesize          0
boreratio           0
stroke              0
compressionratio    0
horsepower          0
peakrpm             0
citympg             0
highwaympg          0
price               0
dtype: int64
```

In [25]:
```
df=df.drop(['enginetype'],axis=1)
```

In [26]:
```
df.head()
```

Out[26]:

	l	ohc	ohcf	ohcv	rotor	four	six	three	two	car_ID	...	curbweight	enginesize	boreratio	stroke	compressionratio	horsepower	peakrp
0	0	0	0	0	0	1	0	0	0	1	...	2548	130	3.47	2.68	9.0	111	50
1	0	0	0	0	0	1	0	0	0	2	...	2548	130	3.47	2.68	9.0	111	50
2	0	0	0	1	0	0	1	0	0	3	...	2823	152	2.68	3.47	9.0	154	50
3	0	1	0	0	0	1	0	0	0	4	...	2337	109	3.19	3.40	10.0	102	55
4	0	1	0	0	0	0	0	0	0	5	...	2824	136	3.19	3.40	8.0	115	55

5 rows × 25 columns

In [27]:
```
df.describe()
```

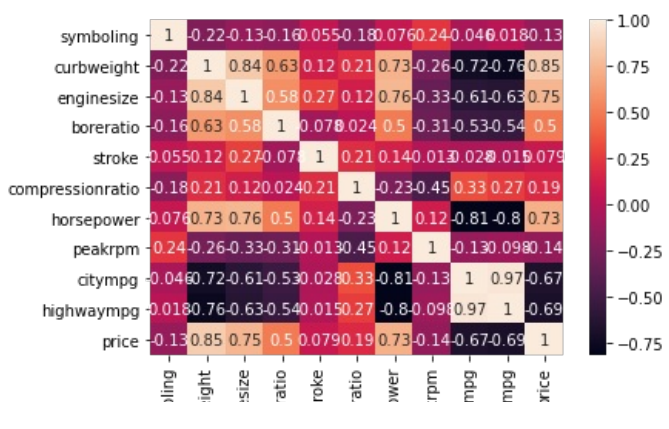
Out[27]:

	l	ohc	ohcf	ohcv	rotor	four	six	three	two	car_ID	...	curbweigh
count	190.000000	190.000000	190.000000	190.000000	190.000000	190.000000	190.000000	190.000000	190.000000	190.000000	...	190.000000
mean	0.063158	0.757895	0.063158	0.042105	0.021053	0.836842	0.084211	0.005263	0.021053	105.463158	...	2480.03157
std	0.243889	0.429489	0.243889	0.201360	0.143939	0.370486	0.278437	0.072548	0.143939	59.925052	...	445.89059
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	...	1488.00000
25%	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	54.250000	...	2128.00000
50%	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	106.500000	...	2395.00000
75%	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	157.750000	...	2823.75000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	205.000000	...	3750.00000

8 rows × 25 columns

In [28]:
```
sns.heatmap(df[["symboling","curbweight","enginesize","boreratio","stroke","compressionratio","horsepower","peakr
plt.figure(figsize=(100,100))
```

Out[28]:
```
<Figure size 7200x7200 with 0 Axes>
```



symbo curbwe engine borer st compressionr horsepo peak city highwayı

<Figure size 7200x7200 with 0 Axes>

In [29]:
```
df=df.drop(['symboling','peakrpm','highwaympg','citympg','stroke',],axis=1)
```

In [30]:
```
x=df.drop(['price'],axis=1)
y=df['price']
```

In [31]:
```
x.shape
```

Out[31]: (190, 19)

In [32]:
```
from sklearn.model_selection import train_test_split
x_tr,x_te,y_tr,y_te=train_test_split(x,y,test_size=0.15,random_state=200)
```

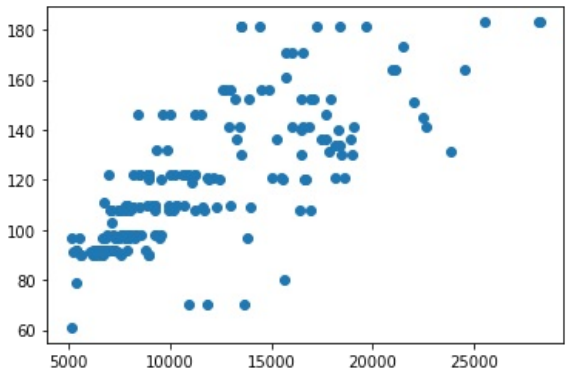
In [33]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_tr=sc.fit_transform(x_tr)
x_te=sc.transform(x_te)
```

In [34]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_tr,y_tr)
```

Out[34]: LinearRegression()

In [35]:
```
plt.scatter(df.price,df.enginesize)
```

Out[35]: <matplotlib.collections.PathCollection at 0x249837fa8b0>



In [36]:
```
y_pred=model.predict(x_te)
```

In [37]:
```
y_pred
```

Out[37]:
```
array([ 9791.77546556, 11332.03184687,  7527.94188211, 14722.92462009,
       17621.90830733, 13262.03776563, 19064.81182464,  6215.42094522,
       16000.47771019, 10494.94351621,  6923.48284459,  9883.86780493,
        6908.27158035, 11304.72153732, 15779.95592913,  6636.45566505,
       11355.1996256 , 15441.07000946, 17259.21935798,  5909.5160443 ,
        6697.25389125,  7776.36578613, 13314.10699518,  7672.45992822,
        6084.68701374, 13526.7635203 , 22695.28487201,  6887.84267262,
       10351.75369187])
```

In [38]:
```
model.score(x_te,y_te)
```

Out[38]:
```
0.856182405140828
```

In [39]:
```
print(model.intercept_)
print(model.coef_)
```

```
11617.796068322985
[ -767.1467782    335.86913948  -445.18141048  -643.49090382
  -259.53509202  -985.6587882    868.61653579   265.19689594
  -259.53509202  -824.58213014   543.60308068  -504.70448004
  1247.55206498  -190.06077879  3106.44591898 -1826.26646283
  1230.53681948   611.98160241  1106.01633542]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: