In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
import seaborn as sns
```

In [5]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [6]:
```
df=pd.read_csv("cars.csv")
```

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

Out[7]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	?	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	23	28	16845
201	-1	95	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160	19	25	19045
202	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134	18	23	21485
203	-1	95	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106	26	27	22470
204	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	19	25	22625

205 rows × 15 columns

In [8]:
```
df.head()
```

Out[8]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	?	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          205 non-null    int64  
 1   normalized-losses  205 non-null    object 
 2   make               205 non-null    object 
 3   fuel-type          205 non-null    object 
 4   body-style         205 non-null    object 
 5   drive-wheels       205 non-null    object 
 6   engine-location    205 non-null    object 
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object 
 10  engine-size        205 non-null    int64  
 11  horsepower         205 non-null    object 
 12  city-mpg           205 non-null    int64  
 13  highway-mpg        205 non-null    int64  
 14  price              205 non-null    int64  
dtypes: float64(2), int64(5), object(8)
memory usage: 24.1+ KB
```

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

Out[10]:

	symboling	width	height	engine-size	city-mpg	highway-mpg	price
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	65.907805	53.724878	126.907317	25.219512	30.751220	13227.478049
std	1.245307	2.145204	2.443522	41.642693	6.542142	6.886443	7902.651615
min	-2.000000	60.300000	47.800000	61.000000	13.000000	16.000000	5118.000000
25%	0.000000	64.100000	52.000000	97.000000	19.000000	25.000000	7788.000000
50%	1.000000	65.500000	54.100000	120.000000	24.000000	30.000000	10345.000000
75%	2.000000	66.900000	55.500000	141.000000	30.000000	34.000000	16500.000000
max	3.000000	72.300000	59.800000	326.000000	49.000000	54.000000	45400.000000

In [11]:
```
#change the data type of normalise-losses
df["normalized-losses"].replace("?",np.nan,inplace=True)
```

In [12]:
```
#change the data type of hoursepower
df["horsepower"].replace("?",np.nan,inplace=True)
```

In [13]:
```
df["normalized-losses"]=df["normalized-losses"].astype(float)
```

In [14]:
```
df.head()
```

Out[14]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	NaN	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	NaN	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	NaN	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450

In [15]:
```
#replace the null value of normalized-losses by mean
nmean=df["normalized-losses"].mean()
df["normalized-losses"].fillna(nmean,inplace=True)
```

In [16]:

```
df.isnull().sum()
```

Out[16]:

```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           2
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [17]:

```
df.dropna()
```

Out[17]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	23	28	16845
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160	19	25	19045
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134	18	23	21485
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106	26	27	22470
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	19	25	22625

203 rows × 15 columns

In [18]:

```
df.isnull().sum()
```

Out[18]:

```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           2
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [19]:

```
df = df.dropna(how='all')
```

In [20]:
```
df
```

Out[20]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	23	28	16845
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160	19	25	19045
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134	18	23	21485
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106	26	27	22470
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	19	25	22625

205 rows × 15 columns

In [21]:
```
df.isnull().sum()
```

Out[21]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           2
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [22]:
```
f = df.dropna(thresh=len(df.columns)-1)
```

In [23]:
```
f
```

Out[23]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	13495
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111	21	27	16500
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154	19	26	16500
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102	24	30	13950
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115	18	22	17450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	23	28	16845
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160	19	25	19045
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134	18	23	21485
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106	26	27	22470
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114	19	25	22625

205 rows × 15 columns

In [24]:
```
df.isnull().sum()
```

Out[24]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           2
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [25]:
```
df["horsepower"]=df["horsepower"].astype(float)
```

In [26]:
```
df
```

Out[26]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111.0	21	27	13495
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111.0	21	27	16500
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154.0	19	26	16500
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102.0	24	30	13950
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115.0	18	22	17450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114.0	23	28	16845
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160.0	19	25	19045
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134.0	18	23	21485
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106.0	26	27	22470
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114.0	19	25	22625

205 rows × 15 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          205 non-null    int64  
 1   normalized-losses  205 non-null    float64
 2   make               205 non-null    object 
 3   fuel-type          205 non-null    object 
 4   body-style         205 non-null    object 
 5   drive-wheels       205 non-null    object 
 6   engine-location    205 non-null    object 
 7   width              205 non-null    float64
 8   height             205 non-null    float64
 9   engine-type        205 non-null    object 
 10  engine-size        205 non-null    int64  
 11  horsepower         203 non-null    float64
 12  city-mpg           205 non-null    int64  
 13  highway-mpg        205 non-null    int64  
 14  price              205 non-null    int64  
dtypes: float64(4), int64(5), object(6)
memory usage: 24.1+ KB
```

In [28]:
```
df.isnull().sum()
```

Out[28]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           2
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

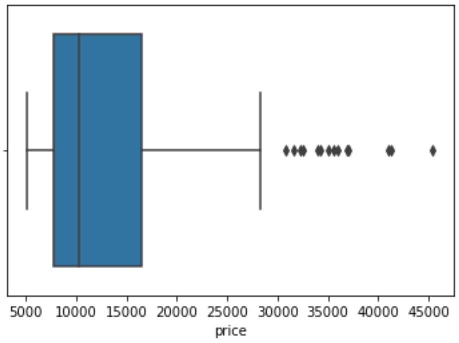
In [29]:
```
#replace the null value of normalized-losses by mean
nmean=df["horsepower"].mean()
df["horsepower"].fillna(nmean,inplace=True)
```

In [30]:
```
df.isnull().sum()
```

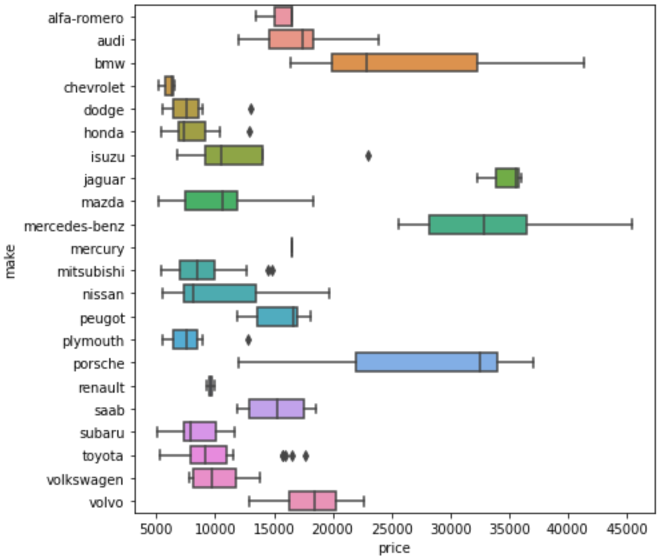
Out[30]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
body-style           0
drive-wheels         0
engine-location      0
width                0
height               0
engine-type          0
engine-size          0
horsepower           0
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [31]:
```
#check the outlier and handle them
sns.boxplot(data=df,x="price")
```

Out[31]:
```
<AxesSubplot:xlabel='price'>
```



In [32]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



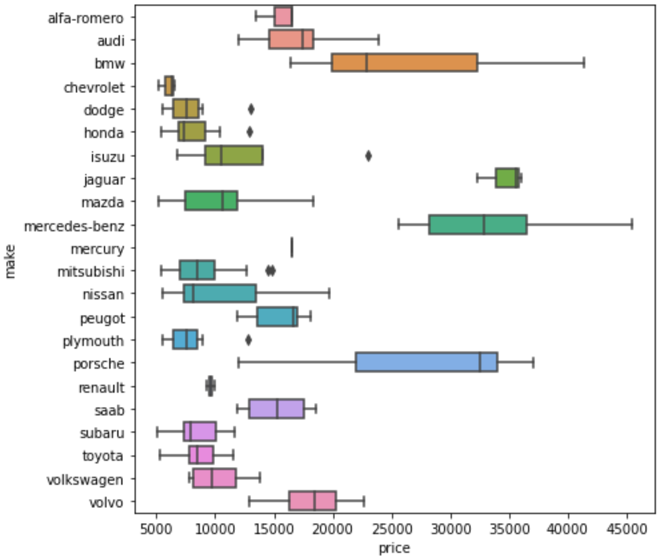
In [33]:
```
#removing ouliers of toyota
df[(df["make"]=="toyota")&(df["price"]>13000)]
```

Out[33]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
172	2	134.0	toyota	gas	convertible	rwd	front	65.6	53.0	ohc	146	116.0	24	30	17669
178	3	197.0	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	171	161.0	20	24	16558
179	3	197.0	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	171	161.0	19	24	15998
180	-1	90.0	toyota	gas	sedan	rwd	front	66.5	54.1	dohc	171	156.0	20	24	15690
181	-1	122.0	toyota	gas	wagon	rwd	front	66.5	54.1	dohc	161	156.0	19	24	15750

In [34]:
```
#removing ouliers of toyota
df.drop(index=[172,178,179,180,181],inplace=True)
```

In [35]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



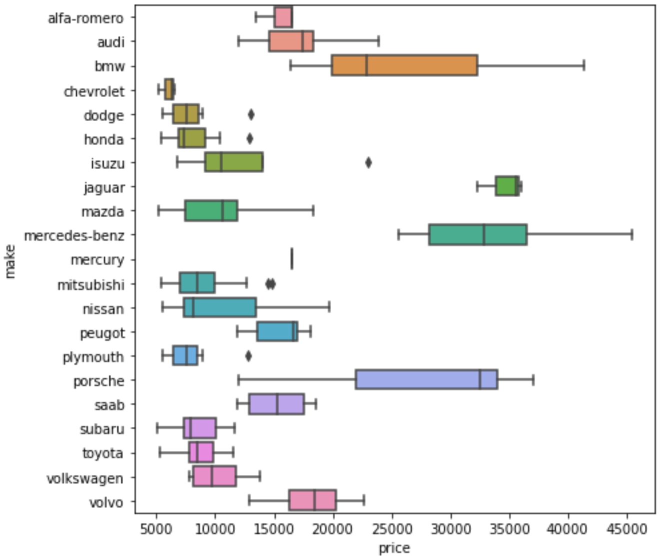
In [36]:
```
#removing ouliers   of
df[(df["make"]=="renault")&(df["price"]>1000)]
```

Out[36]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
130	0	122.0	renault	gas	wagon	fwd	front	66.5	55.2	ohc	132	104.256158	23	31	9295
131	2	122.0	renault	gas	hatchback	fwd	front	66.6	50.5	ohc	132	104.256158	23	31	9895

In [37]:
```
#removing ouliers of toyota
df.drop(index=[130,131],inplace=True)
```

In [38]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



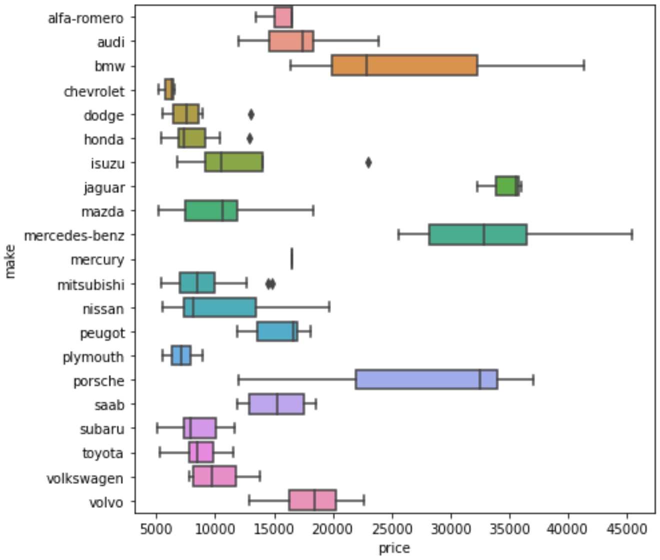
In [39]:
```
#removing ouliers plymouth   of
df[(df["make"]=="plymouth")&(df["price"]>12000)]
```

Out[39]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
124	3	122.0	plymouth	gas	hatchback	rwd	front	66.3	50.2	ohc	156	145.0	19	24	12764

In [40]:
```
#removing ouliers of toyota
df.drop(index=[124],inplace=True)
```

In [41]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



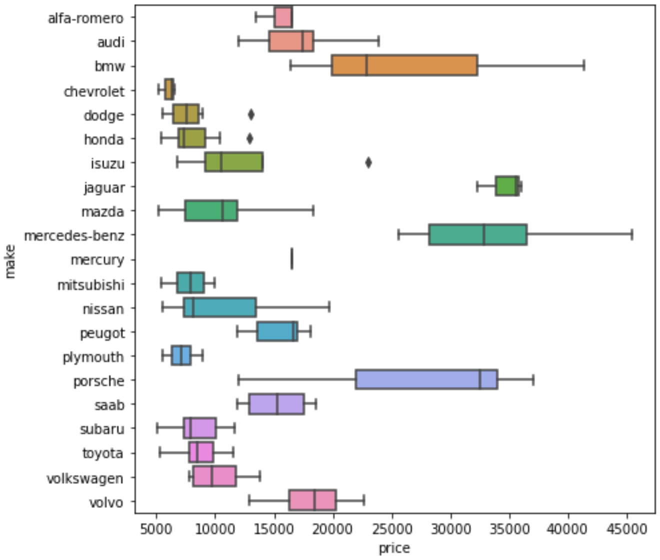
In [42]:
```
#removing ouliers mitsubshi   of
df[(df["make"]=="mitsubishi")&(df["price"]>12000)]
```

Out[42]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
82	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	156	145.0	19	24	12629
83	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	156	145.0	19	24	14869
84	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	156	145.0	19	24	14489

In [43]:
```
#removing ouliers of mitsibishi
df.drop(index=[82,83,84],inplace=True)
```

In [44]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



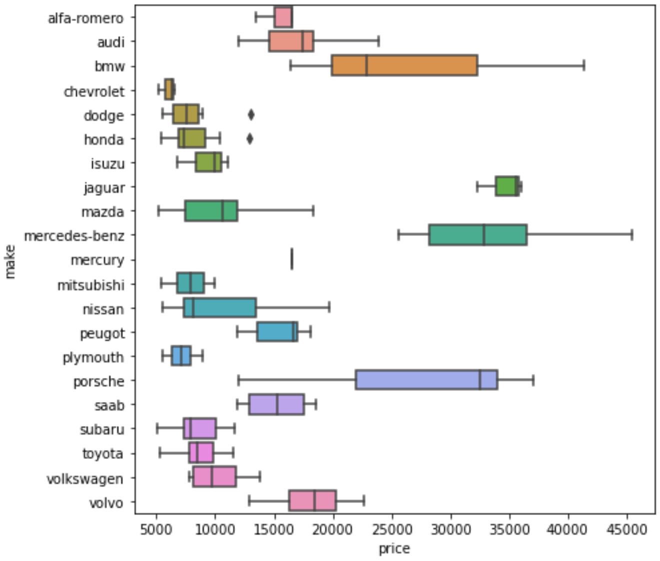
In [45]:
```
#removing ouliers isuzu  of
df[(df["make"]=="isuzu")&(df["price"]>13000)]
```

Out[45]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
45	0	122.0	isuzu	gas	sedan	fwd	front	63.6	52.0	ohc	90	70.0	38	43	23000

In [46]:
```
#removing ouliers of mitsubishi
df.drop(index=[45],inplace=True)
```

In [47]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



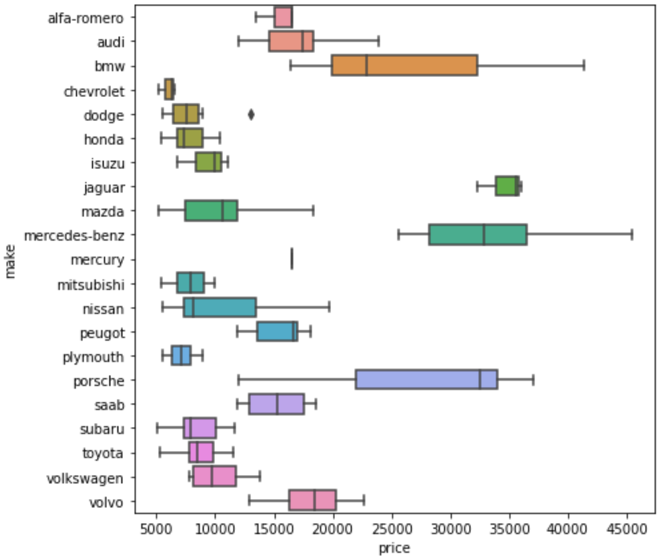
In [48]:
```
#removing ouliers honda  of
df[(df["make"]=="honda")&(df["price"]>12000)]
```

Out[48]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
41	0	85.0	honda	gas	sedan	fwd	front	65.2	54.1	ohc	110	101.0	24	28	12945

In [49]:
```
#removing ouliers of honda
df.drop(index=[41],inplace=True)
```

In [50]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



In [51]:
```
#removing ouliers dodge of
df[(df["make"]=="dodge")&(df["price"]>11000)]
```

Out[51]:

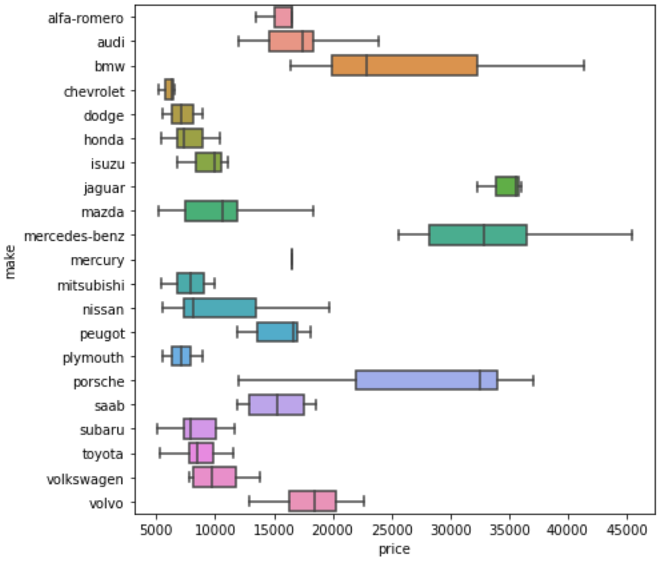
	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg	price
29	3	145.0	dodge	gas	hatchback	fwd	front	66.3	50.2	ohc	156	145.0	19	24	12964

In [52]:

```
#removing ouliers of dodge
df.drop(index=[29],inplace=True)
```

In [53]:

```
plt.figure(figsize=(7,7))
sns.boxplot(data=df,x='price',y="make")
plt.show()
```



In [54]:

```
df_num=df.select_dtypes([int,float])
```

In [55]:

```
df_cat=df.select_dtypes([object])
```

In [56]:

```
#one-hot encoding
pd.get_dummies(df_cat["make"])
```

Out[56]:

	alfa-romero	audi	bmw	chevrolet	dodge	honda	isuzu	jaguar	mazda	mercedes-benz	...	mitsubishi	nissan	peugot	plymouth	porsche	saab	subaru	toyota
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
201	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
202	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
203	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
204	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0

191 rows × 21 columns

In [57]:

```
#perform one- hot encoding  on drive - wheels cols
```

In [58]:

```
#label-encoding
from sklearn.preprocessing import LabelEncoder
```

In [59]:

```
#create instance of label encoder
le=LabelEncoder()
```

In [60]:
```
#apply label encodi
le.fit_transform(df_cat["fuel-type"])
```

Out[60]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1])
```

In [61]:
```
le.classes_
```

Out[61]:
```
array(['diesel', 'gas'], dtype=object)
```

In [62]:
```
#applying label encoding  on all cols by using loop

#import library
from sklearn.preprocessing import LabelEncoder

for i in df_cat:
    #create instance label  encoder
    le=LabelEncoder()
    #apply the label encoding
    df_cat[i]=le.fit_transform(df_cat[i])
```

In [63]:
```
df_cat.head()
```

Out[63]:

	make	fuel-type	body-style	drive-wheels	engine-location	engine-type
0	0	1	0	2	0	0
1	0	1	0	2	0	0
2	0	1	2	2	0	5
3	1	1	3	1	0	3
4	1	1	3	0	0	3

In [64]:
```
df_new=pd.concat([df_num,df_cat],axis=1)
```

In [65]:
```
df_new.head()
```

Out[65]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price	make	fuel-type	body-style	drive-wheels	engine-location	engine-type
0	3	122.0	64.1	48.8	130	111.0	21	27	13495	0	1	0	2	0	0
1	3	122.0	64.1	48.8	130	111.0	21	27	16500	0	1	0	2	0	0
2	1	122.0	65.5	52.4	152	154.0	19	26	16500	0	1	2	2	0	5
3	2	164.0	66.2	54.3	109	102.0	24	30	13950	1	1	3	1	0	3
4	2	164.0	66.4	54.3	136	115.0	18	22	17450	1	1	3	0	0	3

In [66]:
```
#feature scaling

#standarization  (z score) ----->  StandardScaler
#import module
from sklearn.preprocessing import StandardScaler

#create an instance of a class
sd=StandardScaler()

#apply standardscaler
sd.fit_transform(df_new[["width"]])
```

```
       [-0.94714194],
       [-0.94714194],
       [-0.94714194],
       [-0.94714194],
       [-0.94714194],
       [-0.94714194],
       [-0.94714194],
       [-0.5834242 ],
       [-0.90167722],
       [-0.90167722],
       [-0.85621251],
       [-0.85621251],
       [-0.85621251],
       [-0.85621251],
       [-0.90167722],
       [-0.3106359 ],
       [-0.3106359 ],
       [-0.3106359 ],
       [-1.53818327],
       [ 0.05308184],
```

In [67]:
```
#Normalization
#import module
from sklearn.preprocessing import MinMaxScaler

#create an instance of a class
mn=MinMaxScaler()

#apply standardscaler
mn.fit_transform(df_new[["width"]])
```

```
       [0.3       ],
       [0.3       ],
       [0.30833333],
       [0.30833333],
       [0.30833333],
       [0.30833333],
       [0.3       ],
       [0.40833333],
       [0.40833333],
       [0.40833333],
       [0.18333333],
       [0.475     ],
       [0.125     ],
       [0.275     ],
       [0.40833333],
       [0.775     ],
       [0.775     ],
       [0.85833333],
       [0.325     ],
       [0.325     ],
```

In [92]:
```
x=df.iloc[2:-1]
```

In [93]:
```
x
```

Out[93]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154.0	19	26
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102.0	24	30
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115.0	18	22
5	2	122.0	audi	gas	sedan	fwd	front	66.3	53.1	ohc	136	110.0	19	25
6	1	158.0	audi	gas	sedan	fwd	front	71.4	55.7	ohc	136	110.0	19	25
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
199	-1	74.0	volvo	gas	wagon	rwd	front	67.2	57.5	ohc	130	162.0	17	22
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	141	114.0	23	28
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	141	160.0	19	25
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	173	134.0	18	23
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	145	106.0	26	27

188 rows × 14 columns

In [70]:
```
y=df["price"]
```

In [71]:
```
y
```

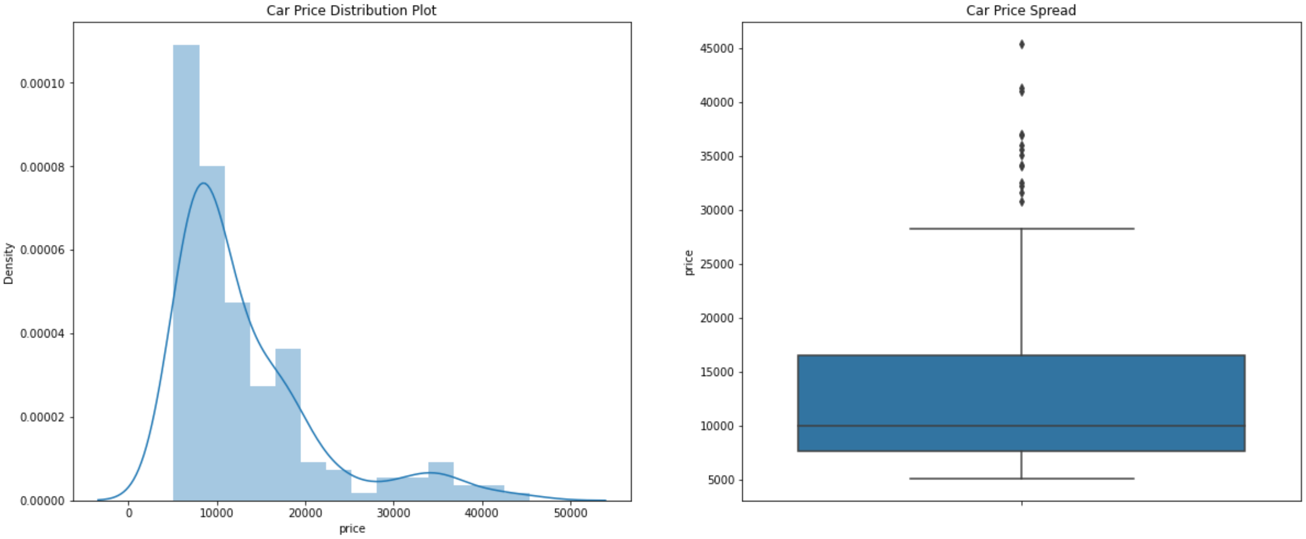
Out[71]:
```
0      13495
1      16500
2      16500
3      13950
4      17450
       ...  
200    16845
201    19045
202    21485
203    22470
204    22625
Name: price, Length: 191, dtype: int64
```

In [72]:
```
from sklearn.model_selection import train_test_split
```

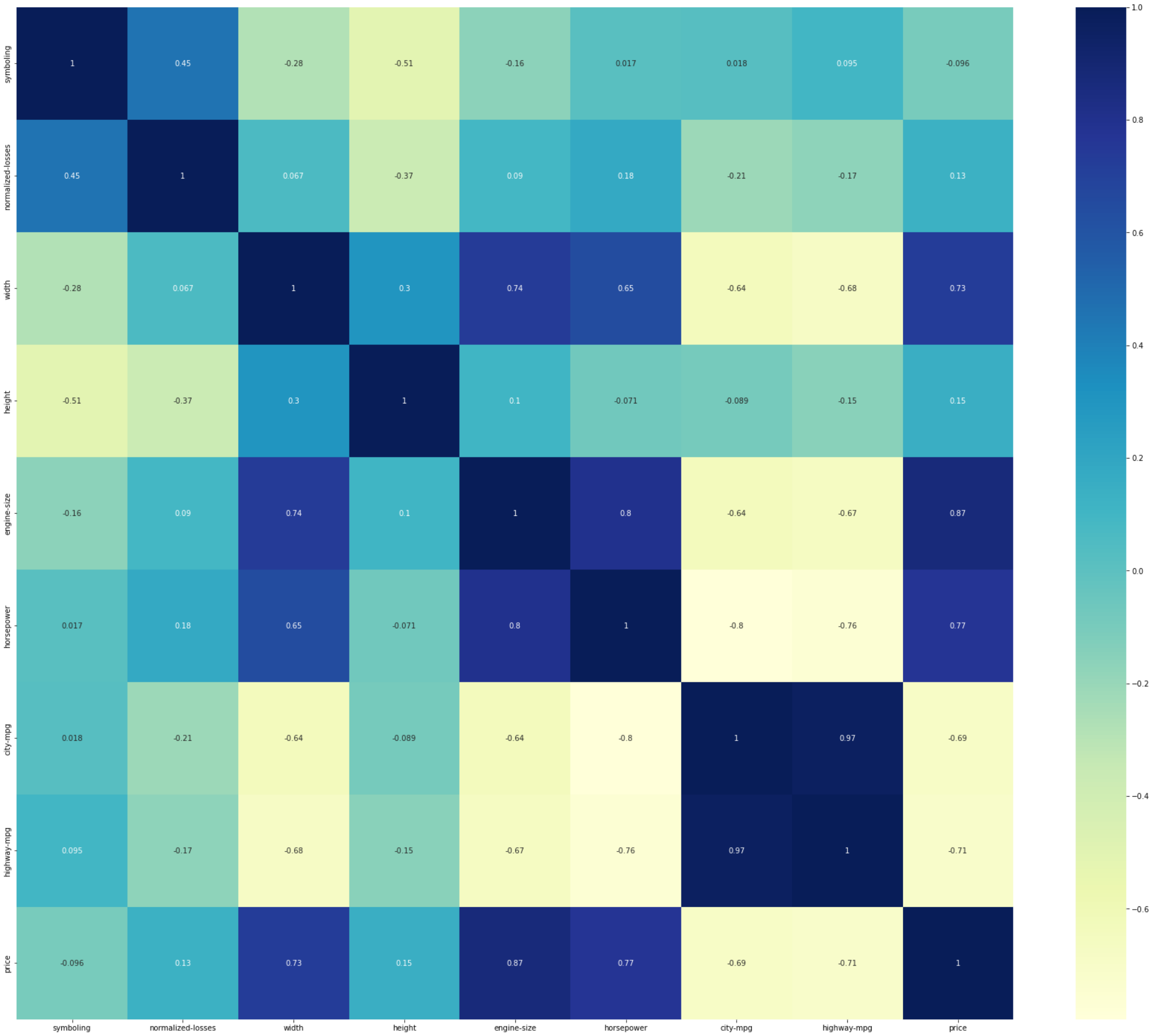
In [73]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)
print("x train: ",x_train.shape)
print("x test: ",x_test.shape)
print("y train: ",y_train.shape)
print("y test: ",y_test.shape)
```

```
x train:  (152, 14)
x test:  (39, 14)
y train:  (152,)
y test:  (39,)
```

```
In [76]: plt.figure(figsize=(20,8))

         plt.subplot(1,2,1)
         plt.title('Car Price Distribution Plot')
         sns.distplot(df.price)

         plt.subplot(1,2,2)
         plt.title('Car Price Spread')
         sns.boxplot(y=df.price)

         plt.show()
```



In [78]:
```
plt.figure(figsize = (30, 25))
sns.heatmap(df.corr(), annot = True, cmap="YlGnBu")
plt.show()
```



In [83]:
```
from sklearn.model_selection import train_test_split
```

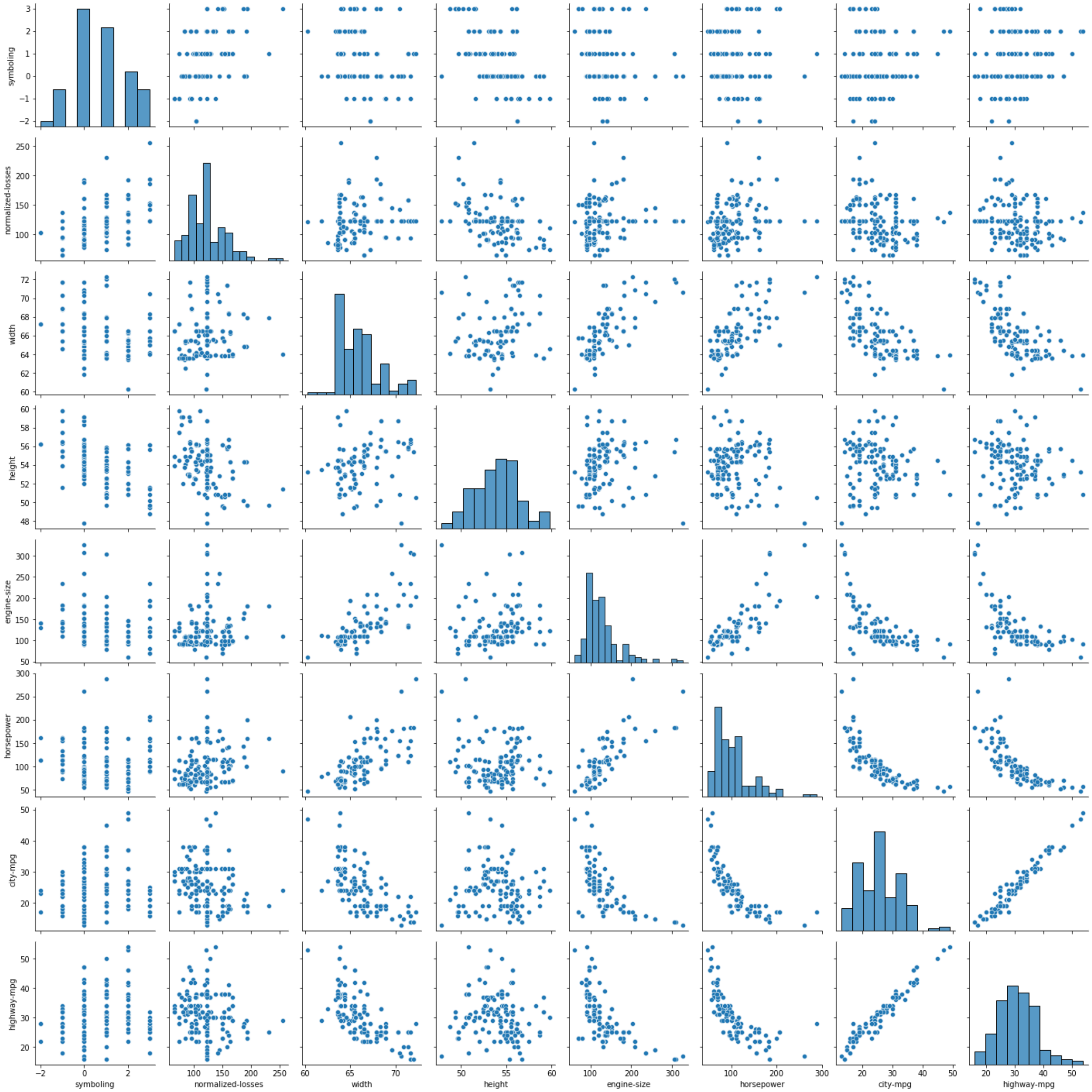
In [84]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [85]:
```
#creating the MLmodel
from sklearn.linear_model import LinearRegression
```

In [86]:
```
#creating instance of model /inheritance of model
lr=LinearRegression()
```

In [ ]:
```
#train the model
lr.fit(x_train,y_train)
```

In [97]:
```
sns.pairplot(df)
plt.show()
```



In [100]:
```
df.head()
```

Out[100]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	horsepower	city-mpg	highway-mpg
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111.0	21	27
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130	111.0	21	27
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152	154.0	19	26
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109	102.0	24	30
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136	115.0	18	22

In [ ]: