In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

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

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

Out[3]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engin si
0	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	13
1	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	13
2	1	?	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	15
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc	10
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc	13

In [4]:
```
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
```

- there is null values in normalised losses and horsepower columns.

- need to convert the datatype into int.

In [5]:
```
df["normalized-losses"].value_counts()
```

Out[5]:
```
?      41
161    11
91      8
150     7
134     6
128     6
104     6
85      5
94      5
65      5
102     5
74      5
168     5
103     5
95      5
106     4
93      4
118     4
148     4
122     4
83      3
125     3
154     3
115     3
137     3
101     3
119     2
87      2
89      2
192     2
197     2
158     2
81      2
188     2
194     2
153     2
129     2
108     2
110     2
164     2
145     2
113     2
256     1
107     1
90      1
231     1
142     1
121     1
78      1
98      1
186     1
77      1
Name: normalized-losses, dtype: int64
```

In [6]:

```
df["horsepower"].value_counts()
```

Out[6]:

68	19
70	11
69	10
116	9
110	8
95	7
88	6
62	6
101	6
160	6
114	6
84	5
97	5
102	5
145	5
82	5
76	5
111	4
92	4
123	4
86	4
90	3
73	3
85	3
207	3
182	3
121	3
152	3
112	2
56	2
161	2
156	2
94	2
52	2
?	2
162	2
155	2
184	2
100	2
176	2
55	1
262	1
134	1
115	1
140	1
48	1
58	1
60	1
78	1
135	1
200	1
64	1
120	1
72	1
154	1
288	1
143	1
142	1

```
175     1
106     1
Name: horsepower, dtype: int64
```

In [7]:
```
#handling null values
#normalized losses
```

In [8]:
```
#replacing the null values by nan
df["normalized-losses"].replace("?" ,np.nan, inplace = True)
```

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

Out[9]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engin si
0	3	NaN	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	13
1	3	NaN	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	13
2	1	NaN	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	15
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc	10
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc	13

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

Out[10]:
```
symboling            0
normalized-losses    41
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 [11]:
```
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  164 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 [12]:
```
df["normalized-losses"] = df["normalized-losses"].astype(float)
```

In [13]:
```
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  164 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         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(3), int64(5), object(7)
memory usage: 24.1+ KB
```

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

Out[14]:

	symboling	normalized-losses	width	height	engine-size	city-mpg	highway-mpg	pri
count	205.000000	164.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.0000(
mean	0.834146	122.000000	65.907805	53.724878	126.907317	25.219512	30.751220	13227.4780
std	1.245307	35.442168	2.145204	2.443522	41.642693	6.542142	6.886443	7902.6516
min	-2.000000	65.000000	60.300000	47.800000	61.000000	13.000000	16.000000	5118.0000(

	symboling	normalized-losses	width	height	engine-size	city-mpg	highway-mpg	pri
25%	0.000000	94.000000	64.100000	52.000000	97.000000	19.000000	25.000000	7788.0000
50%	1.000000	115.000000	65.500000	54.100000	120.000000	24.000000	30.000000	10345.0000
75%	2.000000	150.000000	66.900000	55.500000	141.000000	30.000000	34.000000	16500.0000
max	3.000000	256.000000	72.300000	59.800000	326.000000	49.000000	54.000000	45400.0000

In [15]:
```
p = df["normalized-losses"].mean()
```

In [16]:
```
p
```

Out[16]:
```
122.0
```

In [17]:
```
df["normalized-losses"].fillna(p, inplace = True)
```

In [18]:
```
df
```

Out[18]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	
...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	

205 rows × 15 columns

In [19]:
```
#horsepower
```

In [20]:
```
df["horsepower"].replace("?",np.nan, inplace = True)
```

In [21]:
```
df["horsepower"]
```

Out[21]:
```
0      111
1      111
2      154
3      102
4      115
      ...
200    114
201    160
202    134
203    106
204    114
Name: horsepower, Length: 205, dtype: object
```

In [22]:
```
df["horsepower"] = df["horsepower"].astype(float)
```

In [23]:
```
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 [24]:
```
df.describe()
```

Out[24]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg
count	205.000000	205.000000	205.000000	205.000000	205.000000	203.000000	205.000000	205.000000
mean	0.834146	122.000000	65.907805	53.724878	126.907317	104.256158	25.219512	30.751220
std	1.245307	31.681008	2.145204	2.443522	41.642693	39.714369	6.542142	6.886443
min	-2.000000	65.000000	60.300000	47.800000	61.000000	48.000000	13.000000	16.000000

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg
25%	0.000000	101.000000	64.100000	52.000000	97.000000	70.000000	19.000000	25.000000
50%	1.000000	122.000000	65.500000	54.100000	120.000000	95.000000	24.000000	30.000000
75%	2.000000	137.000000	66.900000	55.500000	141.000000	116.000000	30.000000	34.000000
max	3.000000	256.000000	72.300000	59.800000	326.000000	288.000000	49.000000	54.000000

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

Out[25]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	
...	...	...	...	...	...	...	...	...	...	...	...
200	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	
201	-1	95.0	volvo	gas	sedan	rwd	front	68.8	55.5	ohc	
202	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv	
203	-1	95.0	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc	
204	-1	95.0	volvo	gas	sedan	rwd	front	68.9	55.5	ohc	

203 rows × 15 columns

In [26]:
```
df.isnull().sum()
```

Out[26]:
```
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 [27]:
```
c = df["horsepower"].mean()
```

In [28]:
```
c
```

Out[28]:
```
104.25615763546799
```

In [29]:
```
df.dropna(inplace = True)
```

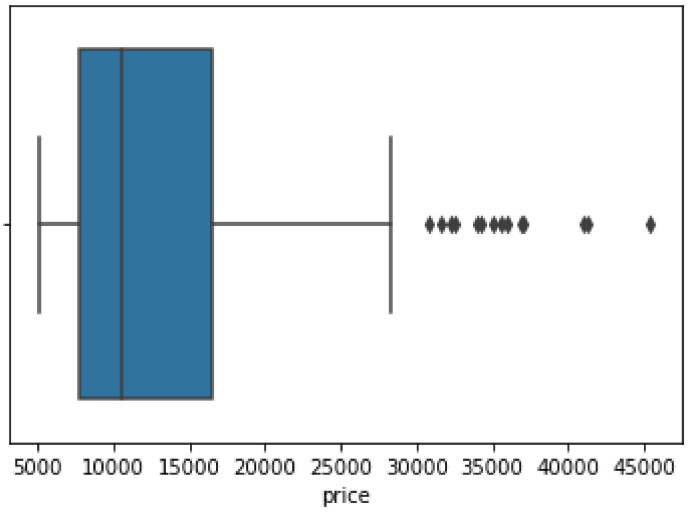
In [30]:
```
df.isnull().sum()
```

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]:
```
#handling outliers
```

In [32]:
```
sns.boxplot(data=df, x=df["price"])
```

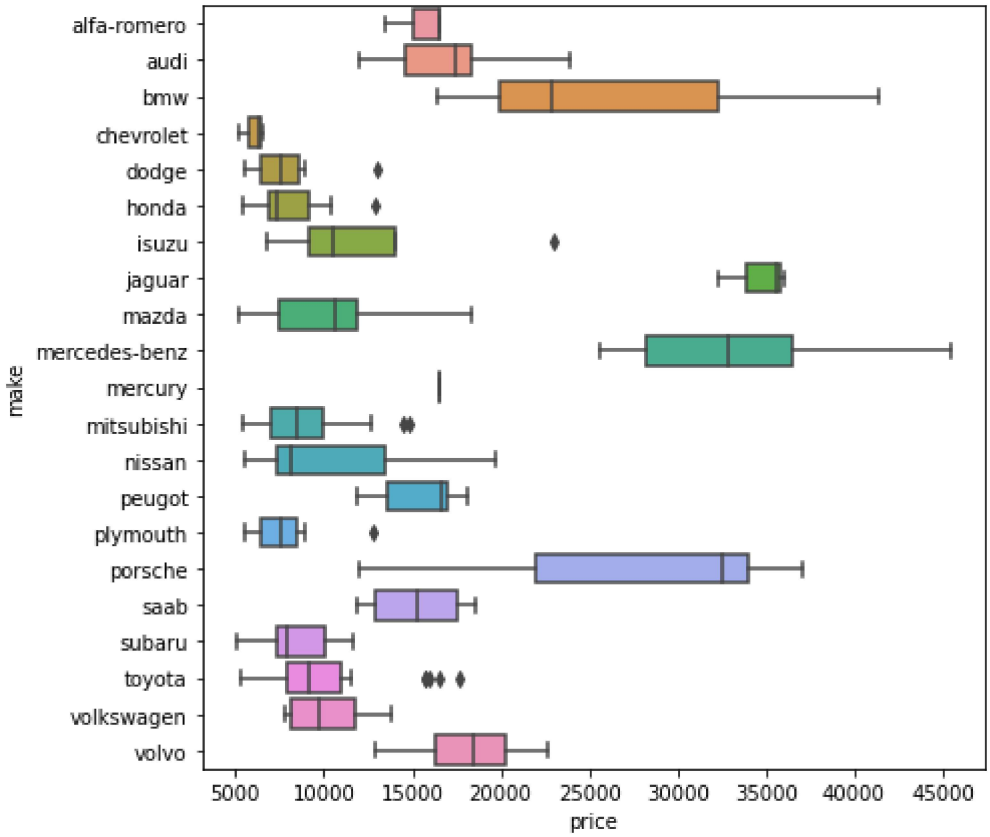
Out[32]:
```
<AxesSubplot:xlabel='price'>
```



In [33]:
```
df["make"].value_counts()
```

Out[33]:
```
toyota           32
nissan           18
mazda            17
honda            13
mitsubishi       13
subaru           12
volkswagen       12
peugot           11
volvo            11
dodge             9
mercedes-benz     8
bmw               8
audi              7
plymouth          7
saab              6
porsche           5
isuzu             4
alfa-romero       3
jaguar            3
chevrolet         3
mercury           1
Name: make, dtype: int64
```

In [34]:
```
plt.figure(figsize=(7,7))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



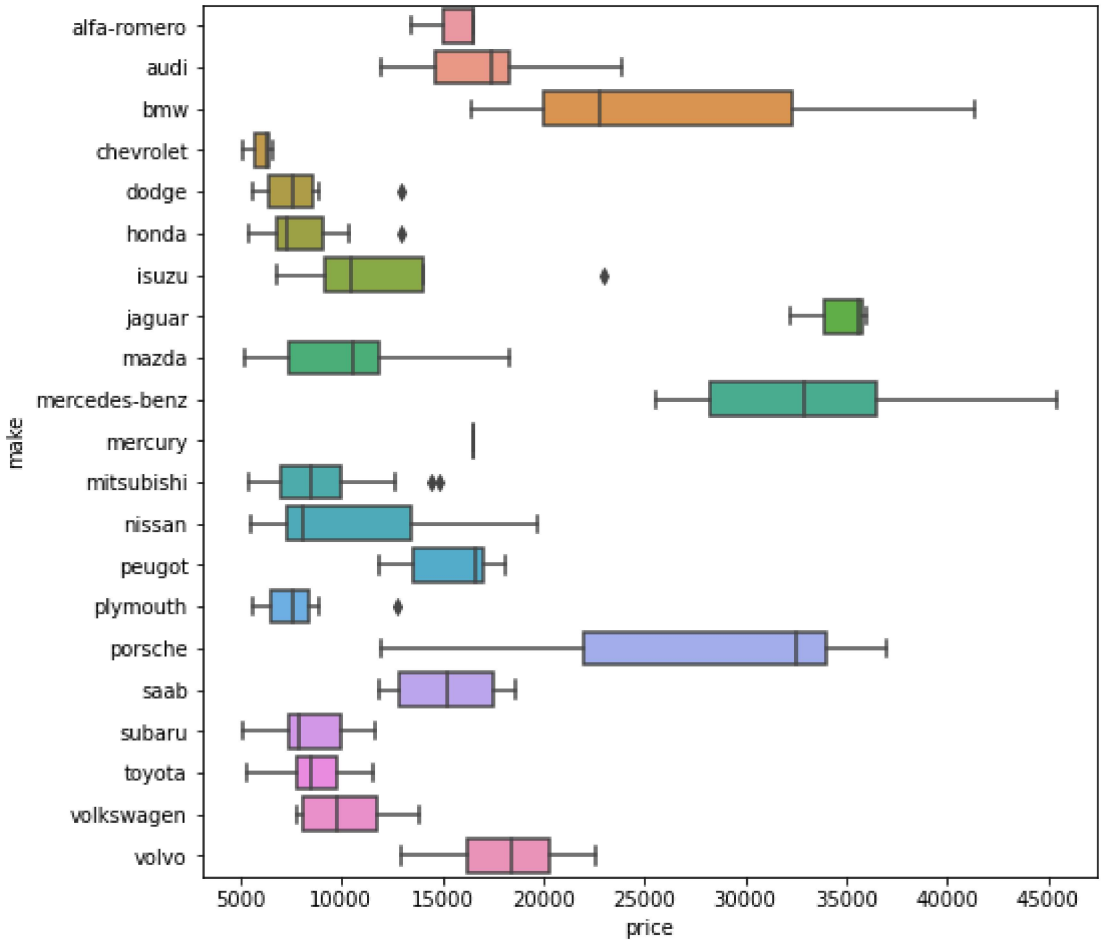
In [35]:
```
df.loc[(df["make"]== "toyota")&(df["price"]>13000)]
```

Out[35]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engi
172	2	134.0	toyota	gas	convertible	rwd	front	65.6	53.0	ohc	
178	3	197.0	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
179	3	197.0	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
180	-1	90.0	toyota	gas	sedan	rwd	front	66.5	54.1	dohc	
181	-1	122.0	toyota	gas	wagon	rwd	front	66.5	54.1	dohc	

In [36]:
```
df.drop(index=[172,178,179,180,181],inplace = True)
```

In [37]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



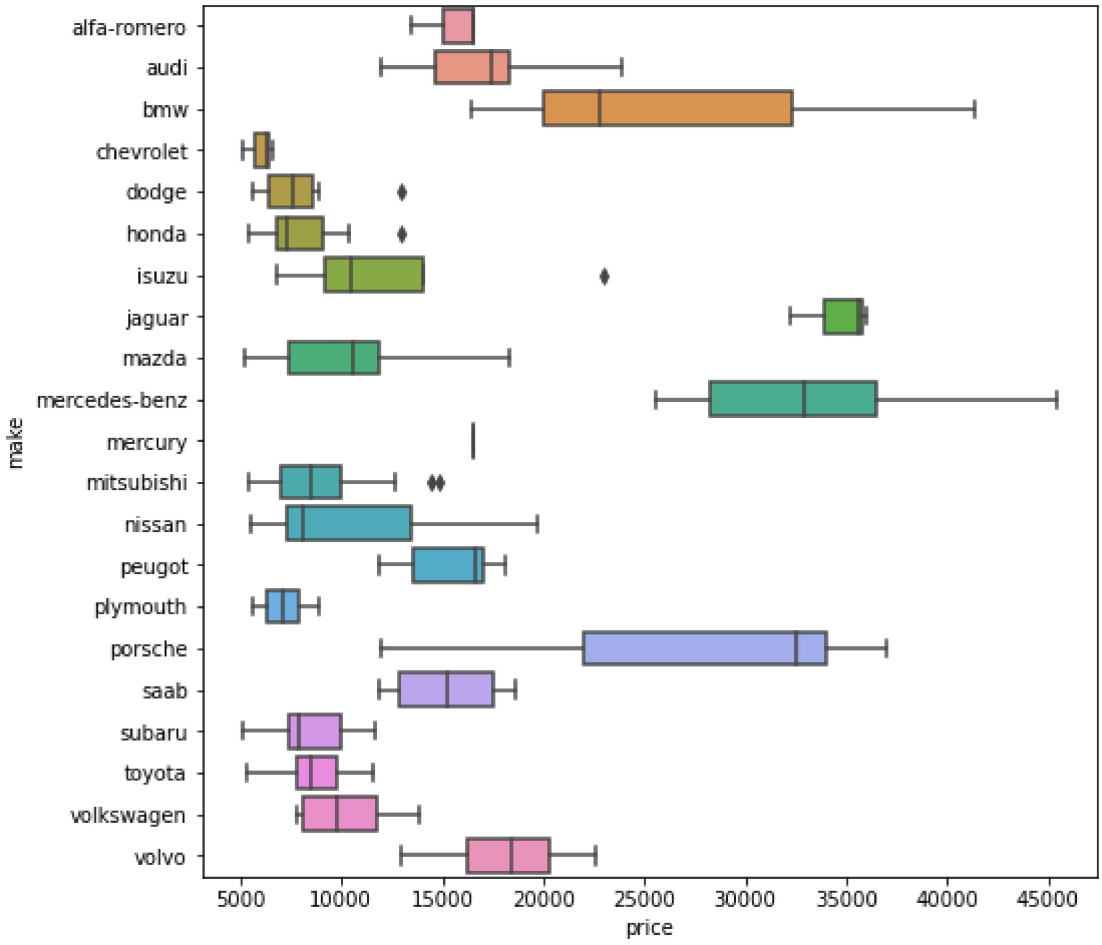
In [38]:
```
df.loc[(df["make"]== "plymouth")&(df["price"]>10000)]
```

Out[38]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
124	3	122.0	plymouth	gas	hatchback	rwd	front	66.3	50.2	ohc	

In [39]:
```
df.drop(index=[124],inplace = True)
```

In [40]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df, x=df["price"],y=df["make"])
plt.show()
```



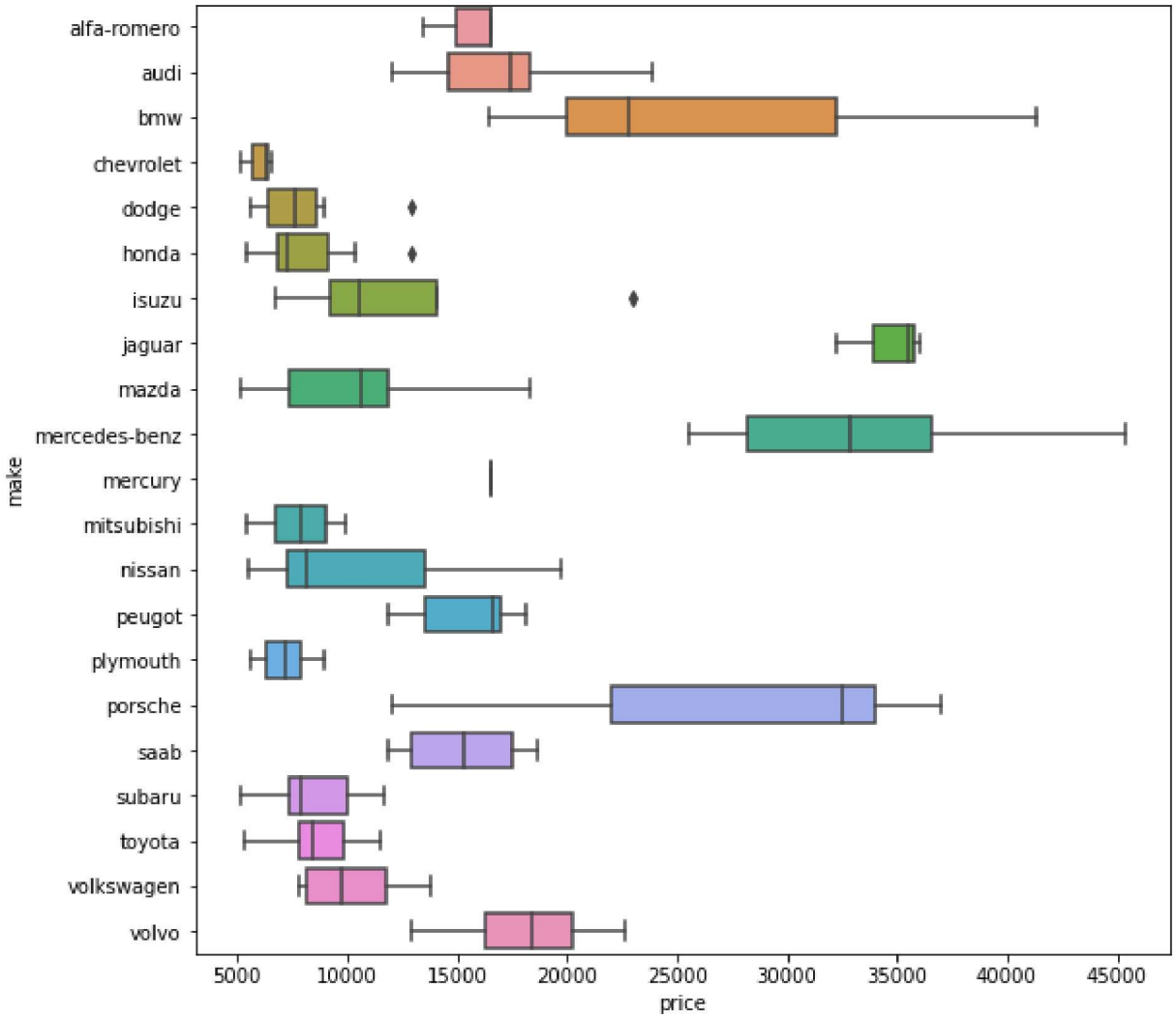
In [41]:
```
df.loc[(df["make"]=="mitsubishi")&(df["price"]>12000)]
```

Out[41]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	eng
82	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	
83	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	
84	3	122.0	mitsubishi	gas	hatchback	fwd	front	66.3	50.2	ohc	

In [42]:
```
df.drop(index=[82,83,84],inplace=True)
```

In [43]:
```
plt.figure(figsize=(9,9))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



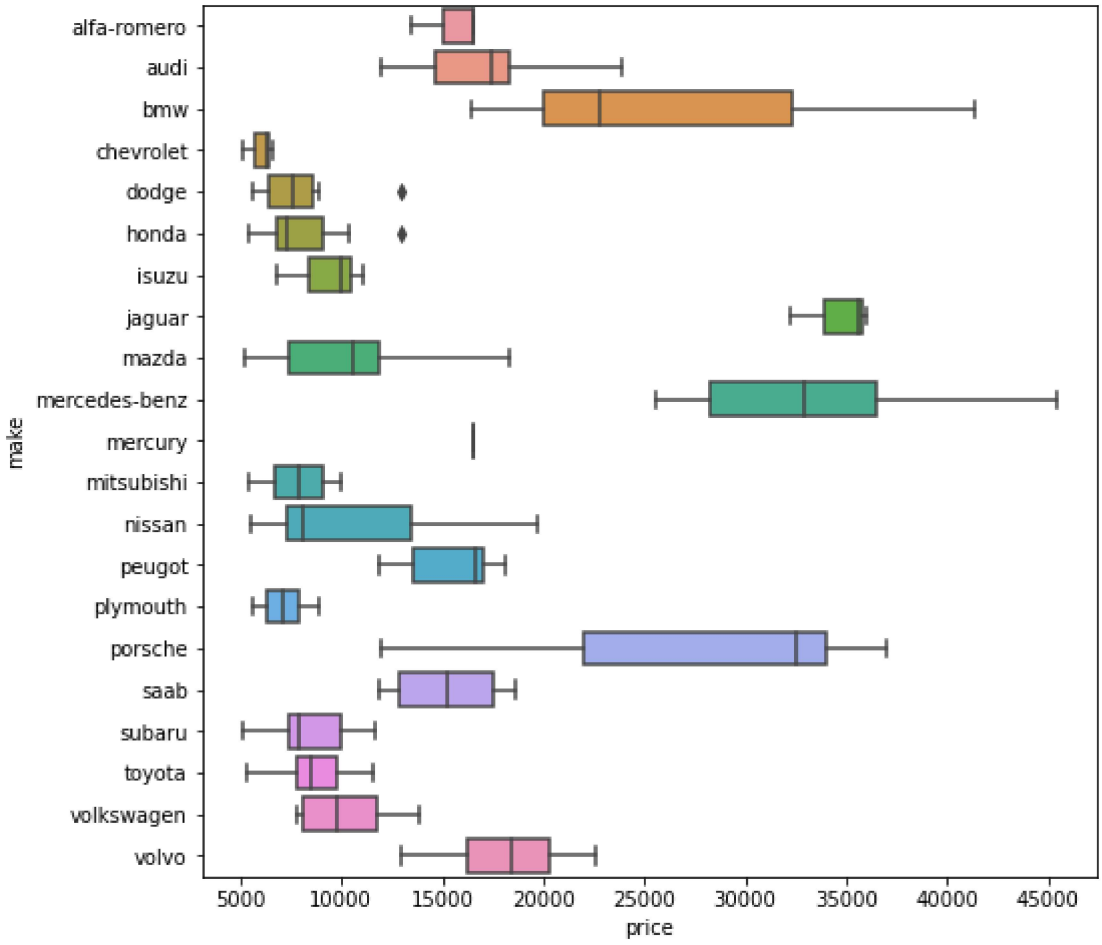
In [44]:
```
df.loc[(df["make"]=="isuzu")&(df["price"]>15000)]
```

Out[44]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	h
45	0	122.0	isuzu	gas	sedan	fwd	front	63.6	52.0	ohc	90	

In [45]:
```
df.drop(index=[45],inplace=True)
```

In [46]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



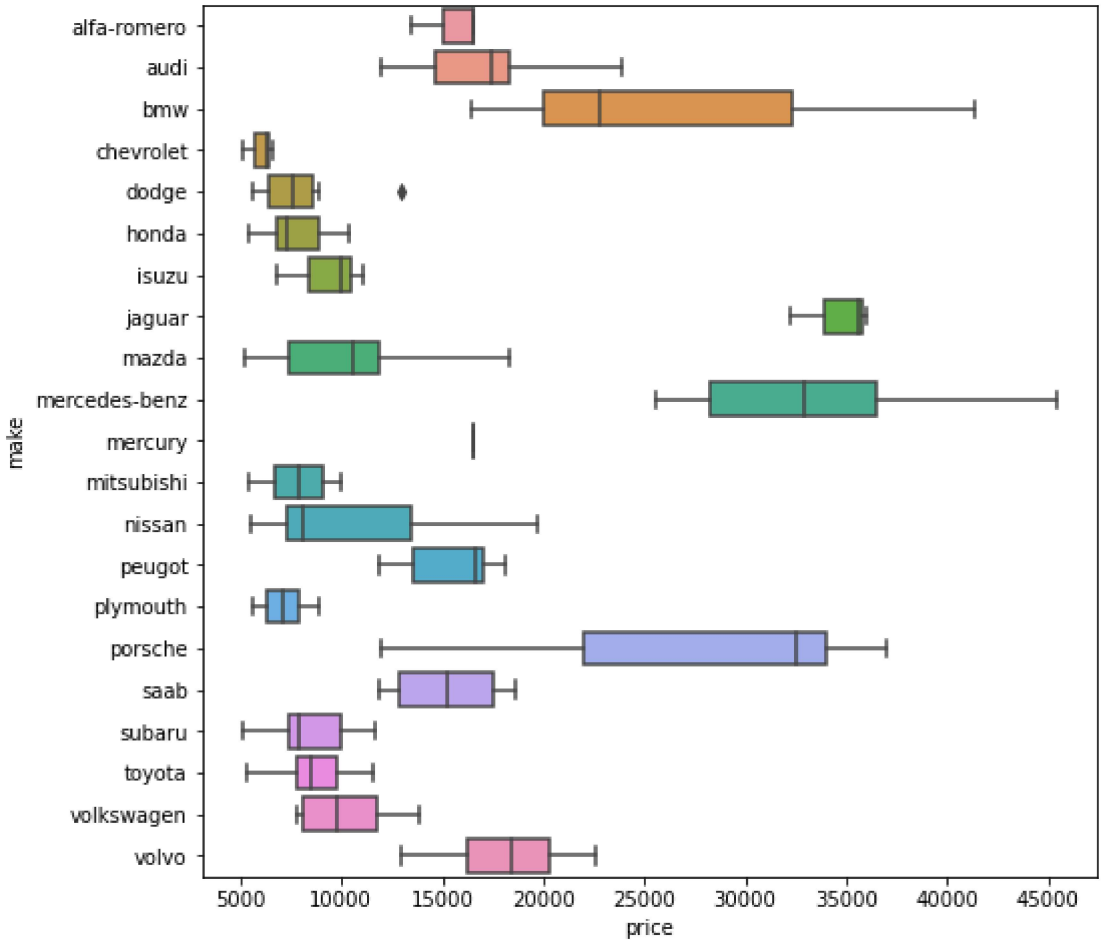
In [47]:
```
df.loc[(df["make"]=="honda")&(df["price"]>12000)]
```

Out[47]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size	I
41	0	85.0	honda	gas	sedan	fwd	front	65.2	54.1	ohc	110	

In [48]:
```
df.drop(index=[41],inplace = True)
```

In [49]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



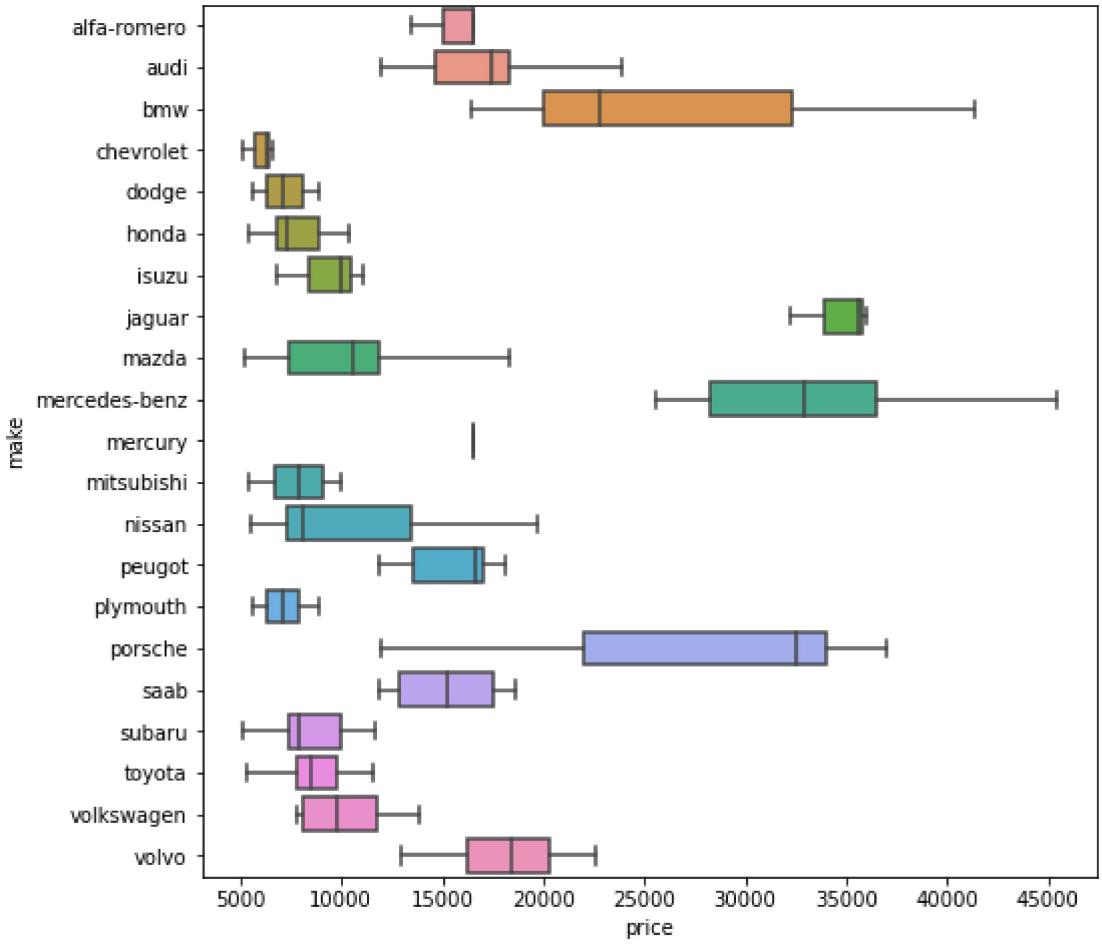
In [50]:
```
df.loc[(df["make"]== "dodge")&(df["price"]>12000)]
```

Out[50]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine siz
29	3	145.0	dodge	gas	hatchback	fwd	front	66.3	50.2	ohc	15

In [51]:
```
df.drop(index=[29],inplace=True)
```

In [52]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df, x=df["price"], y=df["make"])
plt.show()
```



In [53]:
```
#handling categorical values
```

In [54]:
```
df_cat=df.select_dtypes("object")
df_cat
```

Out[54]:

	make	fuel-type	body-style	drive-wheels	engine-location	engine-type
0	alfa-romero	gas	convertible	rwd	front	dohc
1	alfa-romero	gas	convertible	rwd	front	dohc
2	alfa-romero	gas	hatchback	rwd	front	ohcv
3	audi	gas	sedan	fwd	front	ohc
4	audi	gas	sedan	4wd	front	ohc
...	...	...	...	...	...	...
200	volvo	gas	sedan	rwd	front	ohc
201	volvo	gas	sedan	rwd	front	ohc
202	volvo	gas	sedan	rwd	front	ohcv
203	volvo	diesel	sedan	rwd	front	ohc
204	volvo	gas	sedan	rwd	front	ohc

191 rows × 6 columns

In [55]:
```
df_num=df.select_dtypes(["int","float"])
```

In [56]:
```
df_num
```

Out[56]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	64.1	48.8	130	111.0	21	27	13495
1	3	122.0	64.1	48.8	130	111.0	21	27	16500
2	1	122.0	65.5	52.4	152	154.0	19	26	16500
3	2	164.0	66.2	54.3	109	102.0	24	30	13950
4	2	164.0	66.4	54.3	136	115.0	18	22	17450
...	...	...	...	...	...	...	...	...	...
200	-1	95.0	68.9	55.5	141	114.0	23	28	16845
201	-1	95.0	68.8	55.5	141	160.0	19	25	19045
202	-1	95.0	68.9	55.5	173	134.0	18	23	21485
203	-1	95.0	68.9	55.5	145	106.0	26	27	22470
204	-1	95.0	68.9	55.5	141	114.0	19	25	22625

191 rows × 9 columns

In [57]:
```
df_cat=df.select_dtypes(object)
```

In [58]:
```
df_cat
```

Out[58]:

	make	fuel-type	body-style	drive-wheels	engine-location	engine-type
0	alfa-romero	gas	convertible	rwd	front	dohc
1	alfa-romero	gas	convertible	rwd	front	dohc
2	alfa-romero	gas	hatchback	rwd	front	ohcv
3	audi	gas	sedan	fwd	front	ohc
4	audi	gas	sedan	4wd	front	ohc
...	...	...	...	...	...	...
200	volvo	gas	sedan	rwd	front	ohc
201	volvo	gas	sedan	rwd	front	ohc
202	volvo	gas	sedan	rwd	front	ohcv
203	volvo	diesel	sedan	rwd	front	ohc

	make	fuel-type	body-style	drive-wheels	engine-location	engine-type
204	volvo	gas	sedan	rwd	front	ohc

191 rows × 6 columns

In [ ]:

Label encoding

In [59]:
```
#import module
from sklearn.preprocessing import LabelEncoder
```

In [60]:
```
#initialize the class
le=LabelEncoder()
```

In [61]:
```
#applying encoder
for col in df_cat:
    le=LabelEncoder()
    df_cat[col]=le.fit_transform(df_cat[col])
```

In [62]:
```
df_cat
```

Out[62]:

	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
...	...	...	...	...	...	...
200	20	1	3	2	0	3
201	20	1	3	2	0	3
202	20	1	3	2	0	5
203	20	0	3	2	0	3
204	20	1	3	2	0	3

191 rows × 6 columns

In [63]:
```
df_num
```

Out[63]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price
0	3	122.0	64.1	48.8	130	111.0	21	27	13495
1	3	122.0	64.1	48.8	130	111.0	21	27	16500
2	1	122.0	65.5	52.4	152	154.0	19	26	16500
3	2	164.0	66.2	54.3	109	102.0	24	30	13950
4	2	164.0	66.4	54.3	136	115.0	18	22	17450
...	...	...	...	...	...	...	...	...	...
200	-1	95.0	68.9	55.5	141	114.0	23	28	16845
201	-1	95.0	68.8	55.5	141	160.0	19	25	19045
202	-1	95.0	68.9	55.5	173	134.0	18	23	21485
203	-1	95.0	68.9	55.5	145	106.0	26	27	22470
204	-1	95.0	68.9	55.5	141	114.0	19	25	22625

191 rows × 9 columns

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

In [65]:
```
df
```

Out[65]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price	make	fuel typ
0	3	122.0	64.1	48.8	130	111.0	21	27	13495	0	
1	3	122.0	64.1	48.8	130	111.0	21	27	16500	0	
2	1	122.0	65.5	52.4	152	154.0	19	26	16500	0	
3	2	164.0	66.2	54.3	109	102.0	24	30	13950	1	
4	2	164.0	66.4	54.3	136	115.0	18	22	17450	1	
...	...	...	...	...	...	...	...	...	...	...	.
200	-1	95.0	68.9	55.5	141	114.0	23	28	16845	20	
201	-1	95.0	68.8	55.5	141	160.0	19	25	19045	20	
202	-1	95.0	68.9	55.5	173	134.0	18	23	21485	20	
203	-1	95.0	68.9	55.5	145	106.0	26	27	22470	20	
204	-1	95.0	68.9	55.5	141	114.0	19	25	22625	20	

191 rows × 15 columns

In [ ]:

In [ ]:

sepration of x and y

In [66]:
```
x = df.iloc[:,:-1]
y = df.iloc[:,1]
```

In [67]:
```
y
```

Out[67]:
```
0      122.0
1      122.0
2      122.0
3      164.0
4      164.0
       ...
200     95.0
201     95.0
202     95.0
203     95.0
204     95.0
Name: normalized-losses, Length: 191, dtype: float64
```

train and test

In [68]:
```
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 [69]:
```
xtrain
```

Out[69]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price	make	fuel typ
57	3	150.0	65.7	49.6	70	101.0	17	23	13645	8	
203	-1	95.0	68.9	55.5	145	106.0	26	27	22470	20	
125	3	186.0	68.3	50.2	151	143.0	19	27	22018	15	
40	0	85.0	62.5	54.1	110	86.0	27	33	10295	5	
28	-1	110.0	64.6	59.8	122	88.0	24	30	8921	4	
...	...	...	...	...	...	...	...	...	...	...	
142	0	102.0	65.4	52.5	108	82.0	28	33	7775	17	
146	0	89.0	65.4	53.0	108	82.0	28	32	7463	17	
75	1	122.0	68.0	54.8	140	175.0	19	24	16503	10	
149	0	85.0	65.4	54.9	108	111.0	23	23	11694	17	

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highway-mpg	price	make	fuel typ
38	0	106.0	65.2	53.3	110	86.0	27	33	9095	5	

133 rows × 14 columns

In [70]:
```
ytrain
```

Out[70]:
```
57     150.0
203     95.0
125    186.0
40      85.0
28     110.0
       ...
142    102.0
146     89.0
75     122.0
149     85.0
38     106.0
Name: normalized-losses, Length: 133, dtype: float64
```

model building

In [71]:
```
from sklearn.linear_model import LinearRegression
```

In [72]:
```
lr = LinearRegression()
```

In [73]:
```
lr.fit(xtrain,ytrain)
```

Out[73]:
```
LinearRegression()
```

In [74]:
```
ypred = lr.predict(xtest)
```

In [75]:
```
ypred
```

Out[75]:
```
array([101.,  93., 119., 168., 125., 122.,  78., 122.,  74., 194., 150.,
       122., 161., 134.,  95.,  93., 122., 107., 164., 104., 113., 122.,
        83.,  91., 137., 150., 192., 122.,  94., 129., 122., 122.,  94.,
        98., 153., 122., 119., 128., 115.,  65., 101., 128., 104., 129.,
       106., 122., 122., 122., 122., 104.,  65., 122., 102., 108.,  91.,
       121.,  89., 142.])
```

In [76]:
```
from sklearn.metrics import r2_score
print("Accuracy of model ",round(r2_score(ytest,ypred)*100,2),"%")
```

```
Accuracy of model  100.0 %
```

In [77]:
```
#xtest
from sklearn.preprocessing import OrdinalEncoder
oe1=OrdinalEncoder(categories=[["4wd","fwd","rwd"],["front","rear"]])
x1=oe1.fit_transform([['rwd', 'front']])
```

In [78]:
```
x1
```

Out[78]:
```
array([[2., 0.]])
```

In [79]:
```
x1[0][0]
```

Out[79]:
```
2.0
```

In [80]:
```
x1[0][1]
```

Out[80]:
```
0.0
```

In [81]:
```
import pickle
pickle.dump(oe1 , open("encode.pkl" , "wb"))
```

In [82]:
```
import pickle
pickle.dump(lr,open("predict.pkl" , "wb"))
```

In [ ]:
```

```