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

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

Out[2]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type
0	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc
1	3	?	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc
2	1	?	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv
3	2	164	audi	gas	sedan	fwd	front	66.2	54.3	ohc
4	2	164	audi	gas	sedan	4wd	front	66.4	54.3	ohc
...	...	...	...	...	...	...	...	...	...	...
200	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohc
201	-1	95	volvo	gas	sedan	rwd	front	68.8	55.5	ohc
202	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohcv
203	-1	95	volvo	diesel	sedan	rwd	front	68.9	55.5	ohc
204	-1	95	volvo	gas	sedan	rwd	front	68.9	55.5	ohc

205 rows × 15 columns

In [3]:
```
1 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 [4]:
```
1 df["normalized-losses"].value_counts()
```

Out[4]:
```
?      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 [5]:
```
df["horsepower"].value_counts()
```

Out[5]:

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 [6]:
```
df["normalized-losses"].replace("?",np.nan,inplace=True)
df["horsepower"].replace("?",np.nan,inplace=True)
```

In [7]:
```
df["normalized-losses"]=df["normalized-losses"].astype("float")
df["horsepower"]=df["horsepower"].astype("float")
```

In [8]:
```
nmean=df["normalized-losses"].mean()
hmean=df["horsepower"].mean()
```

In [9]:
```
df["normalized-losses"].fillna(nmean,inplace=True)
df["horsepower"].fillna(hmean,inplace=True)
```

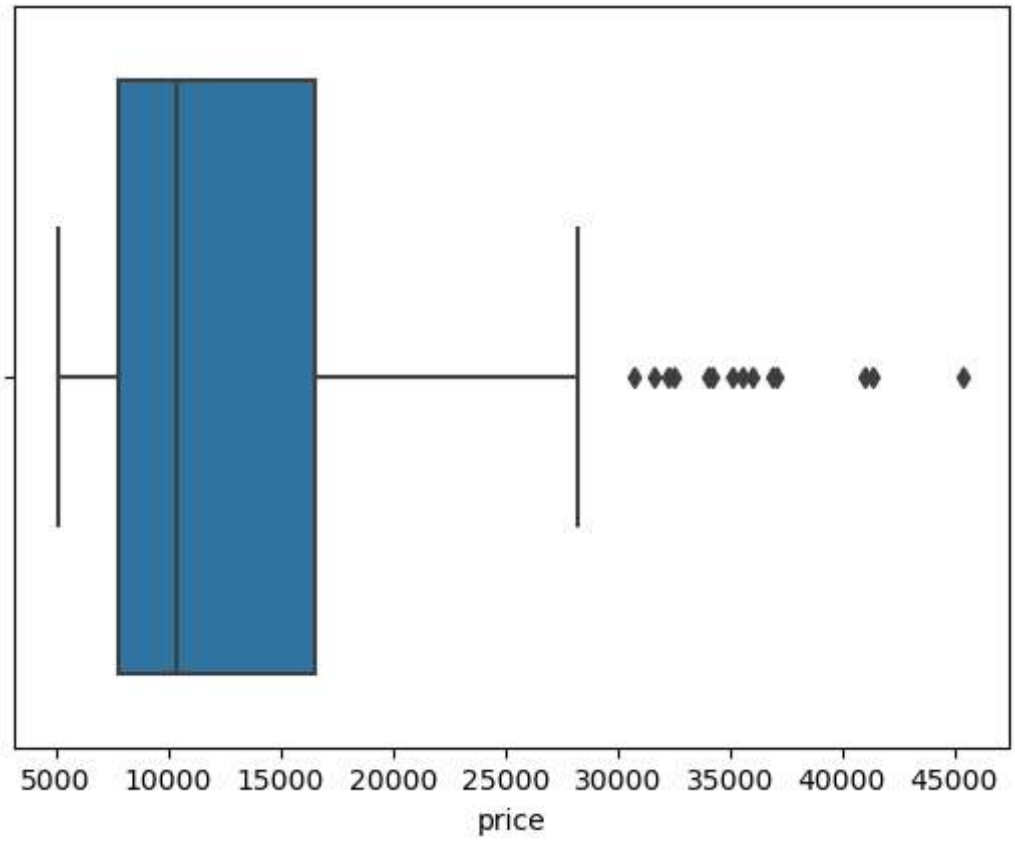
In [10]:
```
df
```

Out[10]:

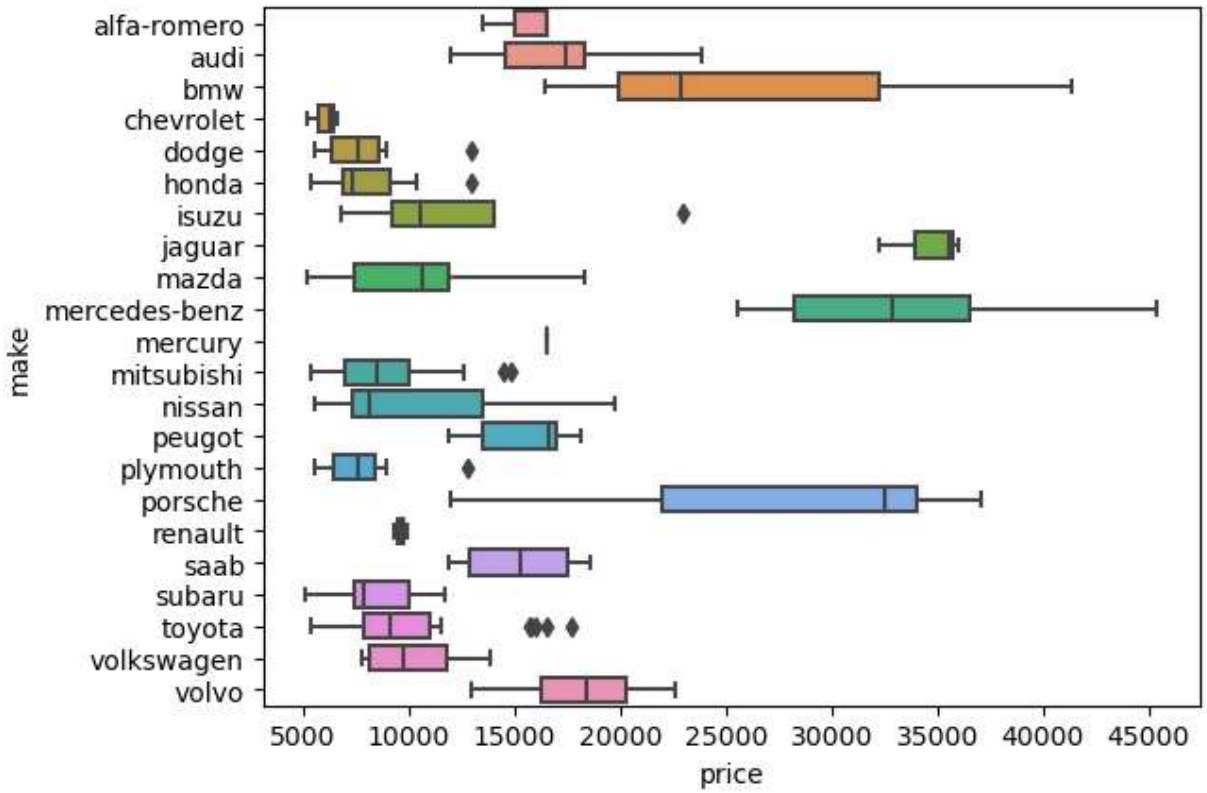
	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv
3	2	164.0	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 [11]:
```
1 sns.boxplot(data=df,x="price")
2 plt.show()
```



In [12]:
```
sns.boxplot(data=df,x="price",y="make")
plt.show()
```



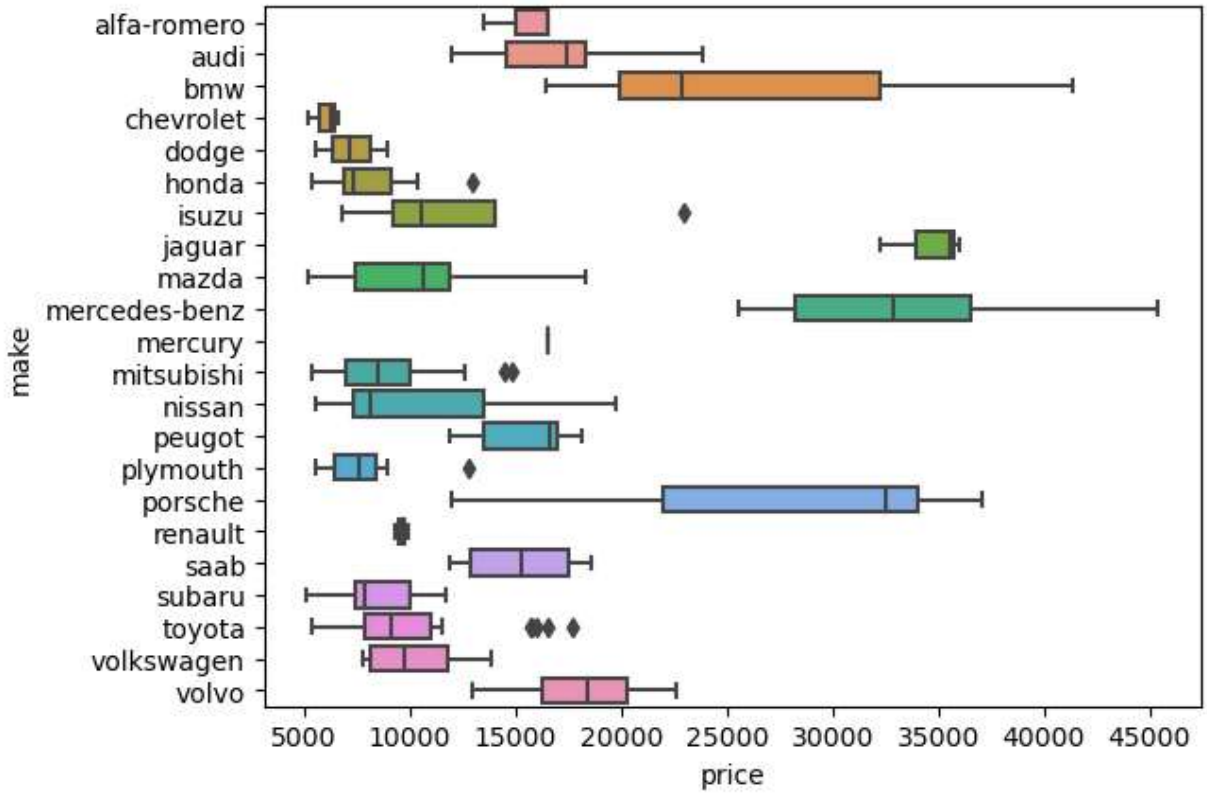
In [13]:
```
1 df[(df.make=="dodge") & (df.price>10000)]
```

Out[13]:

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

In [14]:
```
1 df.drop(29,axis=0,inplace=True)
```

In [15]:
```
1 sns.boxplot(data=df,x="price",y="make")
2 plt.show()
```



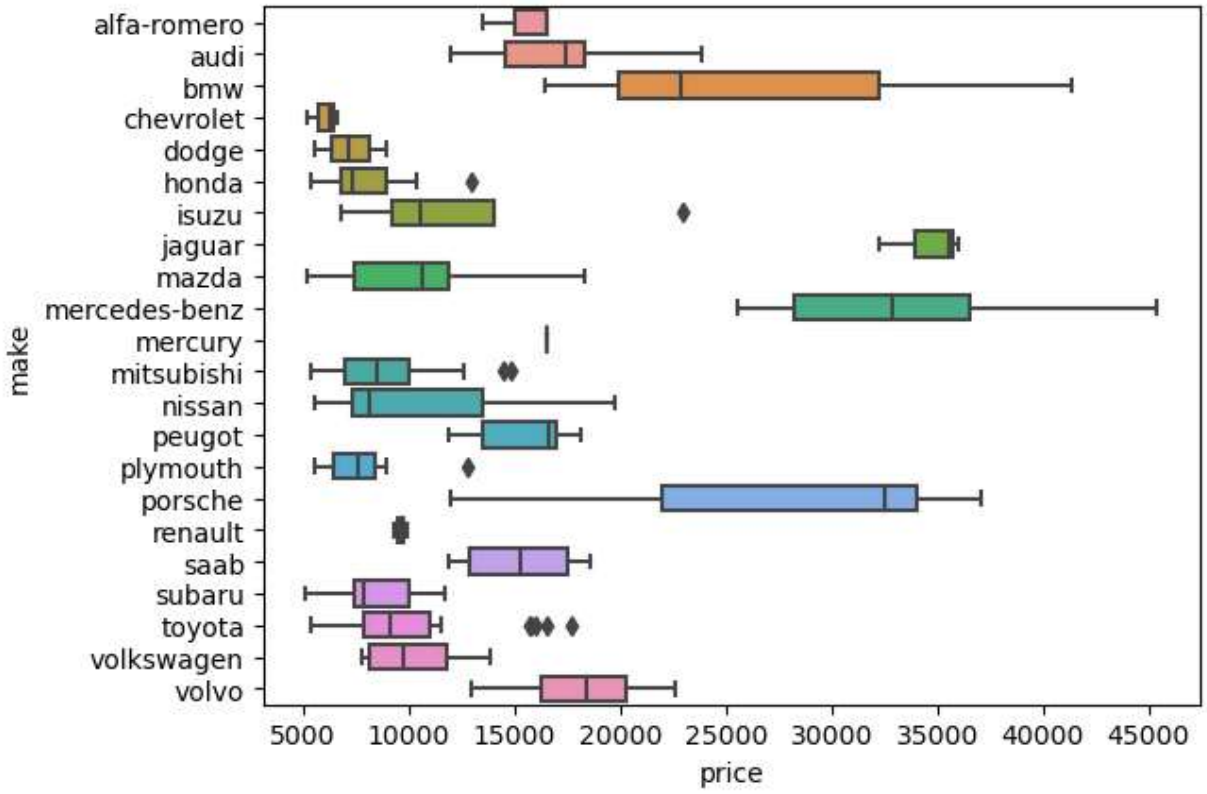
In [16]:
```
1 df[(df.make=="honda") & (df.price>10000)]
```

Out[16]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size
40	0	85.0	honda	gas	sedan	fwd	front	62.5	54.1	ohc	110
41	0	85.0	honda	gas	sedan	fwd	front	65.2	54.1	ohc	110
42	1	107.0	honda	gas	sedan	fwd	front	66.0	51.0	ohc	110

In [17]:
```
df.drop(40,axis=0,inplace=True)
```

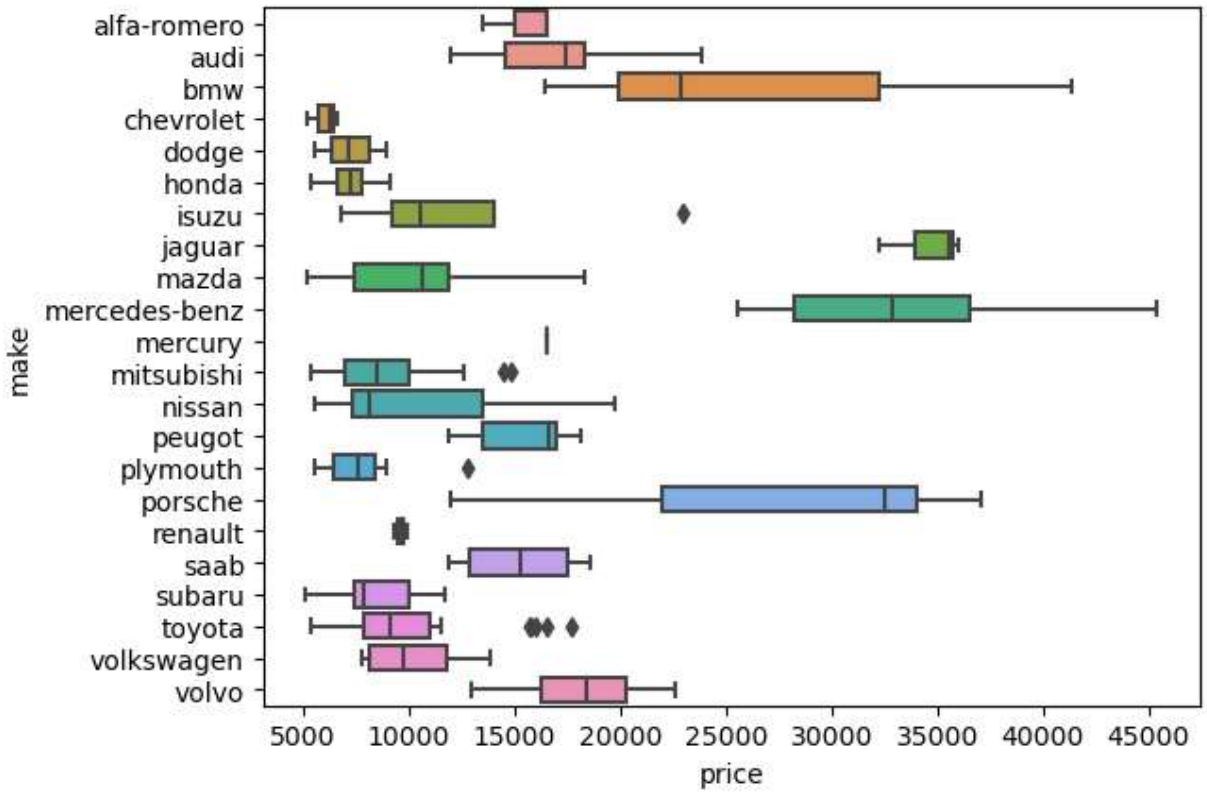
In [18]:
```
sns.boxplot(data=df,x="price",y="make")
plt.show()
```



In [19]:
```
df.drop(41,axis=0,inplace=True)
```

In [20]:
```
df.drop(42,axis=0,inplace=True)
```

In [21]:
```
sns.boxplot(data=df,x="price",y="make")
plt.show()
```



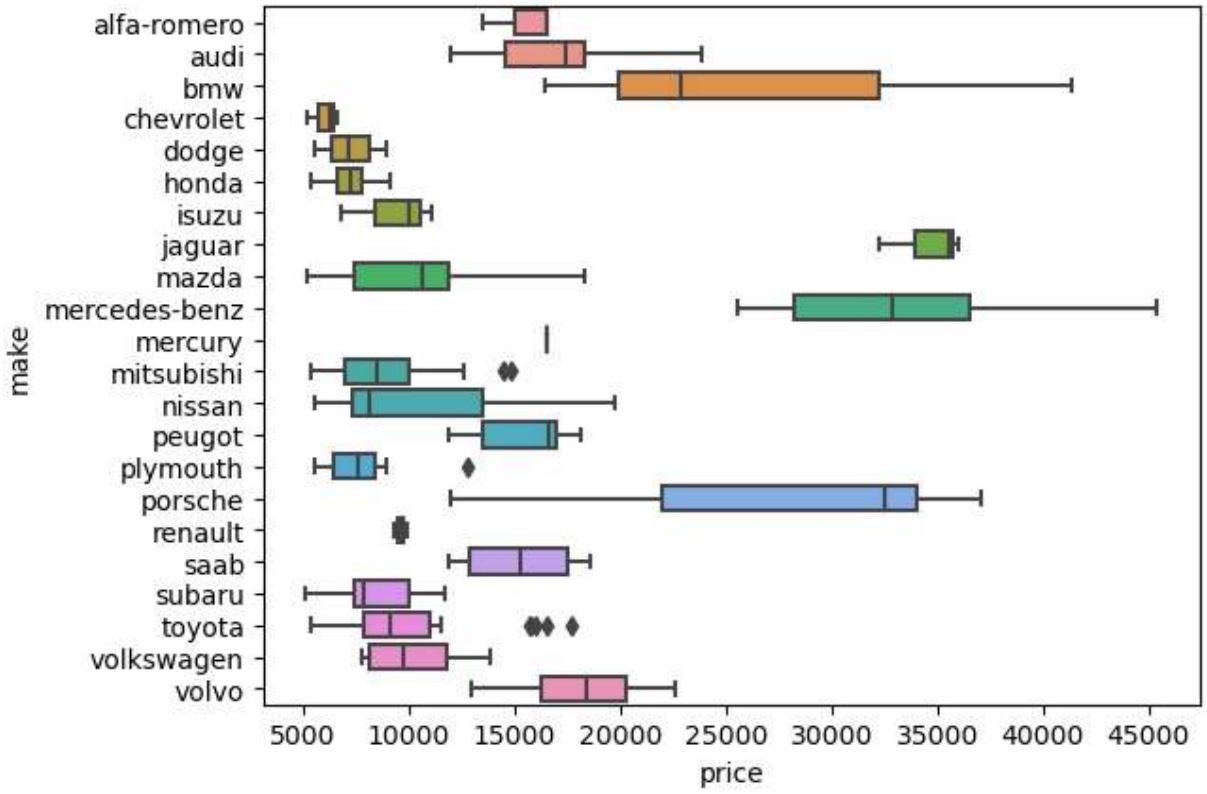
In [22]:
```
df[(df.make=="isuzu") & (df.price>20000)]
```

Out[22]:

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

In [23]:
```
df.drop(45,axis=0,inplace=True)
```

In [24]:
```
sns.boxplot(data=df,x="price",y="make")
plt.show()
```



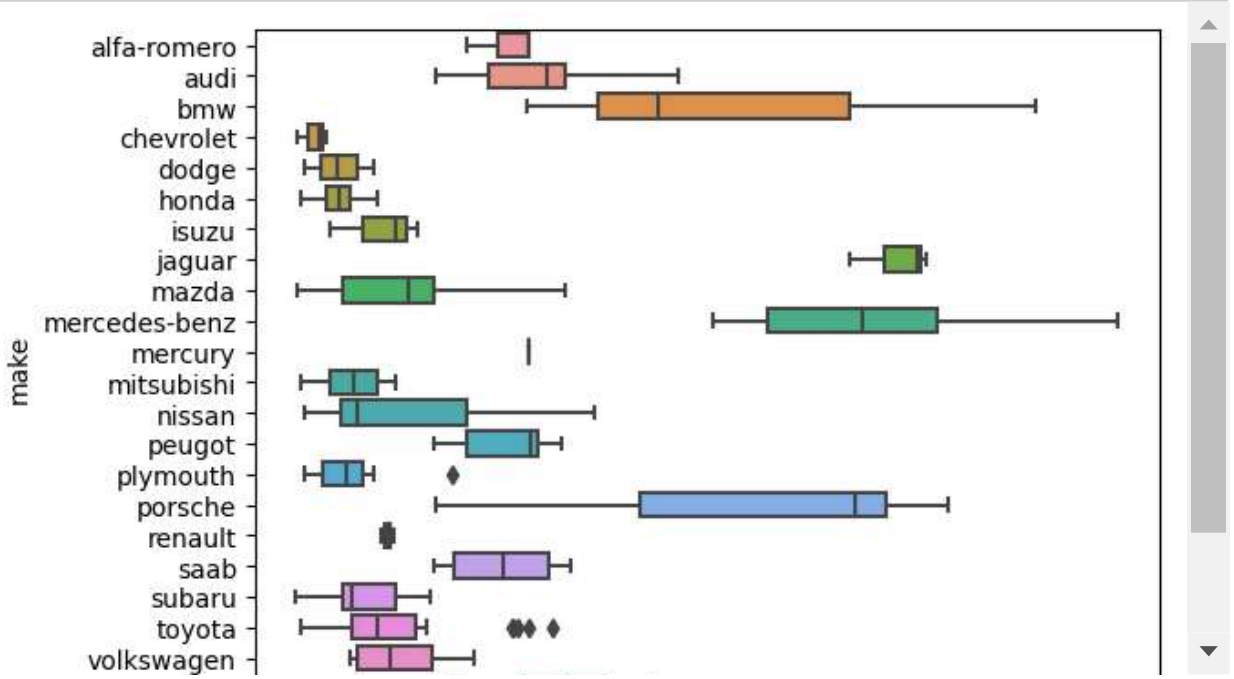
In [25]:
```
df[(df.make=="mitsubishi") & (df.price>12000)]
```

Out[25]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	e
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 [26]:
```
df.drop(list(range(82,85)),axis=0,inplace=True)
```

In [27]:
```
1 sns.boxplot(data=df,x="price",y="make")
2 plt.show()
```



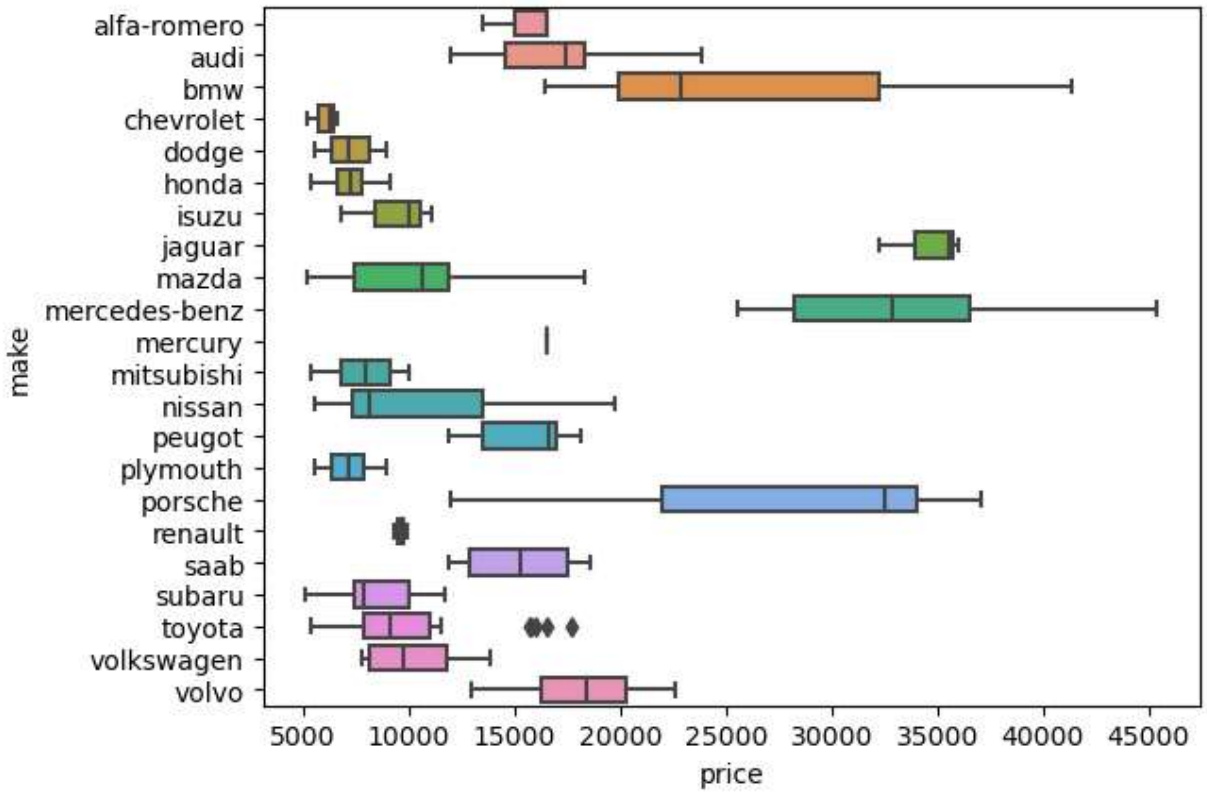
In [28]:
```
1 df[(df.make=="plymouth") & (df.price>12500)]
```

Out[28]:

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

In [29]:
```
1 df.drop(124,axis=0,inplace=True)
```

In [30]:
```
sns.boxplot(data=df,x="price",y="make")
plt.show()
```



In [31]:
```
df[(df.make=="toyota") & (df.price>15000)]
```

Out[31]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
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 [32]:
```
df.drop(list(range(178,182)),axis=0,inplace=True)
```

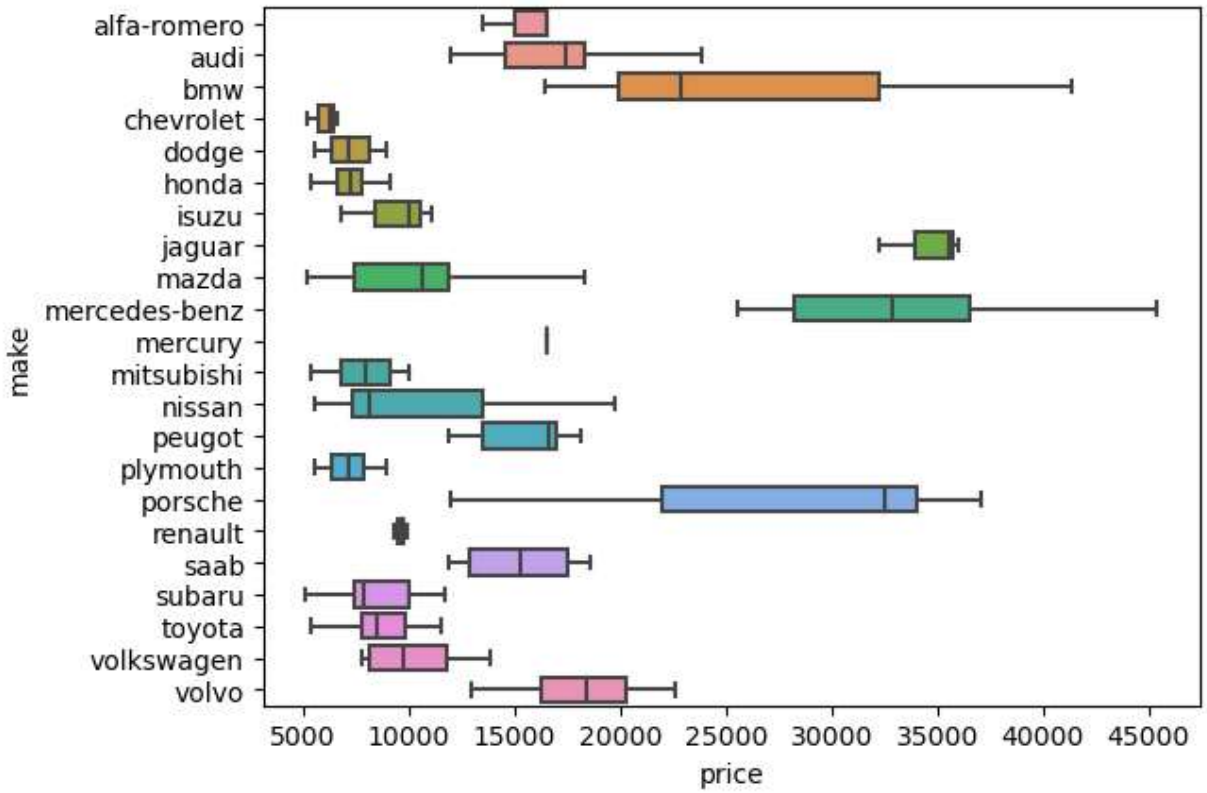
In [33]:
```
df[(df.make=="toyota") & (df.price>15000)]
```

Out[33]:

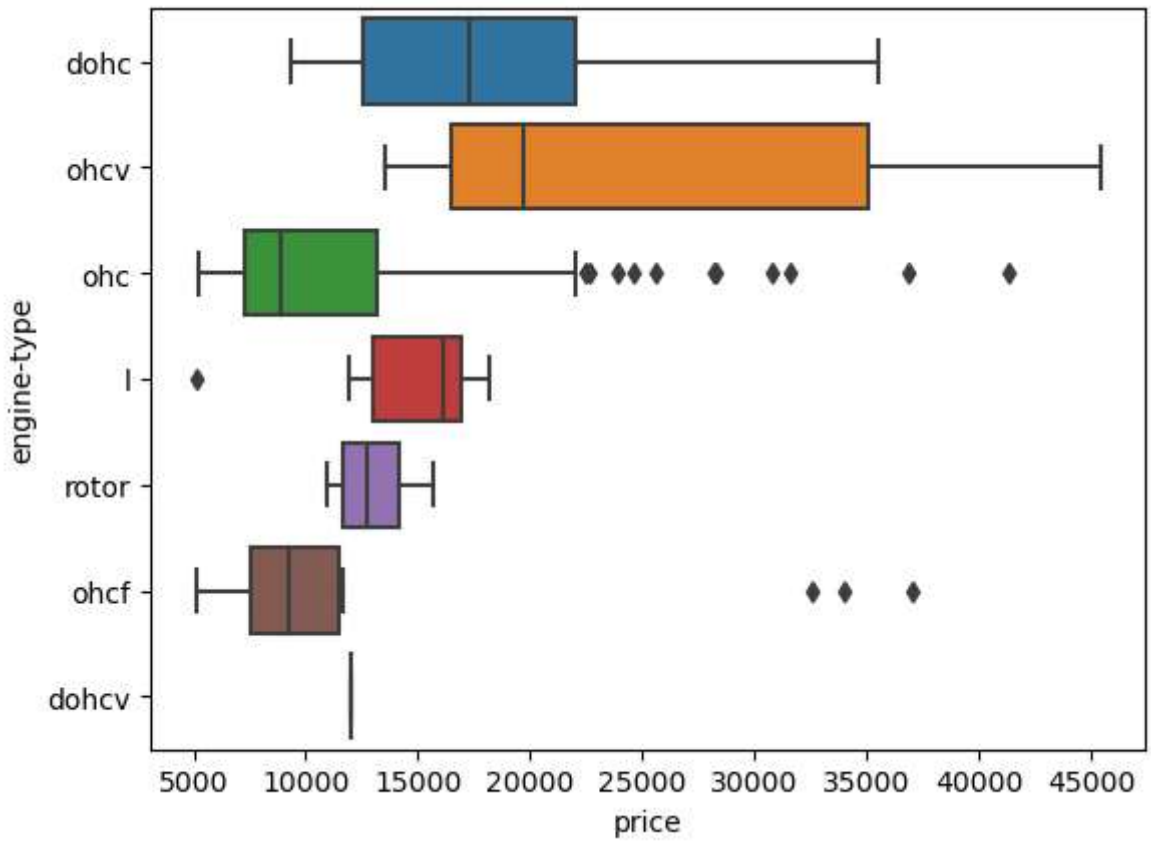
	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
172	2	134.0	toyota	gas	convertible	rwd	front	65.6	53.0	ohc	

In [34]:
```
df.drop(172,axis=0,inplace=True)
```

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



In [36]:
```
1 sns.boxplot(data=df,x="price",y="engine-type")
2 plt.show()
```



In [37]:
```
1 df.select_dtypes(["int64","float64"])
```

Out[37]:

	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 [38]:
```
colname=df.select_dtypes(["int64","float64"]).columns
```

In [39]:
```
colname
```

Out[39]:
```
Index(['symboling', 'normalized-losses', 'width', 'height', 'engine-size',
       'horsepower', 'city-mpg', 'highway-mpg', 'price'],
      dtype='object')
```

In [40]:
```
from scipy.stats import skew
```

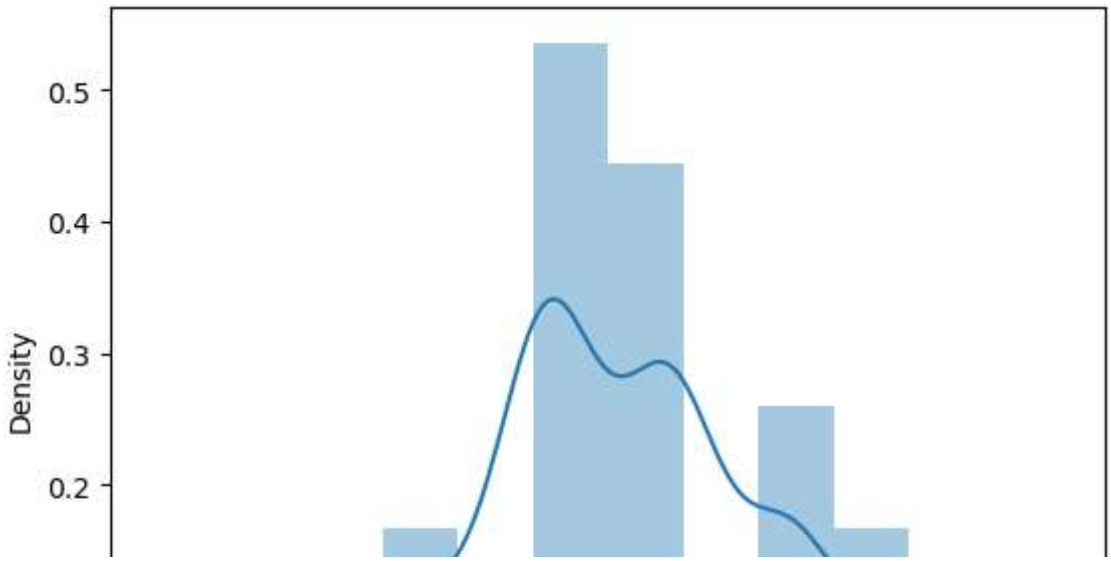
In [41]:
```
skew(df["normalized-losses"])
```

Out[41]:
```
0.8355271321292326
```

In [42]:
```
for col in df[colname]:
    print(col)
    print(skew(df[col]))

    sns.distplot(df[col])
    plt.show()
```

```
symboling
0.20852128897880182
```



In [43]:
```
1 df.corr()
```

Out[43]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highv
symboling	1.000000	0.450621	-0.284079	-0.515772	-0.163469	0.015563	0.017671	0.096
normalized-losses	0.450621	1.000000	0.057961	-0.375559	0.087607	0.182025	-0.212334	-0.168
width	-0.284079	0.057961	1.000000	0.303279	0.737734	0.645904	-0.644322	-0.680
height	-0.515772	-0.375559	0.303279	1.000000	0.100717	-0.070930	-0.087946	-0.152
engine-size	-0.163469	0.087607	0.737734	0.100717	1.000000	0.803458	-0.641732	-0.666
horsepower	0.015563	0.182025	0.645904	-0.070930	0.803458	1.000000	-0.796057	-0.759
city-mpg	0.017671	-0.212334	-0.644322	-0.087946	-0.641732	-0.796057	1.000000	0.969
highway-mpg	0.096186	-0.168241	-0.680900	-0.152771	-0.666019	-0.759531	0.969967	1.000
price	-0.097242	0.127662	0.732957	0.146146	0.870876	0.771584	-0.682634	-0.707

In [44]:
```
1 np.log(-5)
```

Out[44]: nan

In [45]:
```
1 np.sqrt(-3)
```

Out[45]: nan

In [46]:
```
1 df1=df.corr().style.background_gradient()
```

In [47]:
```
1 skew(df["normalized-losses"])
```

Out[47]: 0.8355271321292326

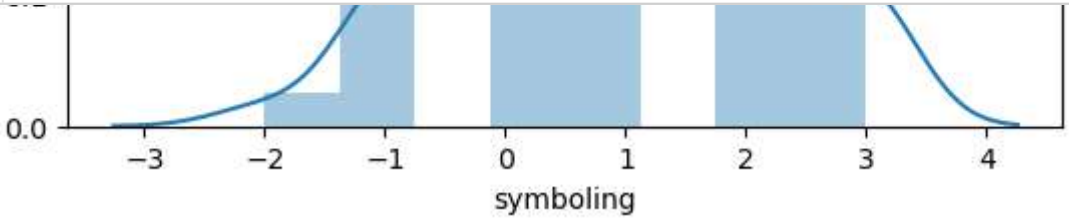
In [48]:
```
1 df["normalized-losses"]=np.log(df["normalized-losses"])
```

In [49]:
```
1 skew(df["normalized-losses"])
```

Out[49]: 0.0061510764084907665

In [50]:
```
for col in df[colname]:
    print(col)
    print(skew(df[col]))

    sns.distplot(df[col])
    plt.show()
```




```
normalized-losses
0.0061510764084907665
```

In [51]:
```
df.corr().style.background_gradient()
```

Out[51]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highv
symboling	1.000000	0.470951	-0.284079	-0.515772	-0.163469	0.015563	0.017671	0.096
normalized-losses	0.470951	1.000000	0.065586	-0.385517	0.100651	0.192955	-0.215013	-0.166
width	-0.284079	0.065586	1.000000	0.303279	0.737734	0.645904	-0.644322	-0.680
height	-0.515772	-0.385517	0.303279	1.000000	0.100717	-0.070930	-0.087946	-0.152
engine-size	-0.163469	0.100651	0.737734	0.100717	1.000000	0.803458	-0.641732	-0.666
horsepower	0.015563	0.192955	0.645904	-0.070930	0.803458	1.000000	-0.796057	-0.759
city-mpg	0.017671	-0.215013	-0.644322	-0.087946	-0.641732	-0.796057	1.000000	0.969
highway-mpg	0.096186	-0.166963	-0.680900	-0.152771	-0.666019	-0.759531	0.969967	1.000
price	-0.097242	0.140551	0.732957	0.146146	0.870876	0.771584	-0.682634	-0.707

In [52]:
```
1 df.corr()
```

Out[52]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highv
symboling	1.000000	0.470951	-0.284079	-0.515772	-0.163469	0.015563	0.017671	0.096
normalized-losses	0.470951	1.000000	0.065586	-0.385517	0.100651	0.192955	-0.215013	-0.166
width	-0.284079	0.065586	1.000000	0.303279	0.737734	0.645904	-0.644322	-0.680
height	-0.515772	-0.385517	0.303279	1.000000	0.100717	-0.070930	-0.087946	-0.152
engine-size	-0.163469	0.100651	0.737734	0.100717	1.000000	0.803458	-0.641732	-0.666
horsepower	0.015563	0.192955	0.645904	-0.070930	0.803458	1.000000	-0.796057	-0.759
city-mpg	0.017671	-0.215013	-0.644322	-0.087946	-0.641732	-0.796057	1.000000	0.969
highway-mpg	0.096186	-0.166963	-0.680900	-0.152771	-0.666019	-0.759531	0.969967	1.000
price	-0.097242	0.140551	0.732957	0.146146	0.870876	0.771584	-0.682634	-0.707

In [53]:
```
1 df1=df.corr().style.background_gradient()
```

In [54]:
```
1 skew(df["width"])
```

Out[54]: 0.9233175535392493

In [55]:
```
1 df["width"]=np.log(df["width"])
```

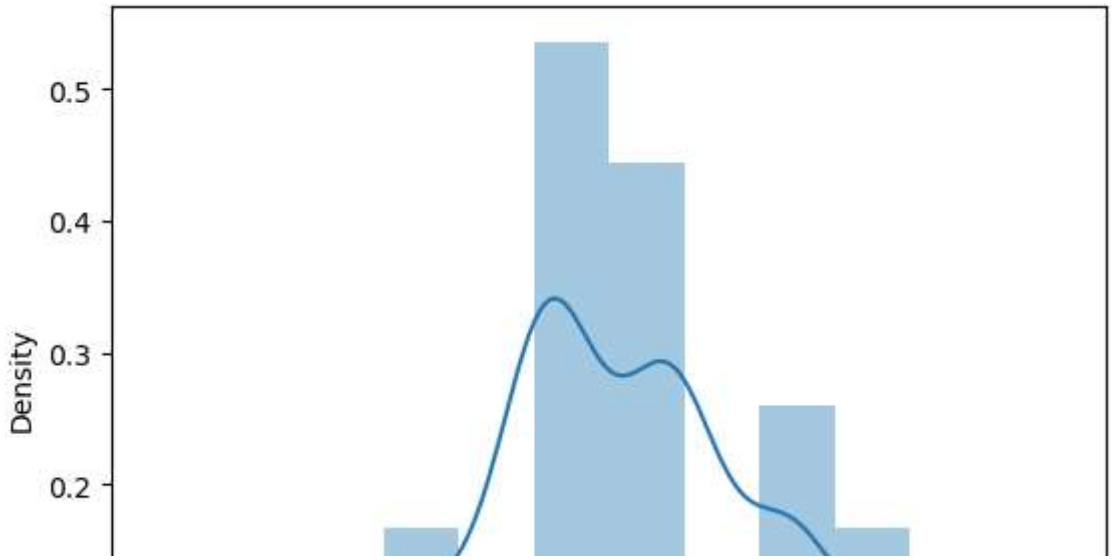
In [56]:
```
1 skew(df["width"])
```

Out[56]: 0.8390778744161934

In [57]:
```
for col in df[colname]:
    print(col)
    print(skew(df[col]))

    sns.distplot(df[col])
    plt.show()
```

```
symboling
0.20852128897880182
```



In [58]:
```
df.corr().style.background_gradient()
```

Out[58]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	highv
symboling	1.000000	0.470951	-0.284644	-0.515772	-0.163469	0.015563	0.017671	0.096
normalized-losses	0.470951	1.000000	0.065209	-0.385517	0.100651	0.192955	-0.215013	-0.166
width	-0.284644	0.065209	1.000000	0.304686	0.735431	0.646026	-0.648199	-0.683
height	-0.515772	-0.385517	0.304686	1.000000	0.100717	-0.070930	-0.087946	-0.152
engine-size	-0.163469	0.100651	0.735431	0.100717	1.000000	0.803458	-0.641732	-0.666
horsepower	0.015563	0.192955	0.646026	-0.070930	0.803458	1.000000	-0.796057	-0.759
city-mpg	0.017671	-0.215013	-0.648199	-0.087946	-0.641732	-0.796057	1.000000	0.969
highway-mpg	0.096186	-0.166963	-0.683760	-0.152771	-0.666019	-0.759531	0.969967	1.000
price	-0.097242	0.140551	0.730630	0.146146	0.870876	0.771584	-0.682634	-0.707

In [59]:
```
1 df.head()
```

Out[59]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	e
0	3	4.804021	alfa-romero	gas	convertible	rwd	front	4.160444	48.8	dohc	
1	3	4.804021	alfa-romero	gas	convertible	rwd	front	4.160444	48.8	dohc	
2	1	4.804021	alfa-romero	gas	hatchback	rwd	front	4.182050	52.4	ohcv	
3	2	5.099866	audi	gas	sedan	fwd	front	4.192680	54.3	ohc	
4	2	5.099866	audi	gas	sedan	4wd	front	4.195697	54.3	ohc	

In [60]:
```
1 df["fuel-type"]
```

Out[60]:
```
0         gas
1         gas
2         gas
3         gas
4         gas
        ...
200       gas
201       gas
202       gas
203    diesel
204       gas
Name: fuel-type, Length: 191, dtype: object
```

In [61]:
```
1 df["fuel-type"].value_counts()
```

Out[61]:
```
gas       171
diesel     20
Name: fuel-type, dtype: int64
```

In [62]:
```
1 pd.get_dummies(df["fuel-type"])
```

Out[62]:

	diesel	gas
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
...	...	...
200	0	1
201	0	1
202	0	1
203	1	0
204	0	1

191 rows × 2 columns

In [63]:
```
1 pd.get_dummies(df["make"])
```

Out[63]:

	alfa-romero	audi	bmw	chevrolet	dodge	honda	isuzu	jaguar	mazda	mercedes-benz	...	nissan
0	1	0	0	0	0	0	0	0	0	0	...	0
1	1	0	0	0	0	0	0	0	0	0	...	0
2	1	0	0	0	0	0	0	0	0	0	...	0
3	0	1	0	0	0	0	0	0	0	0	...	0
4	0	1	0	0	0	0	0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
200	0	0	0	0	0	0	0	0	0	0	...	0
201	0	0	0	0	0	0	0	0	0	0	...	0
202	0	0	0	0	0	0	0	0	0	0	...	0
203	0	0	0	0	0	0	0	0	0	0	...	0
204	0	0	0	0	0	0	0	0	0	0	...	0

191 rows × 22 columns

In [64]:
```
1 from sklearn.preprocessing import OrdinalEncoder
```

In [65]:
```
oe = OrdinalEncoder()
```

In [66]:
```
oe.fit_transform(df[["make","body-style"]])
```

```
array([[ 0.,  2.],
       [ 0.,  2.],
       [ 0.,  2.],
       [ 1.,  2.],
       [ 1.,  2.],
       [ 1.,  2.],
       [ 2.,  2.],
       [ 2.,  2.],
       [ 2.,  2.],
       [ 3.,  2.],
       [ 3.,  2.],
       [ 3.,  2.],
       [ 4.,  2.],
       [ 4.,  2.],
       [ 4.,  2.],
       [ 5.,  2.],
       [ 5.,  2.],
       [ 5.,  2.],
       [ 6.,  2.],
       [ 6.,  2.],
       [ 6.,  2.],
       [ 7.,  2.],
       [ 7.,  2.],
       [ 7.,  2.],
       [ 8.,  3.],
       [ 8.,  3.],
       [ 8.,  2.],

       [ 8.,  2.],
       [ 8.,  2.],
       [ 8.,  2.],
       [ 8.,  2.],
       [ 8.,  3.],
       [ 8.,  2.],
       [ 8.,  3.],
       [ 8.,  3.],
       [ 8.,  2.],
       [ 8.,  3.],
       [ 8.,  3.],
       [ 9.,  3.],
       [ 9.,  4.],
       [ 9.,  1.],
       [ 9.,  3.],
       [ 9.,  3.],
```

In [67]:
```
catcol = df.select_dtypes(object).columns
```

In [68]:
```
from sklearn.preprocessing import OrdinalEncoder
```

In [69]:
```
oe = OrdinalEncoder()
```

In [70]:
```
df[catcol]=oe.fit_transform(df[catcol])
```

In [71]:
```
1 df
```

Out[71]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engi s
0	3	4.804021	0.0	1.0	0.0	2.0	0.0	4.160444	48.8	0.0	
1	3	4.804021	0.0	1.0	0.0	2.0	0.0	4.160444	48.8	0.0	
2	1	4.804021	0.0	1.0	2.0	2.0	0.0	4.182050	52.4	5.0	
3	2	5.099866	1.0	1.0	3.0	1.0	0.0	4.192680	54.3	3.0	
4	2	5.099866	1.0	1.0	3.0	0.0	0.0	4.195697	54.3	3.0	
...	...	...	...	...	...	...	...	...	...	...	
200	-1	4.553877	21.0	1.0	3.0	2.0	0.0	4.232656	55.5	3.0	
201	-1	4.553877	21.0	1.0	3.0	2.0	0.0	4.231204	55.5	3.0	
202	-1	4.553877	21.0	1.0	3.0	2.0	0.0	4.232656	55.5	5.0	
203	-1	4.553877	21.0	0.0	3.0	2.0	0.0	4.232656	55.5	3.0	
204	-1	4.553877	21.0	1.0	3.0	2.0	0.0	4.232656	55.5	3.0	

191 rows × 15 columns

In [72]:
```
1 from sklearn.preprocessing import StandardScaler
```

In [73]:
```
1 ss = StandardScaler()
```

In [74]:
```
df.iloc[:,:-1]=ss.fit_transform(df.iloc[:,:-1])
df
```

Out[74]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	h
0	1.857613	0.137829	-1.945079	0.341993	-3.094345	1.219274	-0.126323	-0.833279	-2.07
1	1.857613	0.137829	-1.945079	0.341993	-3.094345	1.219274	-0.126323	-0.833279	-2.07
2	0.184009	0.137829	-1.945079	0.341993	-0.744365	1.219274	-0.126323	-0.172960	-0.59
3	1.020811	1.303434	-1.785837	0.341993	0.430625	-0.572121	-0.126323	0.151925	0.18
4	1.020811	1.303434	-1.785837	0.341993	0.430625	-2.363516	-0.126323	0.244119	0.18
...	...	...	...	...	...	...	...	...	
200	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67
201	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.329280	0.67
202	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67
203	-1.489596	-0.847717	1.398989	-2.924038	0.430625	1.219274	-0.126323	1.373669	0.67
204	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67

191 rows × 15 columns

In [75]:
```
x = df.iloc[:,:-1]
y = df.iloc[:,-1]
```

In [76]:
```
y
```

Out[76]:
```
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 [77]:
```
1 x
```

Out[77]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	h
0	1.857613	0.137829	-1.945079	0.341993	-3.094345	1.219274	-0.126323	-0.833279	-2.07
1	1.857613	0.137829	-1.945079	0.341993	-3.094345	1.219274	-0.126323	-0.833279	-2.07
2	0.184009	0.137829	-1.945079	0.341993	-0.744365	1.219274	-0.126323	-0.172960	-0.59
3	1.020811	1.303434	-1.785837	0.341993	0.430625	-0.572121	-0.126323	0.151925	0.18
4	1.020811	1.303434	-1.785837	0.341993	0.430625	-2.363516	-0.126323	0.244119	0.18
...	...	...	...	...	...	...	...	...	
200	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67
201	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.329280	0.67
202	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67
203	-1.489596	-0.847717	1.398989	-2.924038	0.430625	1.219274	-0.126323	1.373669	0.67
204	-1.489596	-0.847717	1.398989	0.341993	0.430625	1.219274	-0.126323	1.373669	0.67

191 rows × 14 columns

In [78]:
```
1 x.shape
```

Out[78]: (191, 14)

In [79]:
```
1 y.shape
```

Out[79]: (191,)

In [80]:
```
1 from sklearn.model_selection import train_test_split
2 xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [81]:
```
1 xtrain
```

Out[81]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	h
59	0.184009	0.357642	-0.671148	0.341993	-0.744365	-0.572121	-0.126323	0.290112	-0.06
203	-1.489596	-0.847717	1.398989	-2.924038	0.430625	1.219274	-0.126323	1.373669	0.67
127	1.857613	0.137829	0.443541	0.341993	-1.919355	1.219274	7.916228	-0.407154	-0.92
43	-0.652793	0.137829	-0.989631	0.341993	0.430625	1.219274	-0.126323	-1.950051	-0.14
28	-1.489596	-0.270112	-1.308113	0.341993	1.605614	-0.572121	-0.126323	-0.595810	2.43
...	...	...	...	...	...	...	...	...	
142	-0.652793	-0.567605	0.921265	0.341993	0.430625	-0.572121	-0.126323	-0.219656	-0.55
146	-0.652793	-1.104759	0.921265	0.341993	1.605614	-0.572121	-0.126323	-0.219656	-0.35
77	1.020811	1.230695	-0.193424	0.341993	-0.744365	-0.572121	-0.126323	-0.690576	-1.25
149	-0.652793	-1.285936	0.921265	0.341993	1.605614	-2.363516	-0.126323	-0.219656	0.42
38	-0.652793	-0.416051	-1.148872	0.341993	-0.744365	-0.572121	-0.126323	-0.313261	-0.22

133 rows × 14 columns

In [82]:
```
1 ytrain
```

Out[82]:
```
59      8845
203    22470
127    34028
43      6785
28      8921
       ...
142     7775
146     7463
77      6189
149    11694
38      9095
Name: price, Length: 133, dtype: int64
```

In [ ]:
```
1
```