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

Handling missing values

In [4]:
```
df["normalized-losses"].unique()
```

Out[4]:
```
array(['?', '164', '158', '192', '188', '121', '98', '81', '118', '148',
       '110', '145', '137', '101', '78', '106', '85', '107', '104', '113',
       '150', '129', '115', '93', '142', '161', '153', '125', '128',
       '122', '103', '168', '108', '194', '231', '119', '154', '74',
       '186', '83', '102', '89', '87', '77', '91', '134', '65', '197',
       '90', '94', '256', '95'], dtype=object)
```

In [5]:
```
df["horsepower"].unique()
```

Out[5]:
```
array(['111', '154', '102', '115', '110', '140', '160', '101', '121',
       '182', '48', '70', '68', '88', '145', '58', '76', '60', '86',
       '100', '78', '90', '176', '262', '135', '84', '64', '120', '72',
       '123', '155', '184', '175', '116', '69', '55', '97', '152', '200',
       '95', '142', '143', '207', '288', '?', '73', '82', '94', '62',
       '56', '112', '92', '161', '156', '52', '85', '114', '162', '134',
       '106'], dtype=object)
```

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]:
```
df["normalized-losses"].fillna(df["normalized-losses"].mean(),inplace=True)
df["horsepower"].fillna(df["horsepower"].mean(),inplace=True)
```

In [9]:
```
df
```

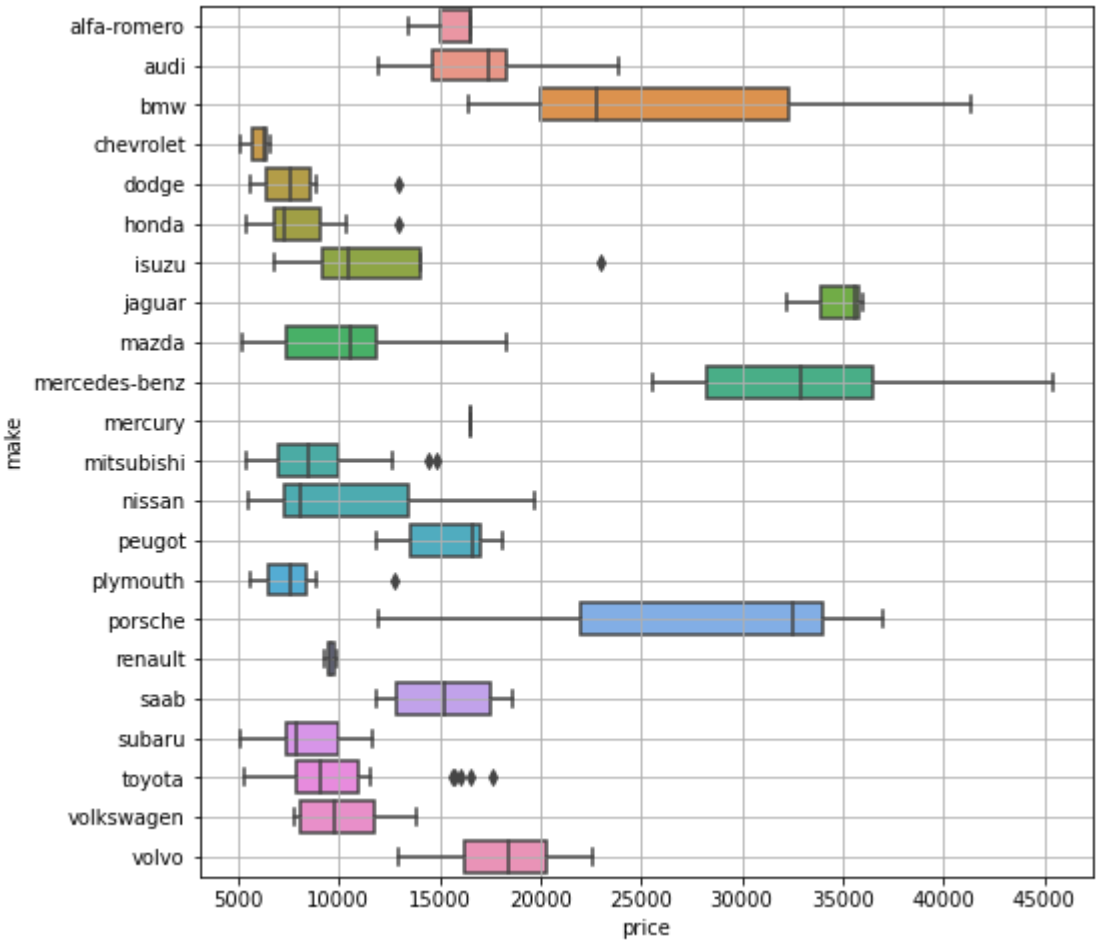
Out[9]:

	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

Outliers removal

In [10]:
```
plt.figure(figsize=(8,8))
sns.boxplot(data=df,x="price",y="make")
plt.grid()
plt.show()
```



In [11]:
```
df[(df["make"]=="dodge") & (df["price"]>10000)]
```

Out[11]:

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

In [12]:
```
df[(df["make"]=="honda") & (df["price"]>12000)]
```

Out[12]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engin siz
41	0	85.0	honda	gas	sedan	fwd	front	65.2	54.1	ohc	11

In [13]:
```
df[(df["make"]=="isuzu") & (df["price"]>20000)]
```

Out[13]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine siz
45	0	122.0	isuzu	gas	sedan	fwd	front	63.6	52.0	ohc	9

In [14]:
```
df[(df["make"]=="mitsubishi") & (df["price"]>13000)]
```

Out[14]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type
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 [15]:
```
df[(df["make"]=="plymouth") & (df["price"]>12000)]
```

Out[15]:

	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 [16]:
```
df[(df["make"]=="toyota") & (df["price"]>15000)]
```

Out[16]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	e
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 [17]:
```
df.drop([29,41,45,83,84,124,172,178,179,180,181],axis=0,inplace=True)
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 194 entries, 0 to 204
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          194 non-null    int64  
 1   normalized-losses  194 non-null    float64
 2   make               194 non-null    object 
 3   fuel-type          194 non-null    object 
 4   body-style         194 non-null    object 
 5   drive-wheels       194 non-null    object 
 6   engine-location    194 non-null    object 
 7   width              194 non-null    float64
 8   height             194 non-null    float64
 9   engine-type        194 non-null    object 
 10  engine-size        194 non-null    int64  
 11  horsepower         194 non-null    float64
 12  city-mpg           194 non-null    int64  
 13  highway-mpg        194 non-null    int64  
 14  price              194 non-null    int64  
dtypes: float64(4), int64(5), object(6)
memory usage: 24.2+ KB
```

skewness removal

In [19]:
```
colname=df.select_dtypes(["int","float"]).columns
colname
```

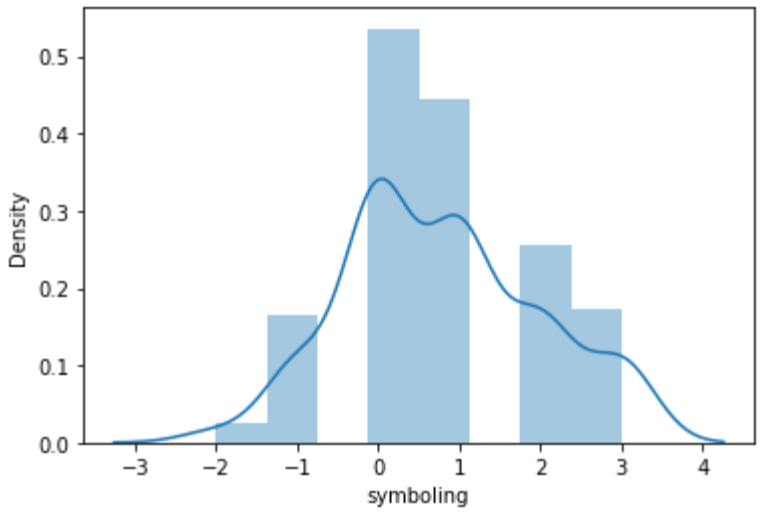
Out[19]:
```
Index(['symboling', 'normalized-losses', 'width', 'height', 'engine-size',
       'horsepower', 'city-mpg', 'highway-mpg', 'price'],
      dtype='object')
```

In [20]:
```
from scipy.stats import skew
```

In [21]:
```
for col in df[colname]:
    print(col)
    print(skew(df[col]))

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

```
symboling
0.21386866184357742
```



normalized-losses

In [22]:
```
df.corr().style.background_gradient()
```

Out[22]:

	symboling	normalized-losses	width	height	engine-size	horsepower	city-mpg	hig
symboling	1.000000	0.447922	-0.272388	-0.521495	-0.153671	0.027074	0.007189	0.0
normalized-losses	0.447922	1.000000	0.066622	-0.368540	0.090258	0.183385	-0.212276	-0.1
width	-0.272388	0.066622	1.000000	0.296011	0.735112	0.643906	-0.641401	-0.6
height	-0.521495	-0.368540	0.296011	1.000000	0.096041	-0.078245	-0.078815	-0.1
engine-size	-0.153671	0.090258	0.735112	0.096041	1.000000	0.803956	-0.642711	-0.6
horsepower	0.027074	0.183385	0.643906	-0.078245	0.803956	1.000000	-0.797166	-0.7
city-mpg	0.007189	-0.212276	-0.641401	-0.078815	-0.642711	-0.797166	1.000000	0.9
highway-mpg	0.084238	-0.168904	-0.677911	-0.142926	-0.667078	-0.761009	0.970113	1.0
price	-0.095905	0.129973	0.730503	0.147010	0.869638	0.768921	-0.680412	-0.7

to remove skewness we need two things 1st in skew value which is close to 0 and 2d thing is correlation which is close to 1 and when both condition is not

satisfied that time we remove skewness

there are 2 ways to remove skewness by taking 1) log OR 2) square root

In [23]:
```
df["normalized-losses"]=np.sqrt(df["normalized-losses"])
```

In [24]:
```
skew(df["normalized-losses"])
```

Out[24]:
```
0.4136415061835428
```

Handling categorical data (Encoding)

In [25]:
```
catcol=df.select_dtypes(object).columns
catcol
```

Out[25]:
```
Index(['make', 'fuel-type', 'body-style', 'drive-wheels', 'engine-location',
       'engine-type'],
      dtype='object')
```

there are 3 main types of encoder are there 1) One hot encoder 2) Label encoder 3) Ordinal encoder

In [26]:
```
from sklearn.preprocessing import OrdinalEncoder
oe=OrdinalEncoder()
df[catcol]=oe.fit_transform(df[catcol])
```

In [27]:
```
df
```

Out[27]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engin si
0	3	11.045361	0.0	1.0	0.0	2.0	0.0	64.1	48.8	0.0	1
1	3	11.045361	0.0	1.0	0.0	2.0	0.0	64.1	48.8	0.0	1
2	1	11.045361	0.0	1.0	2.0	2.0	0.0	65.5	52.4	5.0	1
3	2	12.806248	1.0	1.0	3.0	1.0	0.0	66.2	54.3	3.0	1
4	2	12.806248	1.0	1.0	3.0	0.0	0.0	66.4	54.3	3.0	1
...	...	...	...	...	...	...	...	...	...	...	
200	-1	9.746794	21.0	1.0	3.0	2.0	0.0	68.9	55.5	3.0	1
201	-1	9.746794	21.0	1.0	3.0	2.0	0.0	68.8	55.5	3.0	1
202	-1	9.746794	21.0	1.0	3.0	2.0	0.0	68.9	55.5	5.0	1
203	-1	9.746794	21.0	0.0	3.0	2.0	0.0	68.9	55.5	3.0	1
204	-1	9.746794	21.0	1.0	3.0	2.0	0.0	68.9	55.5	3.0	1

194 rows × 15 columns

scalling

In [28]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
df.iloc[:,:-1]=ss.fit_transform(df.iloc[:,:-1])
df
```

Out[28]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	
0	1.846173	0.082835	-1.934007	0.339032	-3.111634	1.234608	-0.125327	-0.820757	-2.
1	1.846173	0.082835	-1.934007	0.339032	-3.111634	1.234608	-0.125327	-0.820757	-2.
2	0.176441	0.082835	-1.934007	0.339032	-0.748984	1.234608	-0.125327	-0.179636	-0.
3	1.011307	1.349433	-1.774620	0.339032	0.432341	-0.566249	-0.125327	0.140924	0.
4	1.011307	1.349433	-1.774620	0.339032	0.432341	-2.367105	-0.125327	0.232512	0.
...	...	...	...	...	...	...	...	...	
200	-1.493292	-0.851218	1.413123	0.339032	0.432341	1.234608	-0.125327	1.377370	0.
201	-1.493292	-0.851218	1.413123	0.339032	0.432341	1.234608	-0.125327	1.331576	0.
202	-1.493292	-0.851218	1.413123	0.339032	0.432341	1.234608	-0.125327	1.377370	0.
203	-1.493292	-0.851218	1.413123	-2.949576	0.432341	1.234608	-0.125327	1.377370	0.
204	-1.493292	-0.851218	1.413123	0.339032	0.432341	1.234608	-0.125327	1.377370	0.

194 rows × 15 columns

In [ ]: