In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
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	eng
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	

In [4]:
```
df.shape
```

Out[4]:
```
(205, 15)
```

In [5]:
```
df.describe()
```

Out[5]:

	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 [6]:
```
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 [7]:
```
num_col=df.select_dtypes(include=['int','float']).columns
```

In [8]:
```
num_col
```

Out[8]:
```
Index(['symboling', 'width', 'height', 'engine-size', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')
```

In [9]:
```
len(num_col)
```

Out[9]:
```
7
```

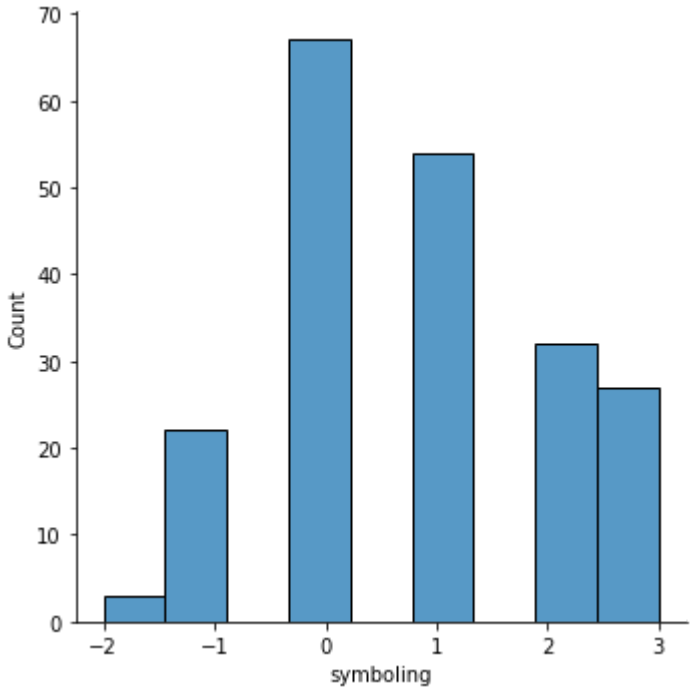
In [10]:
```
df.describe()
```

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

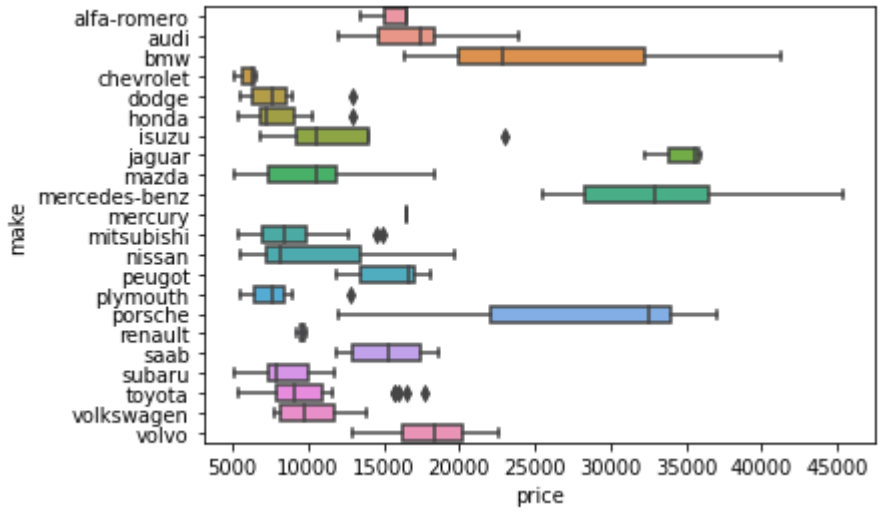
In [11]:
```
sns.displot(df['symboling'])
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x19bf7a4b610>
```



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

Out[12]:
```
<AxesSubplot:xlabel='price', ylabel='make'>
```



In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [14]:
```
df.make.unique()
```

Out[14]:
```
array(['alfa-romero', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
       'isuzu', 'jaguar', 'mazda', 'mercedes-benz', 'mercury',
       'mitsubishi', 'nissan', 'peugot', 'plymouth', 'porsche', 'renault',
       'saab', 'subaru', 'toyota', 'volkswagen', 'volvo'], dtype=object)
```

In [15]:
```
df.make.value_counts()
```

Out[15]:
```
toyota           32
nissan           18
mazda            17
mitsubishi       13
honda            13
volkswagen       12
subaru           12
peugot           11
volvo            11
dodge             9
mercedes-benz     8
bmw               8
audi              7
plymouth          7
saab              6
porsche           5
isuzu             4
jaguar            3
chevrolet         3
alfa-romero       3
renault           2
mercury           1
Name: make, dtype: int64
```

In [16]:
```
df[df['make']=="toyota"]
```

Out[16]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	e
150	1	87	toyota	gas	hatchback	fwd	front	63.6	54.5	ohc	
151	1	87	toyota	gas	hatchback	fwd	front	63.6	54.5	ohc	
152	1	74	toyota	gas	hatchback	fwd	front	63.6	54.5	ohc	
153	0	77	toyota	gas	wagon	fwd	front	63.6	59.1	ohc	
154	0	81	toyota	gas	wagon	4wd	front	63.6	59.1	ohc	
155	0	91	toyota	gas	wagon	4wd	front	63.6	59.1	ohc	
156	0	91	toyota	gas	sedan	fwd	front	64.4	53.0	ohc	
157	0	91	toyota	gas	hatchback	fwd	front	64.4	52.8	ohc	
158	0	91	toyota	diesel	sedan	fwd	front	64.4	53.0	ohc	
159	0	91	toyota	diesel	hatchback	fwd	front	64.4	52.8	ohc	
160	0	91	toyota	gas	sedan	fwd	front	64.4	53.0	ohc	
161	0	91	toyota	gas	hatchback	fwd	front	64.4	52.8	ohc	
162	0	91	toyota	gas	sedan	fwd	front	64.4	52.8	ohc	
163	1	168	toyota	gas	sedan	rwd	front	64.0	52.6	ohc	
164	1	168	toyota	gas	hatchback	rwd	front	64.0	52.6	ohc	
165	1	168	toyota	gas	sedan	rwd	front	64.0	52.6	dohc	
166	1	168	toyota	gas	hatchback	rwd	front	64.0	52.6	dohc	
167	2	134	toyota	gas	hardtop	rwd	front	65.6	52.0	ohc	
168	2	134	toyota	gas	hardtop	rwd	front	65.6	52.0	ohc	
169	2	134	toyota	gas	hatchback	rwd	front	65.6	52.0	ohc	
170	2	134	toyota	gas	hardtop	rwd	front	65.6	52.0	ohc	
171	2	134	toyota	gas	hatchback	rwd	front	65.6	52.0	ohc	
172	2	134	toyota	gas	convertible	rwd	front	65.6	53.0	ohc	
173	-1	65	toyota	gas	sedan	fwd	front	66.5	54.9	ohc	
174	-1	65	toyota	diesel	sedan	fwd	front	66.5	54.9	ohc	
175	-1	65	toyota	gas	hatchback	fwd	front	66.5	53.9	ohc	
176	-1	65	toyota	gas	sedan	fwd	front	66.5	54.9	ohc	
177	-1	65	toyota	gas	hatchback	fwd	front	66.5	53.9	ohc	
178	3	197	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
179	3	197	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
180	-1	90	toyota	gas	sedan	rwd	front	66.5	54.1	dohc	
181	-1	?	toyota	gas	wagon	rwd	front	66.5	54.1	dohc	

In [17]:
```
df[(df['make']=="toyota") & (df['price']>15000)]
```

Out[17]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	en
172	2	134	toyota	gas	convertible	rwd	front	65.6	53.0	ohc	
178	3	197	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
179	3	197	toyota	gas	hatchback	rwd	front	67.7	52.0	dohc	
180	-1	90	toyota	gas	sedan	rwd	front	66.5	54.1	dohc	
181	-1	?	toyota	gas	wagon	rwd	front	66.5	54.1	dohc	

In [25]:
```
df.drop(index=[172,178,179,180,181],inplace=True)
```

In [27]:
```
df.isnull().sum()
```

Out[27]:
```
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 [ ]:
```
maean=df['normalized-losses'].mean()
mean
```

In [19]:
```
df['make'].isnull().sum()
```

Out[19]:
```
0
```

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

Out[20]:

	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 [22]:
```
df['normalized-losses']=df['normalized-losses'].replace('?',np.nan)
```

In [25]:
```
df.dropna(inplace=True)
```

In [26]:
```
df['normalized-losses'].isnull().sum()
```

Out[26]:
```
0
```

In [27]:
```
cat_col=df.select_dtypes('object').columns
```

In [28]:
```
cat_col
```

Out[28]:
```
Index(['normalized-losses', 'make', 'fuel-type', 'body-style', 'drive-wheels',
       'engine-location', 'engine-type', 'horsepower'],
      dtype='object')
```

In [29]:
```
df['fuel-type'].unique()
```

Out[29]:
```
array(['gas', 'diesel'], dtype=object)
```

In [30]:
```
df['horsepower'].unique()
```

Out[30]:
```
array(['102', '115', '110', '140', '101', '121', '48', '70', '68', '88',
       '145', '58', '76', '60', '86', '100', '176', '135', '84', '120',
       '123', '155', '116', '69', '55', '97', '152', '160', '200', '95',
       '142', '143', '73', '82', '94', '111', '62', '56', '112', '92',
       '161', '156', '52', '85', '90', '114', '162', '134', '106'],
      dtype=object)
```

In [31]:
```
df['horsepower'].isnull().sum()
```

Out[31]:
```
0
```

In [34]:
```
df['horsepower']=df['horsepower'].replace("?",np.nan)
```

In [35]:
```
df['horsepower'].isnull().sum()
```

Out[35]:
```
0
```

In [36]:
```
pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\mainawati\appdata\local\prog
rams\python\python310\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\mainawati\appdata\local
\programs\python\python310\lib\site-packages (from sklearn) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\mainawati\appda
ta\local\programs\python\python310\lib\site-packages (from scikit-learn->sklear
n) (3.1.0)
Requirement already satisfied: joblib>=1.0.0 in c:\users\mainawati\appdata\loca
l\programs\python\python310\lib\site-packages (from scikit-learn->sklearn) (1.
1.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\mainawati\appdata\loca
l\programs\python\python310\lib\site-packages (from scikit-learn->sklearn) (1.2
3.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\mainawati\appdata\local
\programs\python\python310\lib\site-packages (from scikit-learn->sklearn) (1.8.
1)
Note: you may need to restart the kernel to use updated packages.

WARNING: You are using pip version 22.0.4; however, version 22.2.2 is availabl
e.
You should consider upgrading via the 'C:\Users\mainawati\AppData\Local\Program
s\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

In [37]:
```
df.columns
```

Out[37]:
```
Index(['symboling', 'normalized-losses', 'make', 'fuel-type', 'body-style',
       'drive-wheels', 'engine-location', 'width', 'height', 'engine-type',
       'engine-size', 'horsepower', 'city-mpg', 'highway-mpg', 'price'],
      dtype='object')
```

In [38]:
```
from sklearn.preprocessing import LabelEncoder
```

In [39]:
```
s=LabelEncoder()
```

In [42]:
```
df['fuel-type']=s.fit_transform(df['fuel-type'])
```

In [43]:
```
df['fuel-type']
```

Out[43]:
```
3      1
4      1
6      1
8      1
10     1
      ..
200    1
201    1
202    1
203    0
204    1
Name: fuel-type, Length: 164, dtype: int32
```

In [44]:
```
df['fuel-type'].unique()
```

Out[44]:
```
array([1, 0])
```

In [45]:
```
df.head()
```

Out[45]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size
3	2	164	audi	1	sedan	fwd	front	66.2	54.3	ohc	109
4	2	164	audi	1	sedan	4wd	front	66.4	54.3	ohc	136
6	1	158	audi	1	sedan	fwd	front	71.4	55.7	ohc	136
8	1	158	audi	1	sedan	fwd	front	71.4	55.9	ohc	131
10	2	192	bmw	1	sedan	rwd	front	64.8	54.3	ohc	108

In [46]:
```
cat_col
```

Out[46]:
```
Index(['normalized-losses', 'make', 'fuel-type', 'body-style', 'drive-wheels',
       'engine-location', 'engine-type', 'horsepower'],
      dtype='object')
```

In [47]:
```
num_col
```

Out[47]:
```
Index(['symboling', 'width', 'height', 'engine-size', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')
```

In [54]:
```
#s=LabelEncoder()
for col in cat_col:
    s=LabelEncoder()
    df[col]=s.fit_transform(df[col])
```

In [55]:
```
df.head()
```

Out[55]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size
3	2	27	0	1	3	1	0	66.2	54.3	2	109
4	2	27	0	1	3	0	0	66.4	54.3	2	136
6	1	25	0	1	3	1	0	71.4	55.7	2	136
8	1	25	0	1	3	1	0	71.4	55.9	2	131
10	2	31	1	1	3	2	0	64.8	54.3	2	108

In [56]:
```
num_col
```

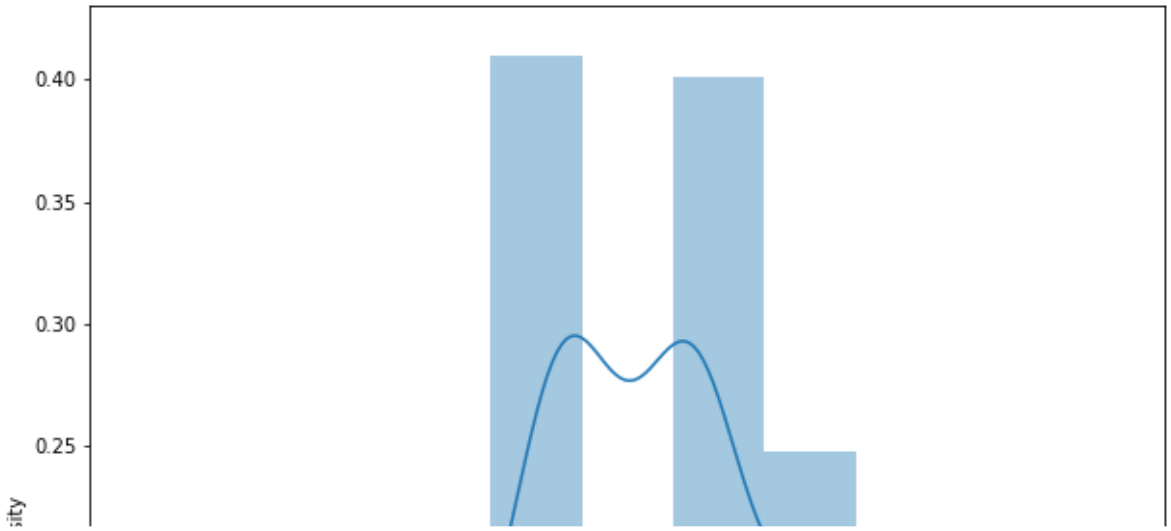
Out[56]:
```
Index(['symboling', 'width', 'height', 'engine-size', 'city-mpg',
       'highway-mpg', 'price'],
      dtype='object')
```

In [58]:
```
for col in num_col:
    print(col)
    plt.figure(figsize=(10,10))
    sns.distplot(df[col])
    plt.show()
```

```
symboling
```

In [63]:
```
for col in num_col:
    print(col)
    print('skewness',df[col].skew())
    print('kurtosis',df[col].kurt())
    plt.figure(figsize=(10,10))
    sns.distplot(df[col])
    plt.show()
```

```
symboling
skewness 0.10049657227421199
kurtosis -0.5947812272761666
```



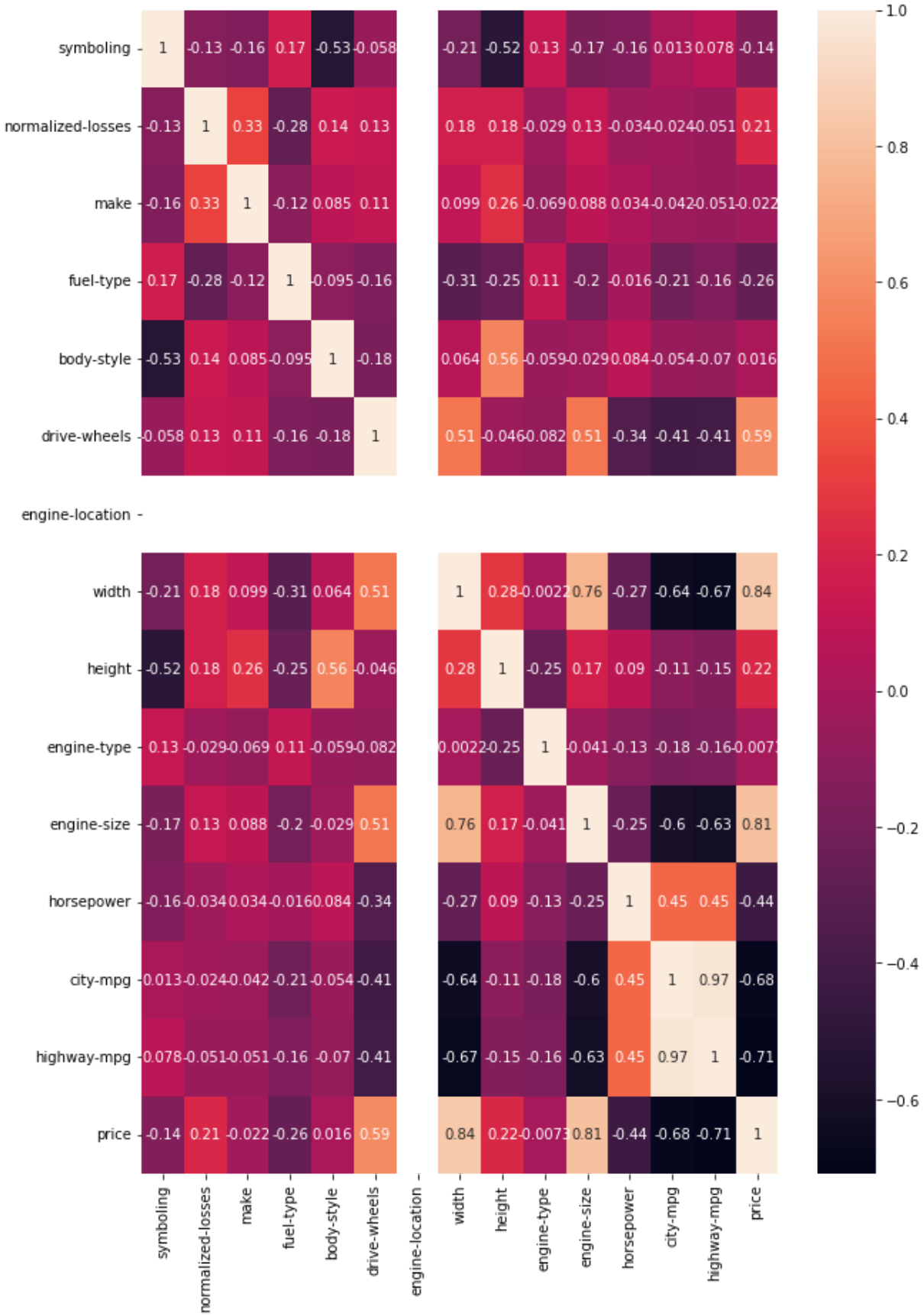
In [64]:
```
corr=df.corr()
```

In [65]:
```
corr
```

Out[65]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width
symboling	1.000000	-0.132364	-0.155640	0.171179	-0.529376	-0.058496	NaN	-0.208303
normalized-losses	-0.132364	1.000000	0.328547	-0.277971	0.142251	0.128753	NaN	0.177806
make	-0.155640	0.328547	1.000000	-0.116861	0.084697	0.105792	NaN	0.098805
fuel-type	0.171179	-0.277971	-0.116861	1.000000	-0.095006	-0.163889	NaN	-0.312443
body-style	-0.529376	0.142251	0.084697	-0.095006	1.000000	-0.178639	NaN	0.063630
drive-wheels	-0.058496	0.128753	0.105792	-0.163889	-0.178639	1.000000	NaN	0.512294
engine-location	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
width	-0.208303	0.177806	0.098805	-0.312443	0.063630	0.512294	NaN	1.000000
height	-0.516420	0.178331	0.256464	-0.250596	0.562439	-0.046305	NaN	0.283885
engine-type	0.125153	-0.029012	-0.068594	0.111097	-0.059247	-0.081515	NaN	-0.002173
engine-size	-0.169342	0.125227	0.087860	-0.197530	-0.028671	0.506697	NaN	0.756317
horsepower	-0.155215	-0.034188	0.034086	-0.016486	0.084412	-0.342580	NaN	-0.274705
city-mpg	0.013028	-0.023534	-0.042058	-0.212318	-0.053769	-0.410807	NaN	-0.644124
highway-mpg	0.078220	-0.051395	-0.050584	-0.157270	-0.069524	-0.408630	NaN	-0.674959
price	-0.144078	0.214196	-0.022170	-0.259010	0.016015	0.591703	NaN	0.841883

In [68]:
```
plt.figure(figsize=(10,15))
sns.heatmap(corr,annot=True)
plt.show()
```



In [ ]: