EDA—cars dataset

January 31, 2023

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

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

[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
```

[4]:
```
df.head()
```

[4]:

	symboling	normalized-losses	make	fuel-type	body-style	\
0	3	?	alfa-romero	gas	convertible	

```
1          3                 ?  alfa-romero       gas  convertible
2          1                 ?  alfa-romero       gas    hatchback
3          2               164         audi       gas        sedan
4          2               164         audi       gas        sedan

  drive-wheels engine-location  width  height engine-type  engine-size  \
0          rwd           front   64.1    48.8        dohc          130
1          rwd           front   64.1    48.8        dohc          130
2          rwd           front   65.5    52.4        ohcv          152
3          fwd           front   66.2    54.3         ohc          109
4          4wd           front   66.4    54.3         ohc          136

  horsepower  city-mpg  highway-mpg  price
0        111        21           27  13495
1        111        21           27  16500
2        154        19           26  16500
3        102        24           30  13950
4        115        18           22  17450
```

[5]:
```
feature=df.iloc[:,:-1]
```

[6]:
```
target=df.iloc[:,-1]
```

[7]:
```
feature
```

[7]:
```
     symboling normalized-losses         make fuel-type   body-style  \
0            3                 ?  alfa-romero       gas  convertible
1            3                 ?  alfa-romero       gas  convertible
2            1                 ?  alfa-romero       gas    hatchback
3            2               164         audi       gas        sedan
4            2               164         audi       gas        sedan
..         …                 …           …        …           …
200         -1                95        volvo       gas        sedan
201         -1                95        volvo       gas        sedan
202         -1                95        volvo       gas        sedan
203         -1                95        volvo    diesel        sedan
204         -1                95        volvo       gas        sedan

    drive-wheels engine-location  width  height engine-type  engine-size  \
0            rwd           front   64.1    48.8        dohc          130
1            rwd           front   64.1    48.8        dohc          130
2            rwd           front   65.5    52.4        ohcv          152
3            fwd           front   66.2    54.3         ohc          109
4            4wd           front   66.4    54.3         ohc          136
..           …             …     …     …          …           …
200          rwd           front   68.9    55.5         ohc          141
201          rwd           front   68.8    55.5         ohc          141
```

```
202          rwd           front   68.9    55.5        ohcv          173
203          rwd           front   68.9    55.5         ohc          145
204          rwd           front   68.9    55.5         ohc          141

    horsepower  city-mpg  highway-mpg
0          111        21           27
1          111        21           27
2          154        19           26
3          102        24           30
4          115        18           22
..         …       …          …
200        114        23           28
201        160        19           25
202        134        18           23
203        106        26           27
204        114        19           25

[205 rows x 14 columns]
```

[8]:
```
target
```

[8]:
```
0      13495
1      16500
2      16500
3      13950
4      17450
       …
200    16845
201    19045
202    21485
203    22470
204    22625
Name: price, Length: 205, dtype: int64
```

[9]:
```
feature['normalized-losses'].unique()
```

[9]:
```
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)
```

[10]:
```
df['normalized-losses'].replace('?',np.nan,inplace=True)
```

[11]:
```
df['normalized-losses'].unique()
```

[11]:
```
array([nan, '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)
```

[12]:
```
df['normalized-losses'].astype('float64')
```

[12]:
```
0        NaN
1        NaN
2        NaN
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: 205, dtype: float64
```

[13]:
```
df['horsepower'].unique()
```

[13]:
```
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)
```

[14]:
```
df['horsepower'].value_counts()
```

[14]:
```
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
```

[15]:
```
df['horsepower'].replace('?',np.nan,inplace=True)
```

[16]:
```
df['horsepower'].astype('float64')
```

[16]:
```
0      111.0
1      111.0
2      154.0
3      102.0
4      115.0
       ...
200    114.0
201    160.0
202    134.0
203    106.0
204    114.0
Name: horsepower, Length: 205, dtype: float64
```

[17]:
```
from sklearn.impute import SimpleImputer
```

[18]:
```
si=SimpleImputer(missing_values=np.nan,strategy='mean')
```

[19]:
```
df[['horsepower','normalized-losses']]=si.
 ↪fit_transform(df[['horsepower','normalized-losses']])
```

[20]:
```
df.head()
```

[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	111.0	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

[21]:
```
df['horsepower'].unique()
```

[21]:
```
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.        ,
       104.25615764,  73.        ,  82.        ,  94.        ,
        62.        ,  56.        , 112.        ,  92.        ,
       161.        , 156.        ,  52.        ,  85.        ,
       114.        , 162.        , 134.        , 106.        ])
```

[22]:
```
feature
```

[22]:

	symboling	normalized-losses	make	fuel-type	body-style	\
0	3	?	alfa-romero	gas	convertible	
1	3	?	alfa-romero	gas	convertible	
2	1	?	alfa-romero	gas	hatchback	
3	2	164	audi	gas	sedan	
4	2	164	audi	gas	sedan	
..	…	…	…	…	…	
200	-1	95	volvo	gas	sedan	
201	-1	95	volvo	gas	sedan	
202	-1	95	volvo	gas	sedan	
203	-1	95	volvo	diesel	sedan	
204	-1	95	volvo	gas	sedan	

	drive-wheels	engine-location	width	height	engine-type	engine-size	\
0	rwd	front	64.1	48.8	dohc	130	
1	rwd	front	64.1	48.8	dohc	130	
2	rwd	front	65.5	52.4	ohcv	152	
3	fwd	front	66.2	54.3	ohc	109	
4	4wd	front	66.4	54.3	ohc	136	
..	…	…	…	…	…	…	
200	rwd	front	68.9	55.5	ohc	141	
201	rwd	front	68.8	55.5	ohc	141	
202	rwd	front	68.9	55.5	ohcv	173	
203	rwd	front	68.9	55.5	ohc	145	
204	rwd	front	68.9	55.5	ohc	141	

	horsepower	city-mpg	highway-mpg
0	111	21	27
1	111	21	27

```
2          154         19            26
3          102         24            30
4          115         18            22
..         …       …           …
200        114         23            28
201        160         19            25
202        134         18            23
203        106         26            27
204        114         19            25

[205 rows x 14 columns]
```

[23]:
```
target
```

[23]:
```
0      13495
1      16500
2      16500
3      13950
4      17450
       …
200    16845
201    19045
202    21485
203    22470
204    22625
Name: price, Length: 205, dtype: int64
```

[24]:
```
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         205 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
```

[25]:
```
colname=feature.select_dtypes(['int64','float64']).columns
```

[26]:
```
colname
```

[26]:
```
Index(['symboling', 'width', 'height', 'engine-size', 'city-mpg',
       'highway-mpg'],
      dtype='object')
```

[27]:
```
feature[colname]
```

[27]:

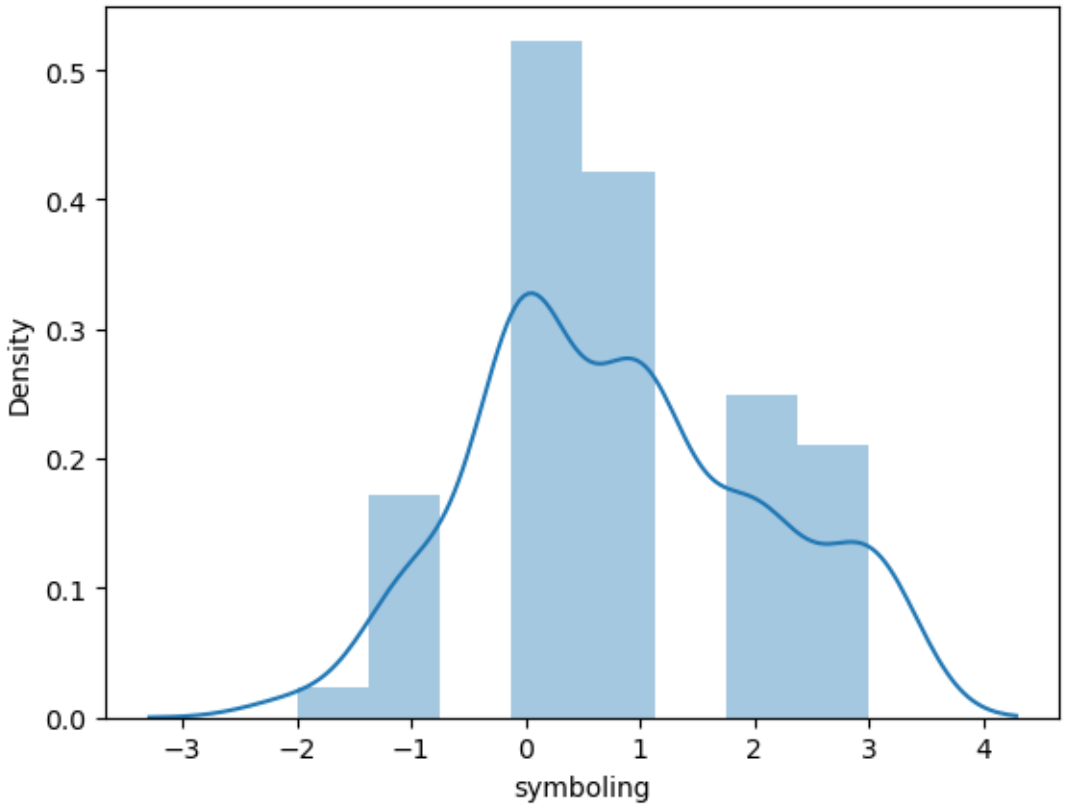
	symboling	width	height	engine-size	city-mpg	highway-mpg
0	3	64.1	48.8	130	21	27
1	3	64.1	48.8	130	21	27
2	1	65.5	52.4	152	19	26
3	2	66.2	54.3	109	24	30
4	2	66.4	54.3	136	18	22
..	…	…	…	…	…	…
200	-1	68.9	55.5	141	23	28
201	-1	68.8	55.5	141	19	25
202	-1	68.9	55.5	173	18	23
203	-1	68.9	55.5	145	26	27
204	-1	68.9	55.5	141	19	25

[205 rows x 6 columns]

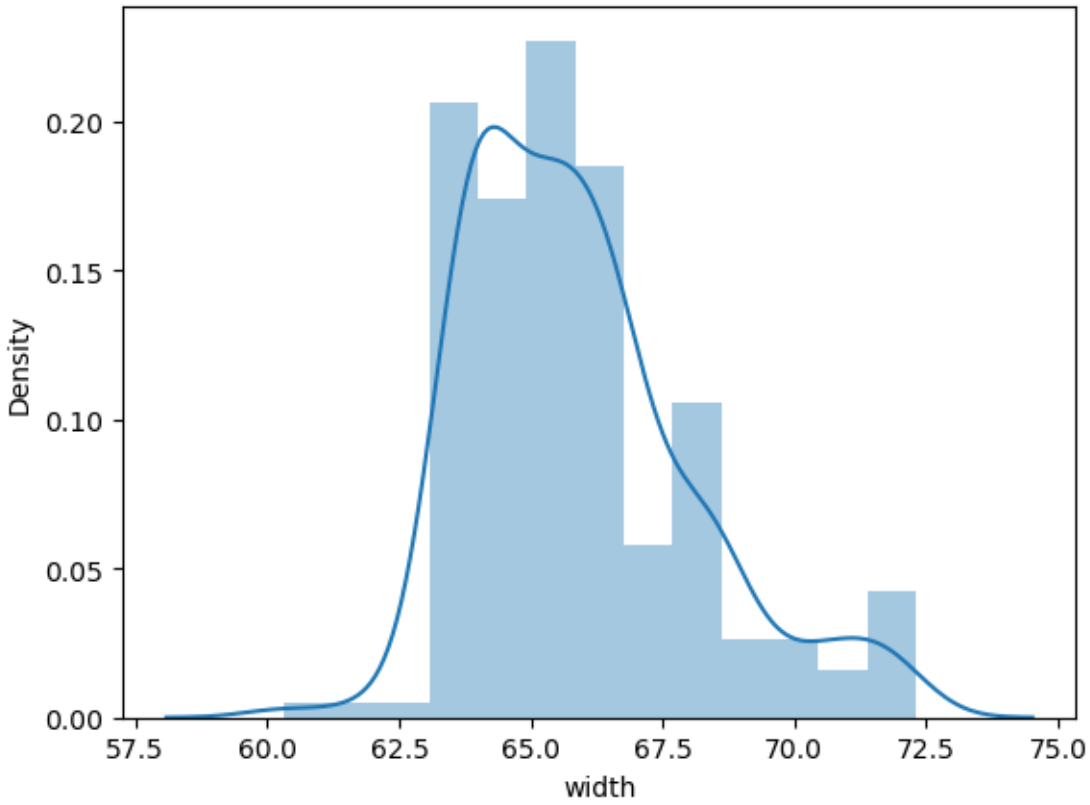
[28]:
```
from scipy.stats import skew
```

[29]:
```
for i in feature[colname]:
    print(i)
    print(skew(feature[i]))
    sns.distplot(feature[i])
    plt.show()
```

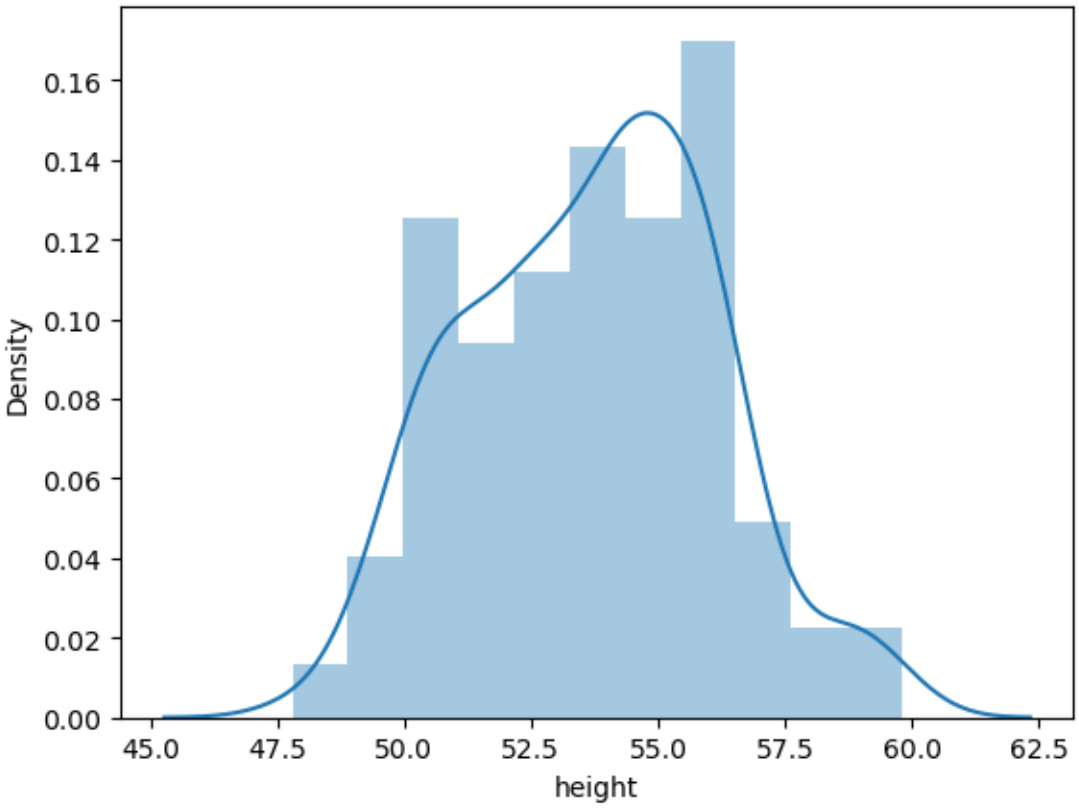
```
symboling
0.20952469094997359
```



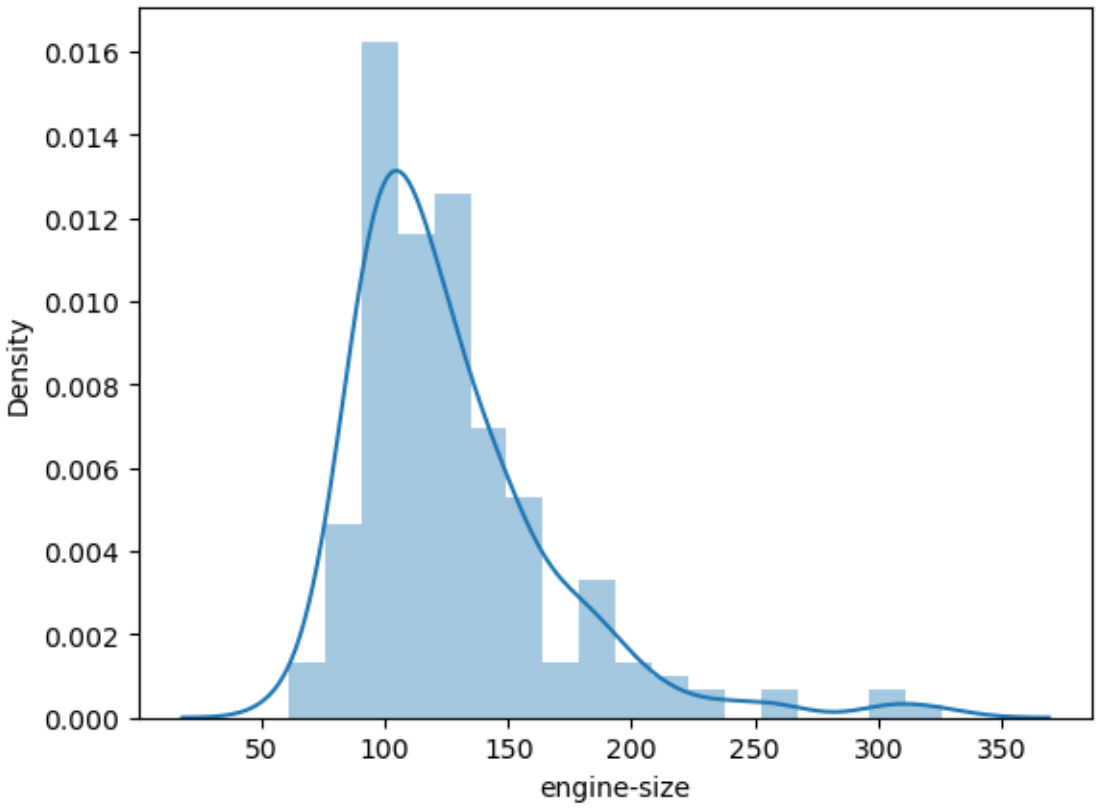

```
width
0.8973753485201392
```



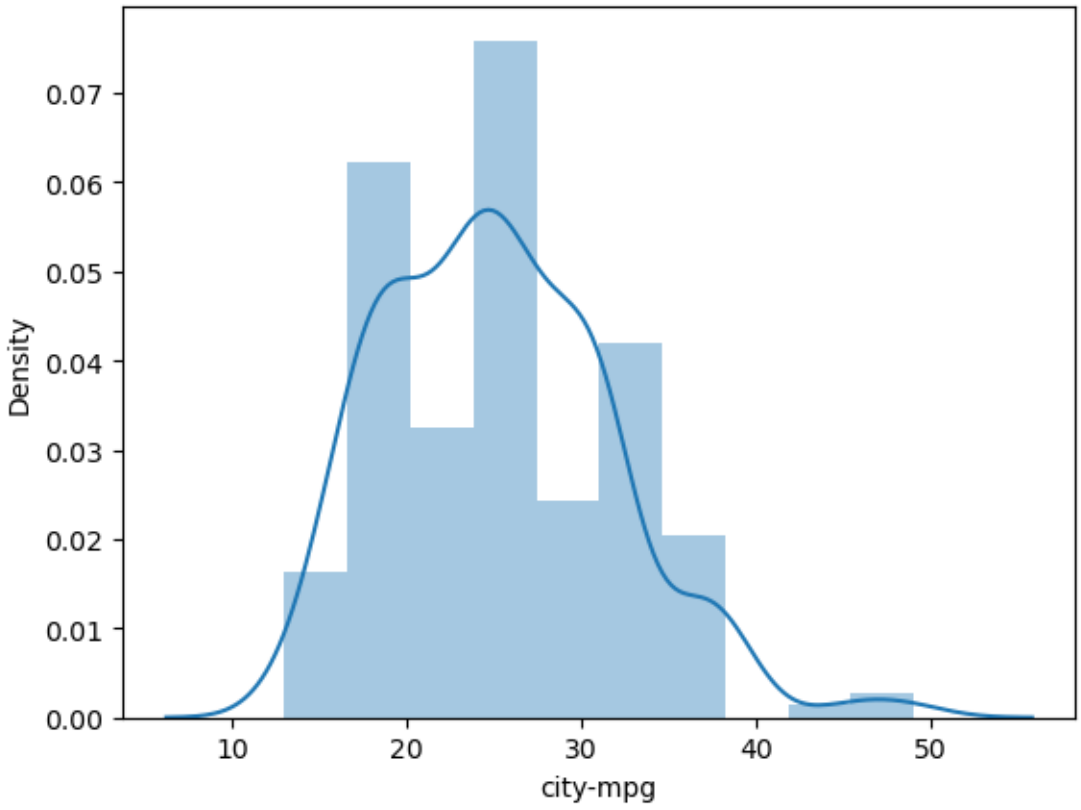

```
height
0.06265991683394276
```



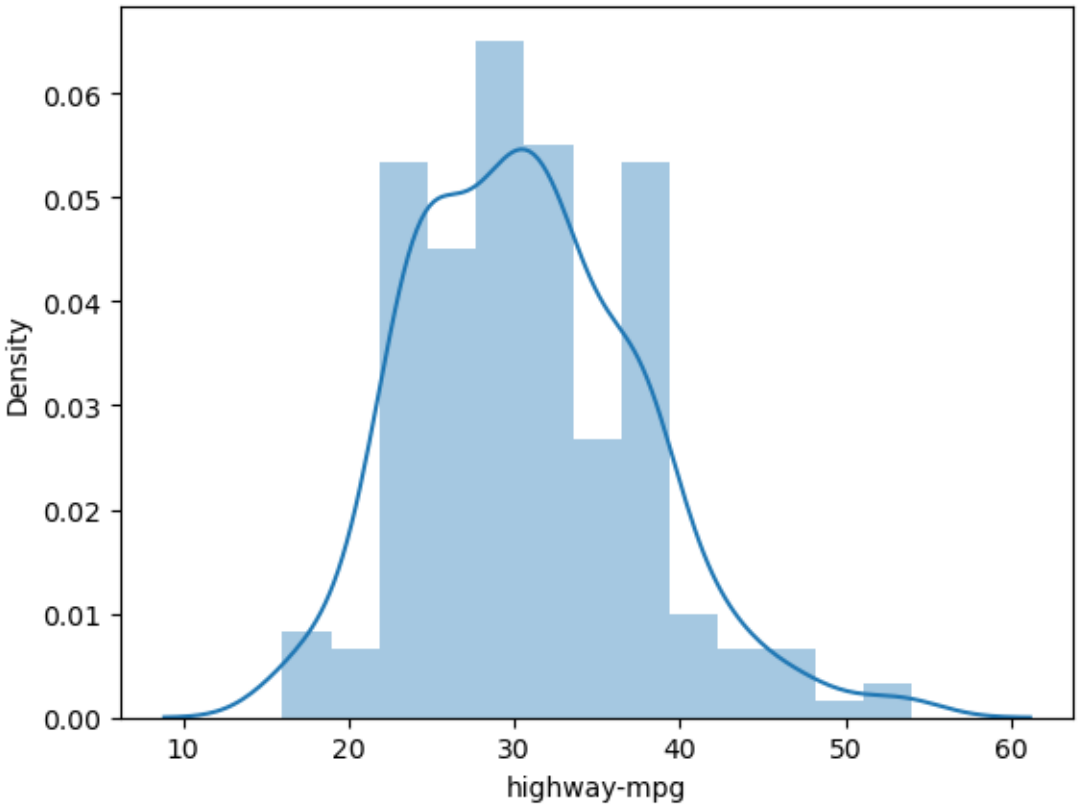

```
engine-size
1.9333748457840114
```




```
city-mpg
0.6588377533622138
```




```
highway-mpg
0.5360379305163596
```



[30]:
```
pd.concat([feature,target],axis=1).corr().style.background_gradient()
```

[30]:
```
<pandas.io.formats.style.Styler at 0x1e5e5f76d30>
```

[31]:
```
feature['height'].unique()
```

[31]:
```
array([48.8, 52.4, 54.3, 53.1, 55.7, 55.9, 52. , 53.7, 56.3, 53.2, 50.8,
       50.6, 59.8, 50.2, 52.6, 54.5, 58.3, 53.3, 54.1, 51. , 53.5, 51.4,
       52.8, 47.8, 49.6, 55.5, 54.4, 56.5, 58.7, 54.9, 56.7, 55.4, 54.8,
       49.4, 51.6, 54.7, 55.1, 56.1, 49.7, 56. , 50.5, 55.2, 52.5, 53. ,
       59.1, 53.9, 55.6, 56.2, 57.5])
```

[32]:
```
feature['normalized-losses']=np.log(feature['normalized-losses'])
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
AttributeError: 'str' object has no attribute 'log'

The above exception was the direct cause of the following exception:

TypeError                                 Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_11564\1058352373.py in <module>
----> 1 feature['normalized-losses']=np.log(feature['normalized-losses'])

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in␣
 ↪__array_ufunc__(self, ufunc, method, *inputs, **kwargs)
   2099         self, ufunc: np.ufunc, method: str, *inputs: Any, **kwargs: Any
   2100     ):
-> 2101         return arraylike.array_ufunc(self, ufunc, method, *inputs,␣
 ↪**kwargs)
   2102
   2103     #␣
 ↪------------------------------------------------------------------------

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\arraylike.py in␣
 ↪array_ufunc(self, ufunc, method, *inputs, **kwargs)
    395         # ufunc(series, …)
    396         inputs = tuple(extract_array(x, extract_numpy=True) for x in␣
 ↪inputs)
--> 397         result = getattr(ufunc, method)(*inputs, **kwargs)
    398     else:
    399         # ufunc(dataframe)

TypeError: loop of ufunc does not support argument 0 of type str which has no␣
 ↪callable log method
```

[ ]:
```
for i in feature[colname]:
    print(i)
    print(skew(feature[i]))
    sns.distplot(feature[i])
    plt.show()
```

[33]:
```
df.head()
```

[33]:

	symboling	normalized-losses	make	fuel-type	body-style	drive-wheels	engine-location	width	height	engine-type	engine-size
0	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130
1	3	122.0	alfa-romero	gas	convertible	rwd	front	64.1	48.8	dohc	130
2	1	122.0	alfa-romero	gas	hatchback	rwd	front	65.5	52.4	ohcv	152
3	2	164.0	audi	gas	sedan	fwd	front	66.2	54.3	ohc	109
4	2	164.0	audi	gas	sedan	4wd	front	66.4	54.3	ohc	136

```
   horsepower  city-mpg  highway-mpg  price
0       111.0        21           27  13495
1       111.0        21           27  16500
2       154.0        19           26  16500
3       102.0        24           30  13950
4       115.0        18           22  17450
```

[37]:
```
from sklearn.preprocessing import OrdinalEncoder  #encoding the categorical␣
↪value
```

[38]:
```
oe=OrdinalEncoder()
```

[34]:
```
catcol=df.select_dtypes('object').columns
```

[36]:
```
catcol
```

[36]:
```
Index(['make', 'fuel-type', 'body-style', 'drive-wheels', 'engine-location',
       'engine-type'],
      dtype='object')
```

[41]:
```
df[catcol]=oe.fit_transform(df[catcol])
```

[42]:
```
df
```

[42]:
```
     symboling  normalized-losses  make  fuel-type  body-style  drive-wheels  \
0            3              122.0   0.0        1.0         0.0           2.0
1            3              122.0   0.0        1.0         0.0           2.0
2            1              122.0   0.0        1.0         2.0           2.0
3            2              164.0   1.0        1.0         3.0           1.0
4            2              164.0   1.0        1.0         3.0           0.0
..         ...                ...   ...        ...         ...           ...
200         -1               95.0  21.0        1.0         3.0           2.0
201         -1               95.0  21.0        1.0         3.0           2.0
202         -1               95.0  21.0        1.0         3.0           2.0
203         -1               95.0  21.0        0.0         3.0           2.0
204         -1               95.0  21.0        1.0         3.0           2.0

     engine-location  width  height  engine-type  engine-size  horsepower  \
0                0.0   64.1    48.8          0.0          130       111.0
1                0.0   64.1    48.8          0.0          130       111.0
2                0.0   65.5    52.4          5.0          152       154.0
3                0.0   66.2    54.3          3.0          109       102.0
4                0.0   66.4    54.3          3.0          136       115.0
..               ...    ...     ...          ...          ...         ...
200              0.0   68.9    55.5          3.0          141       114.0
201              0.0   68.8    55.5          3.0          141       160.0
202              0.0   68.9    55.5          5.0          173       134.0
```

```
203              0.0   68.9    55.5          3.0          145       106.0
204              0.0   68.9    55.5          3.0          141       114.0

     city-mpg  highway-mpg  price
0          21           27  13495
1          21           27  16500
2          19           26  16500
3          24           30  13950
4          18           22  17450
..        …            …     …
200        23           28  16845
201        19           25  19045
202        18           23  21485
203        26           27  22470
204        19           25  22625

[205 rows x 15 columns]
```

[43]:
```
x=df.iloc[:,:-1]
y=df.iloc[:,-1]
```

[44]:
```
x
```

[44]:
```
     symboling  normalized-losses  make  fuel-type  body-style  drive-wheels  \
0            3              122.0   0.0        1.0         0.0           2.0
1            3              122.0   0.0        1.0         0.0           2.0
2            1              122.0   0.0        1.0         2.0           2.0
3            2              164.0   1.0        1.0         3.0           1.0
4            2              164.0   1.0        1.0         3.0           0.0
..         …                …     …          …           …             …
200         -1               95.0  21.0        1.0         3.0           2.0
201         -1               95.0  21.0        1.0         3.0           2.0
202         -1               95.0  21.0        1.0         3.0           2.0
203         -1               95.0  21.0        0.0         3.0           2.0
204         -1               95.0  21.0        1.0         3.0           2.0

     engine-location  width  height  engine-type  engine-size  horsepower  \
0                0.0   64.1    48.8          0.0          130       111.0
1                0.0   64.1    48.8          0.0          130       111.0
2                0.0   65.5    52.4          5.0          152       154.0
3                0.0   66.2    54.3          3.0          109       102.0
4                0.0   66.4    54.3          3.0          136       115.0
..               …     …      …             …            …          …
200              0.0   68.9    55.5          3.0          141       114.0
201              0.0   68.8    55.5          3.0          141       160.0
202              0.0   68.9    55.5          5.0          173       134.0
203              0.0   68.9    55.5          3.0          145       106.0
```

```
204              0.0   68.9    55.5          3.0          141       114.0

     city-mpg  highway-mpg
0          21           27
1          21           27
2          19           26
3          24           30
4          18           22
..        …           …
200        23           28
201        19           25
202        18           23
203        26           27
204        19           25

[205 rows x 14 columns]
```

[45]:
```
y
```

[45]:
```
0      13495
1      16500
2      16500
3      13950
4      17450
       …
200    16845
201    19045
202    21485
203    22470
204    22625
Name: price, Length: 205, dtype: int64
```

[46]:
```
from sklearn.model_selection import train_test_split
```

[47]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

[48]:
```
from sklearn.linear_model import LinearRegression
```

[49]:
```
linreg=LinearRegression()
```

[50]:
```
linreg.fit(xtrain,ytrain)
```

[50]:
```
LinearRegression()
```

[51]:
```
ypred=linreg.predict(xtest)
```

[52]:
```
train=linreg.score(xtrain,ytrain)
test=linreg.score(xtest,ytest)
```

[54]:
```
print(train)
print(test)
```

```
0.8504573774895472
0.7965566780397383
```

[57]:
```
linreg.coef_    #m value
```

[57]:
```
array([ 4.51384957e+01,  1.53127607e+00, -2.00099087e+02, -6.22650015e+02,
       -1.70235175e+02,  1.86860719e+03,  1.64133620e+04,  7.89452171e+02,
        3.62663990e+02,  2.83174279e+02,  9.83682875e+01, -1.08169245e+01,
        3.08017854e+02, -4.17024371e+02])
```

[71]:
```
from sklearn.linear_model import Ridge,Lasso
```

[95]:
```
l2=Ridge(alpha=10)
l2.fit(xtrain,ytrain)
ypred=l2.predict(xtest)
```

[96]:
```
train=l2.score(xtrain,ytrain)
test=l2.score(xtest,ytest)
print(train)
print(test)
```

```
0.8109538582620313
0.8150222867376528
```

[92]:
```
l2.coef_
```

[92]:
```
array([ 9.69712444e+00,  1.91441371e+00, -1.75157314e+02, -0.00000000e+00,
       -3.45177754e+02,  1.31718501e+03,  5.73225082e+03,  4.50558895e+02,
        4.43568844e+02,  2.98102606e+02,  1.11472445e+02,  7.57418730e+00,
        1.09925001e+02, -2.00008987e+02])
```

[88]:
```
l1=Lasso(alpha=150)
l1.fit(xtrain,ytrain)
ypred=l1.predict(xtest)
```

[89]:
```
train=l1.score(xtrain,ytrain)
test=l1.score(xtest,ytest)
print(train)
print(test)
```

```
0.8212288879856346
0.8124969899539802
```

[91]:
```
l1.coef_
```

[91]:
```
array([ 9.69712444e+00,  1.91441371e+00, -1.75157314e+02, -0.00000000e+00,
       -3.45177754e+02,  1.31718501e+03,  5.73225082e+03,  4.50558895e+02,
        4.43568844e+02,  2.98102606e+02,  1.11472445e+02,  7.57418730e+00,
        1.09925001e+02, -2.00008987e+02])
```

1 hyperparameter tuning—-for finding alpha value

[ ]:
```
for i in range(150,200):
    l1=lasso(alpha=i)
    l1.fit(xtrain,ytrain)
```