```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/MPG.csv')
df.head()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	or
0	18.0	8	307.0	130.0	3504	12.0	70	
1	15.0	8	350.0	165.0	3693	11.5	70	

```
df.duplicated().any()
```

```
False
```

```
df.nunique()
```

```
mpg             129
cylinders         5
displacement     82
horsepower       93
weight          351
acceleration     95
model_year       13
origin            3
name            305
dtype: int64
```

```
df.shape
```

```
(398, 9)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    object
 8   name          398 non-null    object
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

```
df.describe()
```

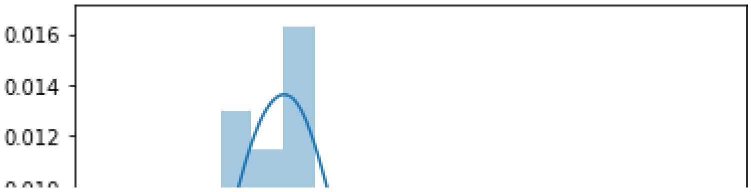
	mpg	cylinders	displacement	horsepower	weight	acceleration
count	398.000000	398.000000	398.000000	392.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	104.469388	2970.424623	15.568090
std	7.815984	1.701004	104.269838	38.491160	846.841774	2.757689
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000
25%	17.500000	4.000000	104.250000	75.000000	2223.750000	13.825000
50%	23.000000	4.000000	148.500000	93.500000	2803.500000	15.500000
75%	29.000000	8.000000	262.000000	126.000000	3608.000000	17.175000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000

```
df.corr()
```

	mpg	cylinders	displacement	horsepower	weight	accelerati
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.42028
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.50541
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.54368
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.68919
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.41745
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.00000
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.28813

```
sns.distplot(df.horsepower)
```

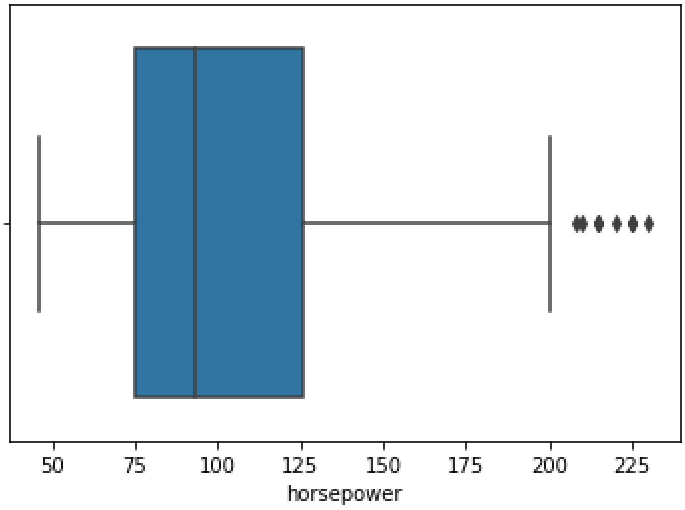
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarnin
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be433b4d0>
```



distribution is right skewed

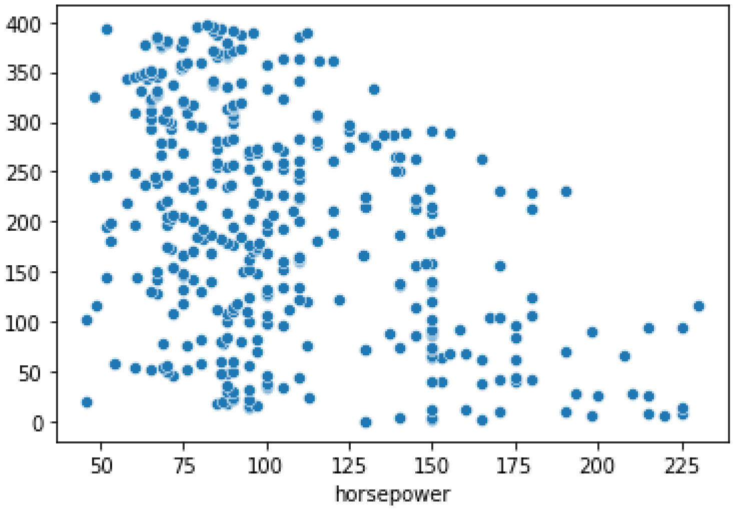
```
sns.boxplot(data = df, x='horsepower')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be4db6a50>
```



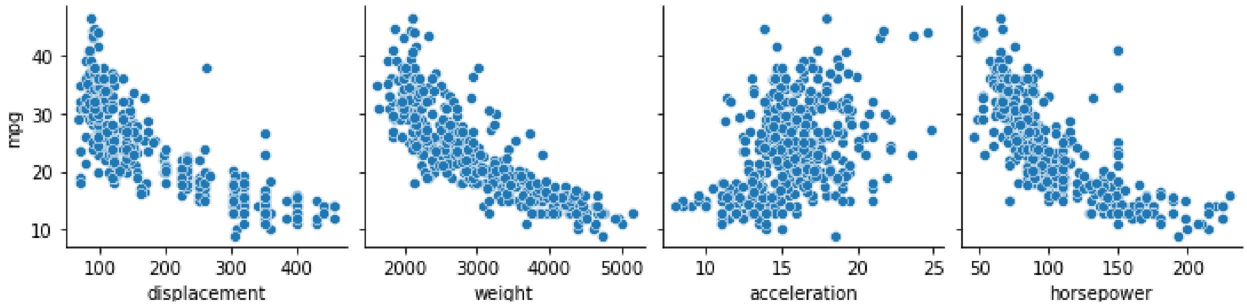
```
sns.scatterplot(data=df,x='horsepower',y=df.index)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be43a3450>
```



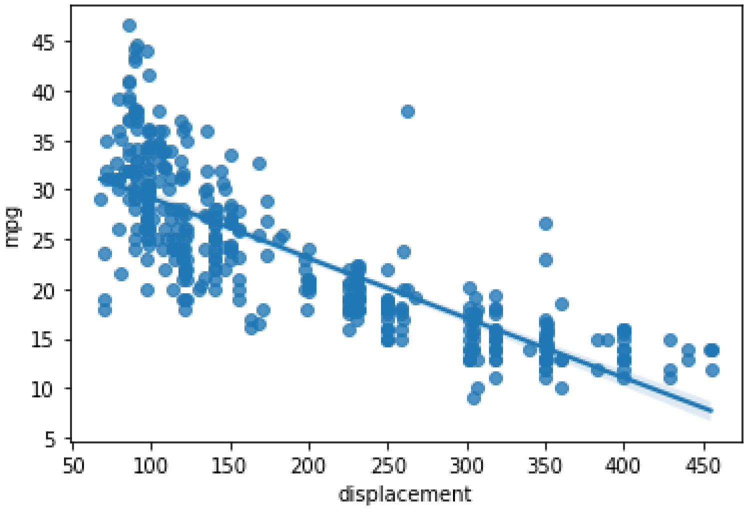
```
df.fillna(df['horsepower'].mode()[0],inplace=True)
```

```
sns.pairplot(df,x_vars = ['displacement','weight','acceleration','horsepower'],y_vars=['mp
<seaborn.axisgrid.PairGrid at 0x7f0be4215b10>
```



```
sns.regplot(y='mpg',x='displacement',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be4379e10>
```



```
y=df['mpg']
```

```
x=df[['displacement','weight','acceleration','horsepower']]
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state =42)
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
sc.fit_transform(xtrain)
sc.transform(xtest)[:5]
```

```
array([[-0.98134964, -1.39881183,  0.63795339, -1.36193777],
       [-0.69930815, -0.40988656,  1.07290607, -0.66787333],
       [ 0.38995555, -0.39916327, -0.9568731 , -0.10728282],
       [ 1.22635446,  1.15690469, -0.88438099,  1.22745649],
       [ 1.22635446,  1.51077313, -0.41318225,  1.22745649]])
```

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(xtrain,ytrain)
lr.coef_, lr.intercept_
```

```
     (array([-0.01095356, -0.00555501,  0.04002192, -0.0258002 ]),
      44.26089655132871)
```

```
lr.score(xtrain,ytrain)
```

```
     0.6969811459861376
```

```
lr.score(xtest,ytest)
```

```
     0.727296531264819
```

```
ypred = lr.predict(xtest)
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error

mean_absolute_error(ytest,ypred), mean_absolute_percentage_error(ytest,ypred)
```

```
     (3.0988328630775333, 0.14220433613678615)
```

polynomial regression

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=2,interaction_only=True,include_bias=True)
xtrain2 = poly.fit_transform(xtrain)
xtest2 = poly.fit_transform(xtest)
lr.fit(xtrain2,ytrain)
lr.score(xtrain2,ytrain),lr.score(xtest2,ytest)
```

```
     (0.7365728587808285, 0.7833181141485053)
```

```
ypred2 = lr.predict(xtest2)
from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error

mean_absolute_error(ytest,ypred2), mean_absolute_percentage_error(ytest,ypred2)
```

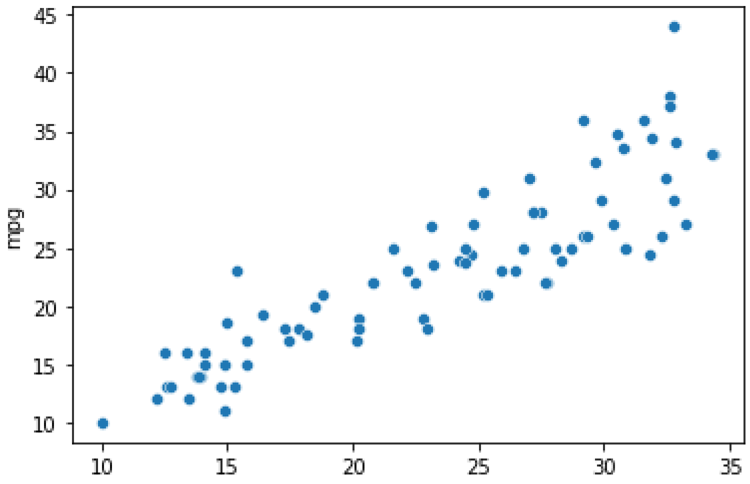
```
     (2.630835319242593, 0.11148450349648739)
```

```
sns.scatterplot(x = ypred,y=ytest)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be4efe110>
```

```
sns.scatterplot(x = ypred2,y=ytest)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0be400a8d0>
```



✓ 0s completed at 11:06 AM