1. Load the dataset into the tool

#importing the libraries

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

#loading the dataset

```
d = pd.read_csv(r'C:/Users/smiwin/OneDrive/Desktop/IBM
Datasets/abalone.csv')
```

2. Perform Below Visualizations

·Univariate Analysis

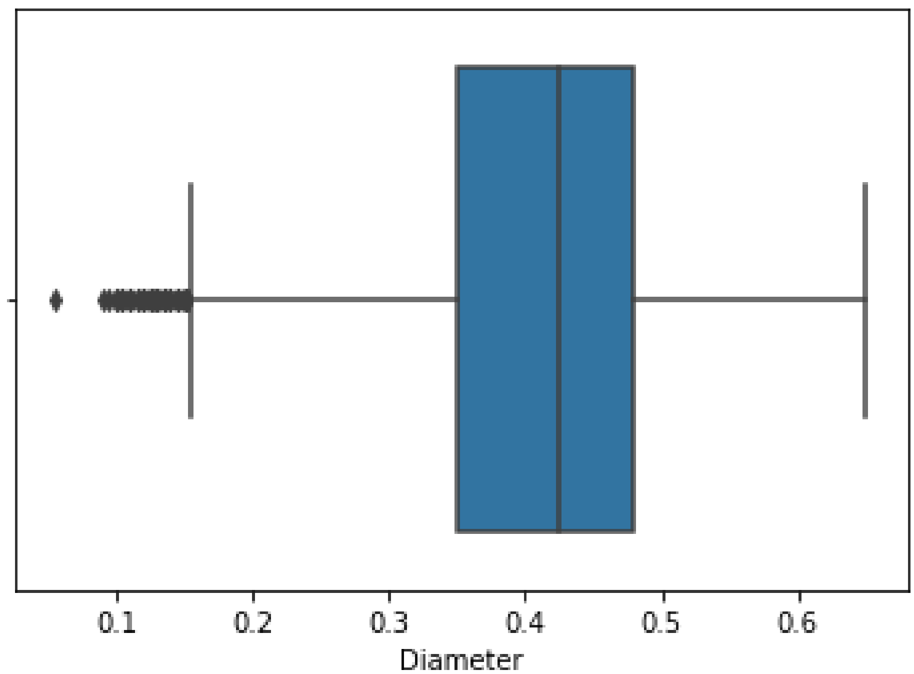
```
d.head()
```

```
 Sex Length Diameter Height Whole weight Shucked weight Viscera weight \
0 M 0.455 0.365 0.095 0.5140 0.2245  0.1010
1 M 0.350 0.265 0.090 0.2255 0.0995  0.0485
2 F 0.530 0.420 0.135 0.6770 0.2565  0.1415
3 M 0.440 0.365 0.125 0.5160 0.2155  0.1140
4 I 0.330 0.255 0.080 0.2050 0.0895  0.0395

 Shell weight Rings
0 0.150 15
1 0.070 7
2 0.210 9
3 0.155 10
4 0.055 7
```

#Boxplot

```
sns.boxplot(d['Diameter'])
```

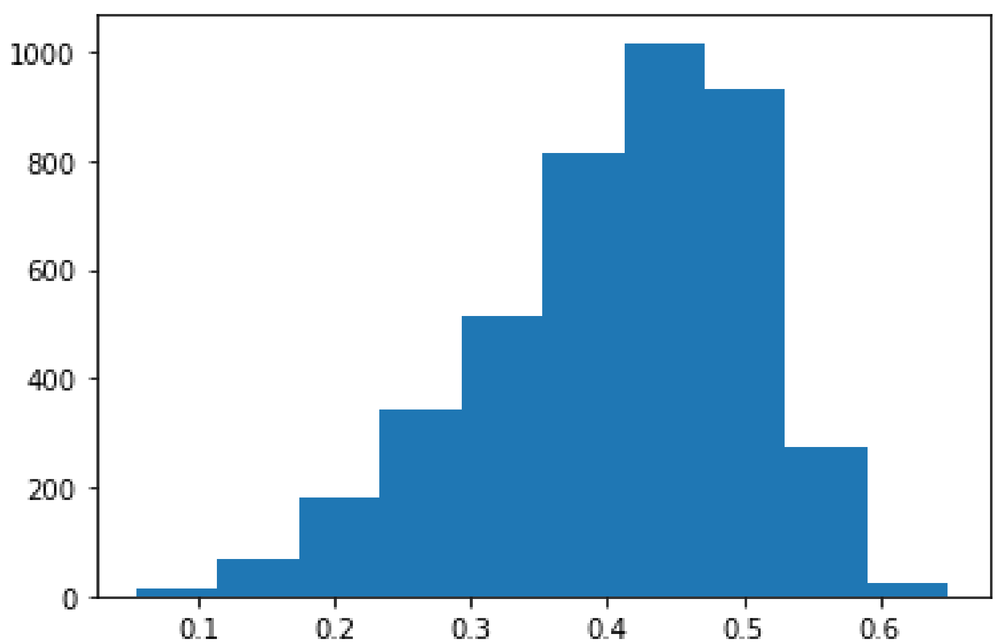
```
<AxesSubplot:xlabel='Diameter'>
```



#histogram

plt.hist(d['Diameter'])

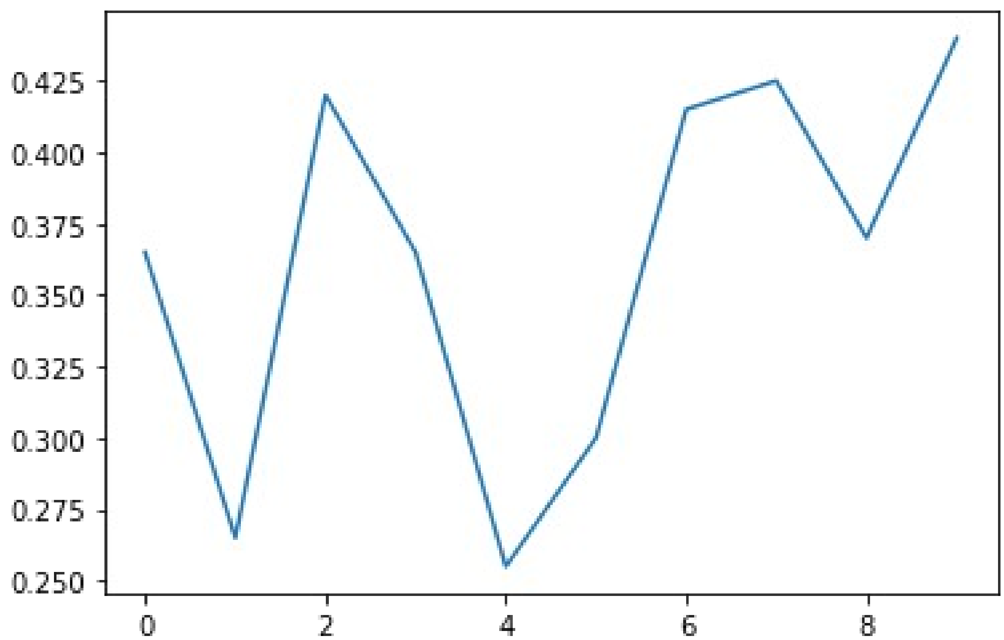
```
(array([ 13., 66., 180., 344., 513., 812., 1017., 934., 275.,  23.]),
 array([0.055 , 0.1145, 0.174 , 0.2335, 0.293 , 0.3525, 0.412 ,  0.4715,
 0.531 , 0.5905, 0.65 ]),
 <BarContainer object of 10 artists>)
```



#line plot

```
plt.plot(d['Diameter'].head(10))
```

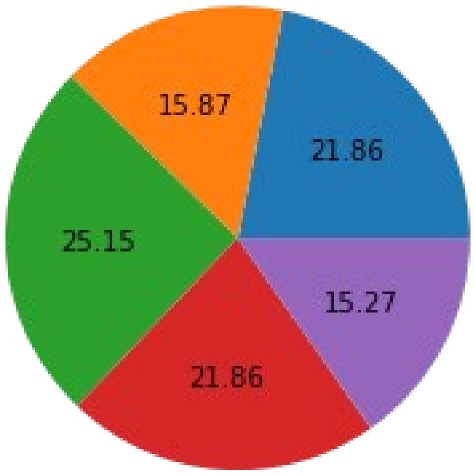
```
[<matplotlib.lines.Line2D at 0x1e536043a90>]
```




```
#piechart
```

```
plt.pie(d['Diameter'].head(),autopct='%.2f')
```

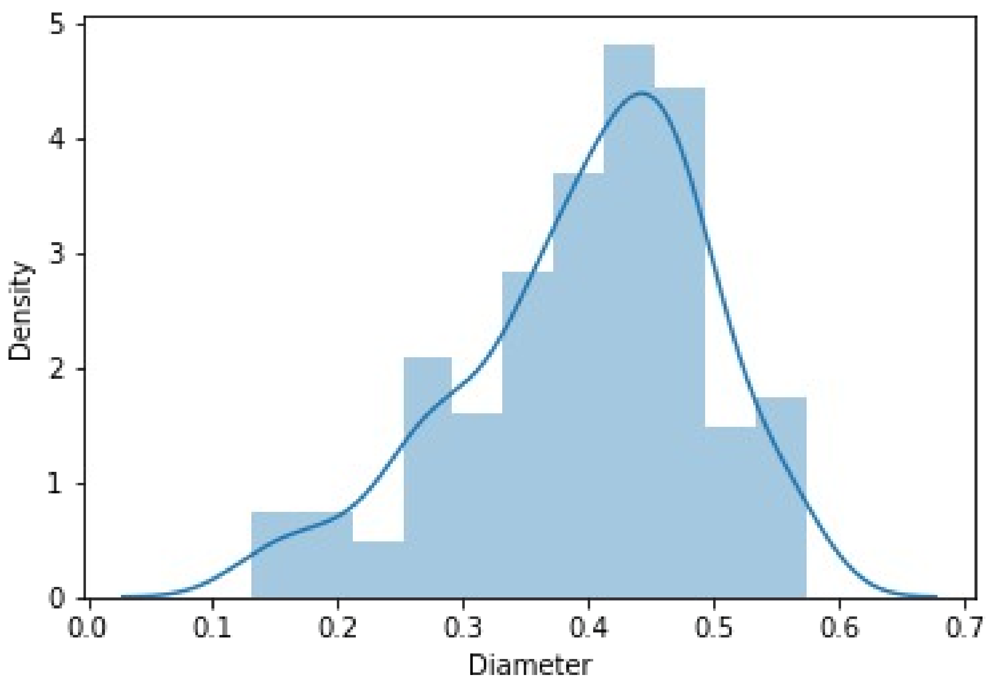
```
([<matplotlib.patches.Wedge at 0x1e5360b52e0>,
<matplotlib.patches.Wedge at 0x1e5360b5a00>,
<matplotlib.patches.Wedge at 0x1e5360c3160>,
<matplotlib.patches.Wedge at 0x1e5360c3880>,
<matplotlib.patches.Wedge at 0x1e5360c3fa0>],
 [Text(0.8507215626110558, 0.6973326486753676, ''),
Text(-0.32611344931648134, 1.0505474849691026, ''),
Text(-1.0998053664078908, -0.020691931287471744, ''),
Text(-0.08269436219656089, -1.096887251480709, ''),
Text(0.9758446362287218, -0.5076684409569241, '')],
[Text(0.464029943242394, 0.3803632629138369, '21.86'),
Text(-0.17788006326353525, 0.5730259008922377, '15.87'),
Text(-0.5998938362224858, -0.011286507974984419, '25.15'),
Text(-0.045106015743578656, -0.5983021371712958, '21.86'),
Text(0.5322788924883937, -0.2769100587037768, '15.27')])
```



#distplot

sns.distplot(d['Diameter'].head(200))

<AxesSubplot:xlabel='Diameter', ylabel='Density'>

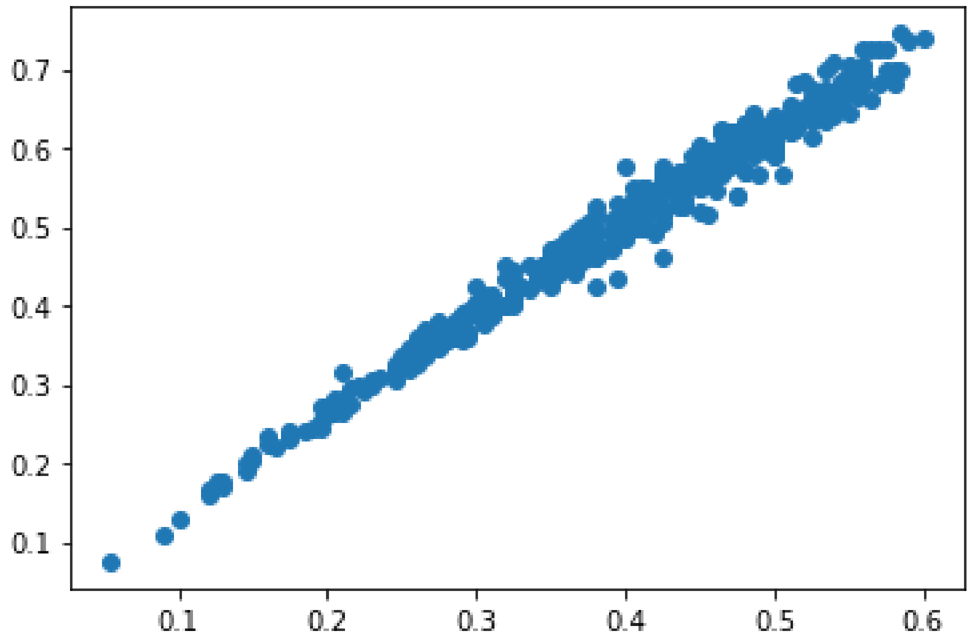


· Bi-Variate Analysis

#scatter plot

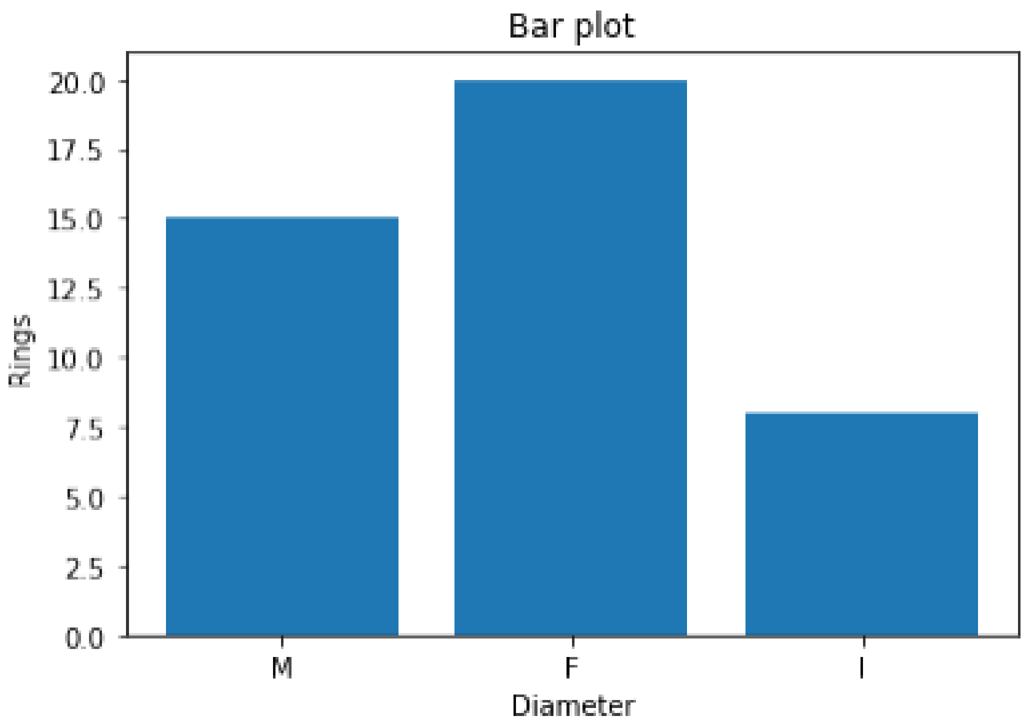
plt.scatter(d['Diameter'].head(500),d['Length'].head(500))

<matplotlib.collections.PathCollection at 0x1e5365f1ca0>



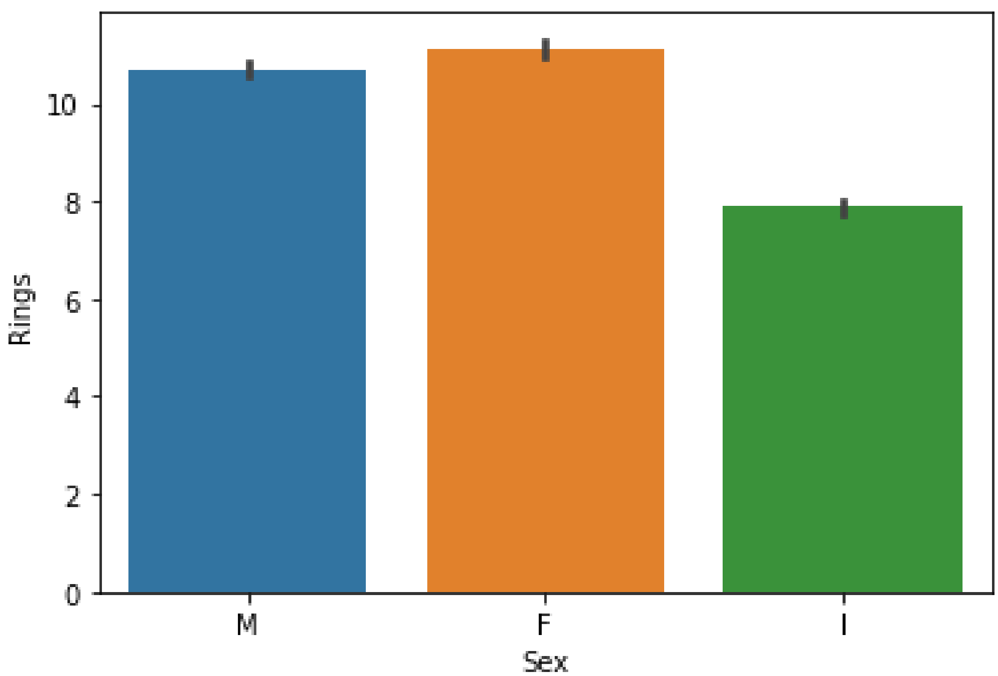

```
#bar plot
plt.bar(d['Sex'].head(10),d['Rings'].head(10)) #labelling of x,y

and result

plt.title('Bar plot')
plt.xlabel('Diameter')
plt.ylabel('Rings')
```

Text(0, 0.5, 'Rings')



sns.barplot(d['Sex'], d['Rings'])

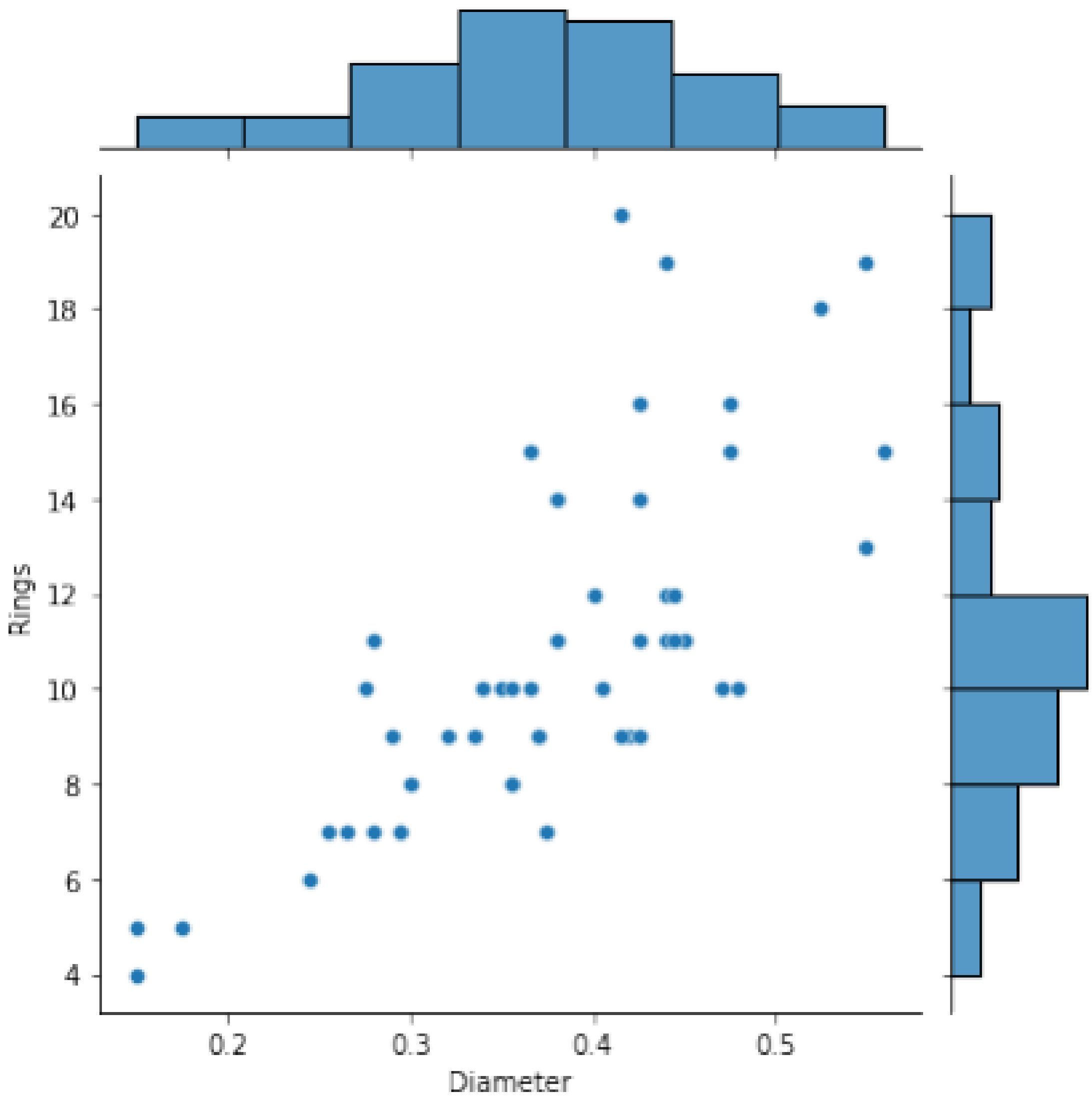
<AxesSubplot:xlabel='Sex', ylabel='Rings'>



#joint plot

sns.jointplot(d['Diameter'].head(50),d['Rings'].head(50))

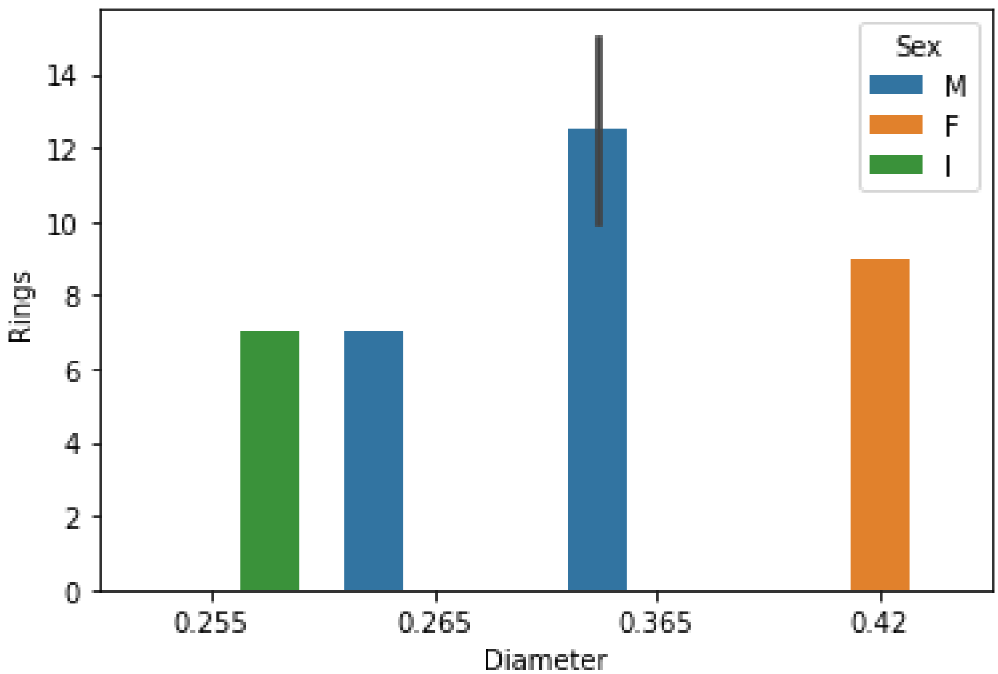
<seaborn.axisgrid.JointGrid at 0x1e5367031f0>



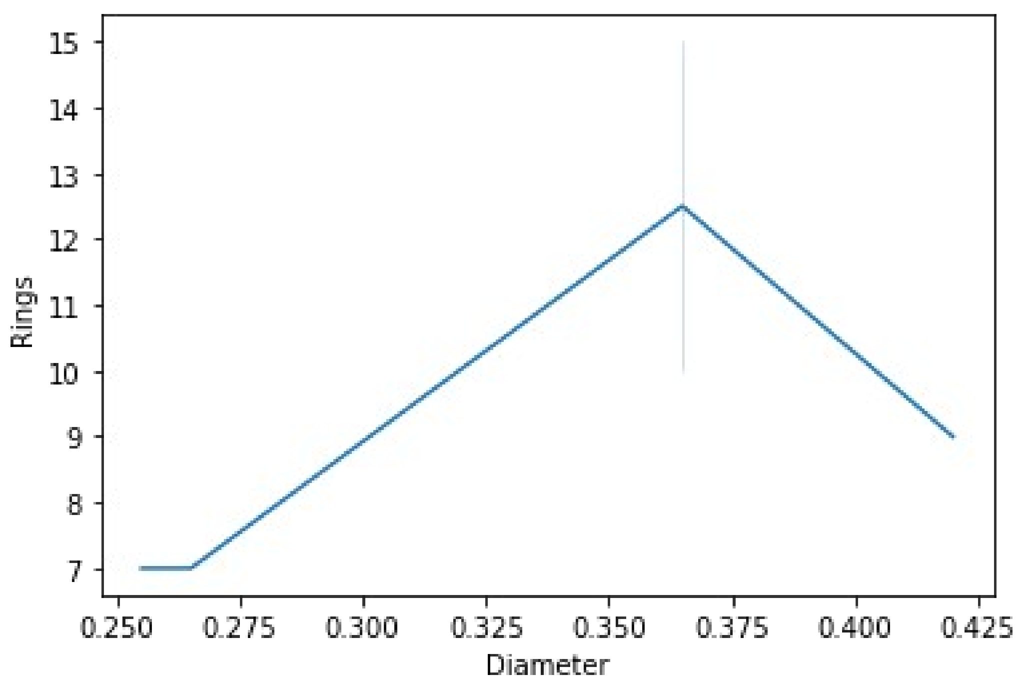
#bar plot

```
sns.barplot('Diameter','Rings',hue='Sex',data=d.head())
```

<AxesSubplot:xlabel='Diameter', ylabel='Rings'>



sns.lineplot(d['Diameter'].head(),d['Rings'].head())
<AxesSubplot:xlabel='Diameter', ylabel='Rings'>

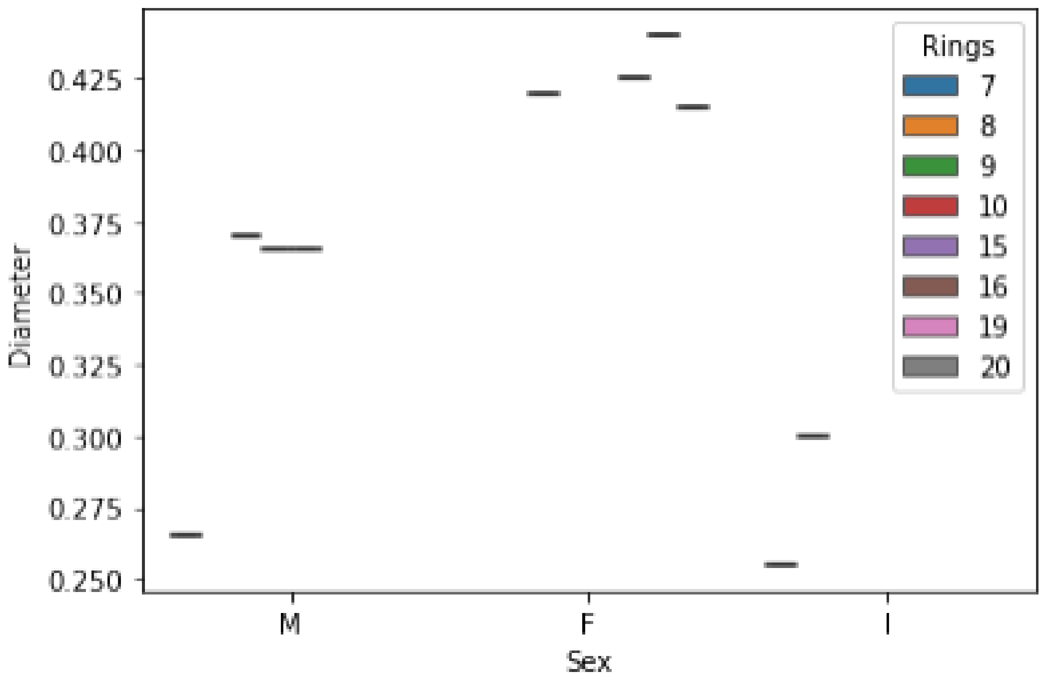


· Multi-Variate Analysis

#boxplot

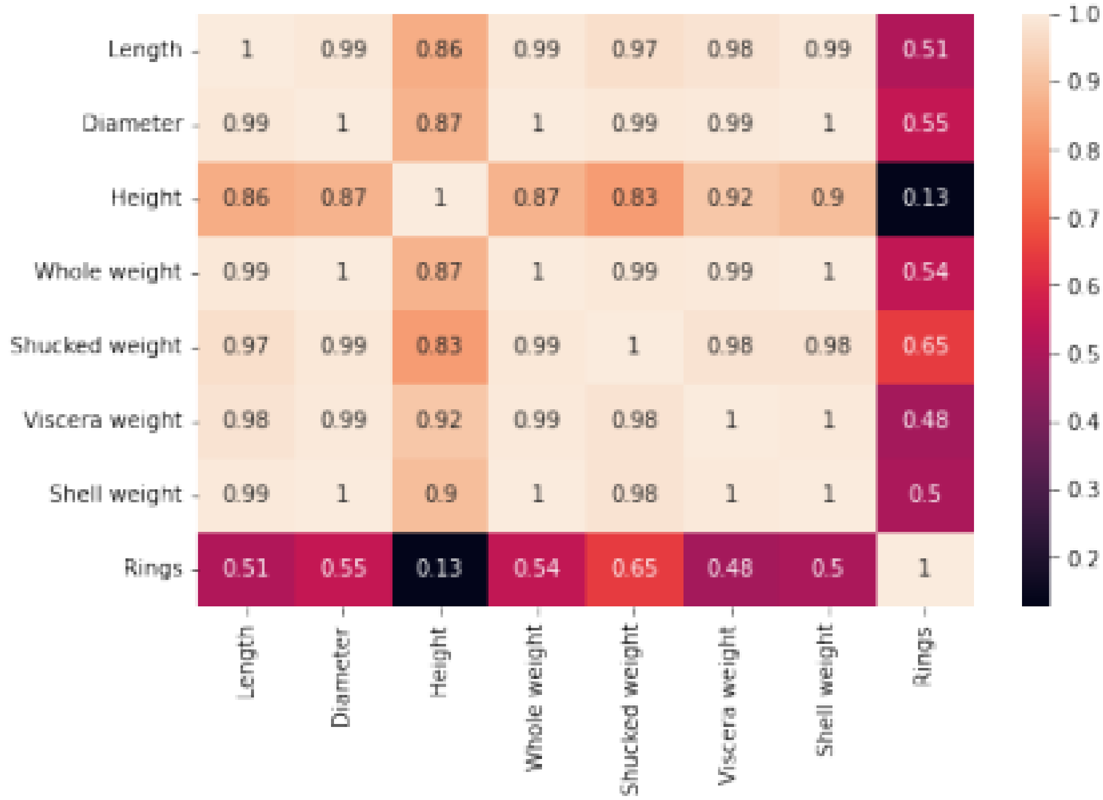
sns.boxplot(d['Sex'].head(10),d['Diameter'].head(10),d['Rings'].head(1 0))

<AxesSubplot:xlabel='Sex', ylabel='Diameter'>




```
#heat map

fig=plt.figure(figsize=(8,5))
sns.heatmap(d.head().corr(),annot=True)
```

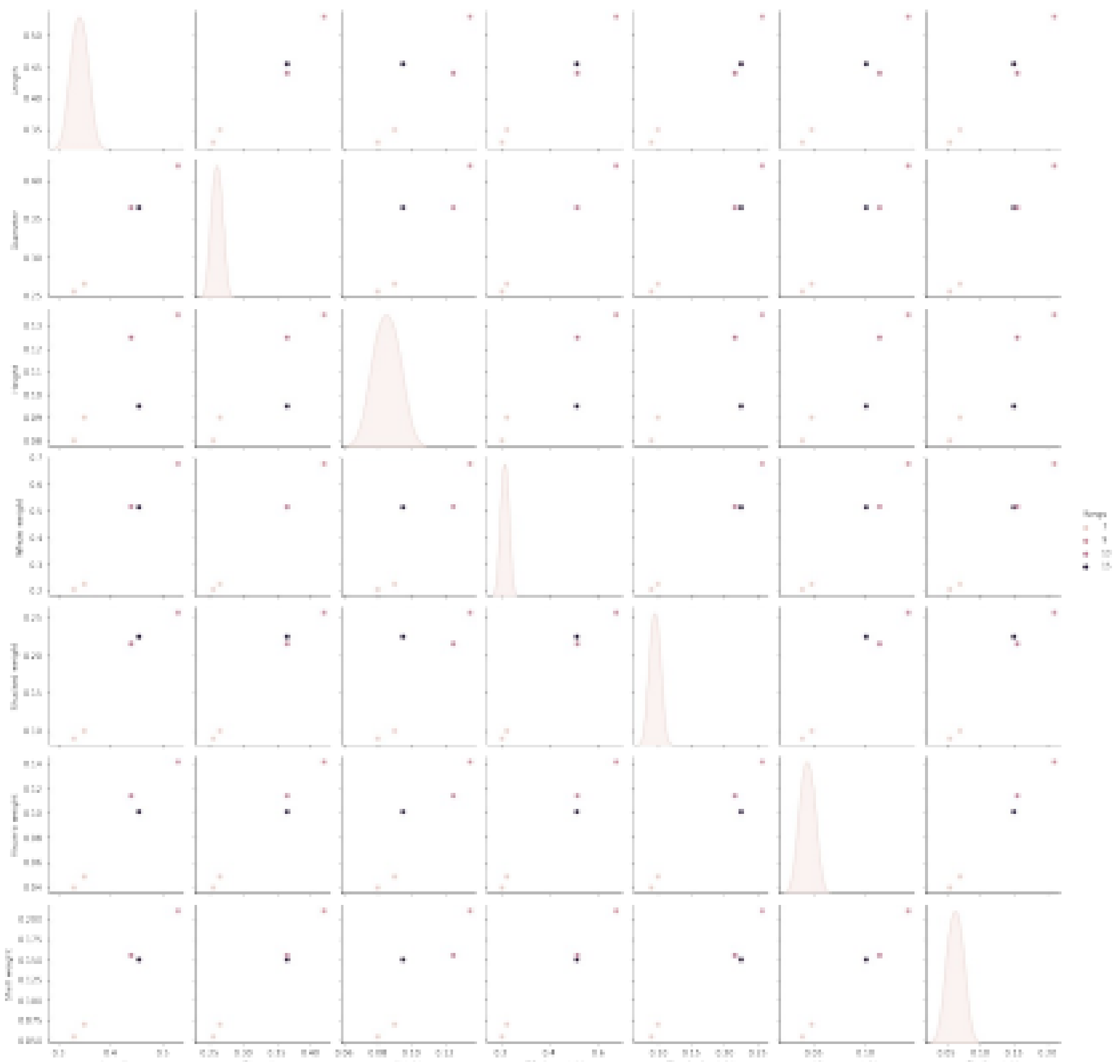
<AxesSubplot:>



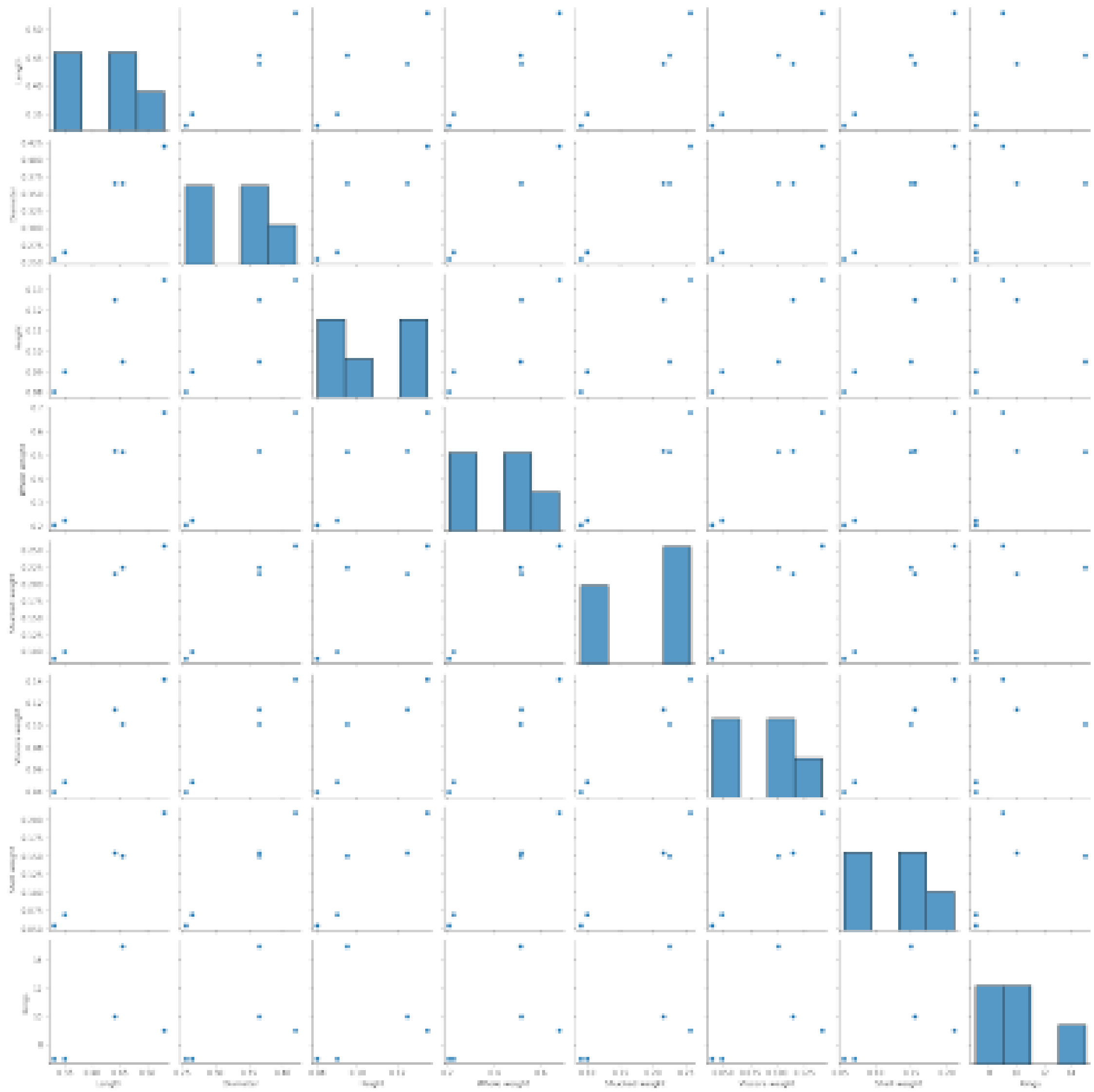
#pair plot

```
sns.pairplot(d.head(),hue='Rings')
```

<seaborn.axisgrid.PairGrid at 0x1e537a68d00>



```
sns.pairplot(d.head())
<seaborn.axisgrid.PairGrid at 0x1e53a6d6640>
```



3. Perform descriptive statistics on the dataset.

d.head()

Sex Length Diameter Height Whole weight Shucked weight Viscera weight \
0 M 0.455 0.365 0.095 0.5140 0.2245 0.1010
1 M 0.350 0.265 0.090 0.2255 0.0995 0.0485
2 F 0.530 0.420 0.135 0.6770 0.2565 0.1415
3 M 0.440 0.365 0.125 0.5160 0.2155 0.1140
4 I 0.330 0.255 0.080 0.2050 0.0895 0.0395
Shell weight Rings
0 0.150 15
1 0.070 7
2 0.210 9
3 0.155 10
4 0.055 7

d.tail()

Sex Length Diameter Height Whole weight Shucked weight \ 4172 F 0.565 0.450 0.165 0.8870 0.3700 4173 M 0.590 0.440 0.135 0.9660 0.4390 4174 M 0.600 0.475 0.205 1.1760 0.5255 4175 F 0.625 0.485 0.150 1.0945 0.5310 4176 M 0.710 0.555 0.195 1.9485 0.9455

Viscera weight Shell weight Rings
4172 0.2390 0.2490 11
4173 0.2145 0.2605 10
4174 0.2875 0.3080 9
4175 0.2610 0.2960 10
4176 0.3765 0.4950 12

d.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4177 entries, 0 to 4176
Data columns (total 9 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 Sex 4177 non-null object
1 Length 4177 non-null float64
2 Diameter 4177 non-null float64
3 Height 4177 non-null float64
4 Whole weight 4177 non-null float64
5 Shucked weight 4177 non-null float64
6 Viscera weight 4177 non-null float64
7 Shell weight 4177 non-null float64
8 Rings 4177 non-null int64
dtypes: float64(7), int64(1), object(1)
memory usage: 293.8+ KB

d.describe()

Length Diameter Height Whole weight Shucked weight \
count 4177.000000 4177.000000 4177.000000 4177.000000 4177.000000
mean 0.523992 0.407881 0.139516 0.828742 0.359367
std 0.120093 0.099240 0.041827 0.490389 0.221963
min 0.075000 0.055000 0.000000 0.002000 0.001000
25% 0.450000 0.350000 0.115000 0.441500 0.186000
50% 0.545000 0.425000 0.140000 0.799500 0.336000
75% 0.615000 0.480000 0.165000 1.153000 0.502000
max 0.815000 0.650000 1.130000 2.825500 1.488000

Viscera weight Shell weight Rings count 4177.000000 4177.000000 4177.000000 mean 0.180594 0.238831 9.933684 std 0.109614 0.139203 3.224169 min 0.000500 0.001500

1.000000 25% 0.093500 0.130000 8.000000 50% 0.171000 0.234000 9.000000 75% 0.253000 0.329000 11.000000 max 0.760000 1.005000 29.000000

d.mode().T

0 1
Sex M NaN
Length 0.55 0.625
Diameter 0.45 NaN
Height 0.15 NaN
Whole weight 0.2225 NaN
Shucked weight 0.175 NaN
Viscera weight 0.1715 NaN
Shell weight 0.275 NaN
Rings 9.0 NaN

d.shape

(4177, 9)

d.kurt()

Length 0.064621
Diameter -0.045476
Height 76.025509
Whole weight -0.023644
Shucked weight 0.595124
Viscera weight 0.084012
Shell weight 0.531926
Rings 2.330687
dtype: float64
d.skew()

Length -0.639873
Diameter -0.609198
Height 3.128817
Whole weight 0.530959
Shucked weight 0.719098
Viscera weight 0.591852
Shell weight 0.620927
Rings 1.114102
dtype: float64

d.var()

Length 0.014422
Diameter 0.009849
Height 0.001750
Whole weight 0.240481

Shucked weight 0.049268
Viscera weight 0.012015
Shell weight 0.019377
Rings 10.395266
dtype: float64

d.nunique()

Sex 3
Length 134
Diameter 111
Height 51
Whole weight 2429
Shucked weight 1515
Viscera weight 880
Shell weight 926
Rings 28
dtype: int64

4. Check for Missing values and deal with them.

#finding missing values

d.isna()

Sex Length Diameter Height Whole weight Shucked weight \ 0 False False False False False False 1 False False False False False False 2 False False False False False False 3 False False False False False False 4 False False False False False False ... ... 4172 False False False False False False
4173 False False False False False False 4174 False False False False False False 4175 False False False False False False 4176 False False False False False False

Viscera weight Shell weight Rings
0 False False False
1 False False False
2 False False False
3 False False False
4 False False False
... ...
4172 False False False
4173 False False False
4174 False False False
4175 False False False
4176 False False False

[4177 rows x 9 columns]

d.isna().any()

Sex False
Length False
Diameter False
Height False
Whole weight False
Shucked weight False
Viscera weight False
Shell weight False
Rings False
dtype: bool

```
d.isna().sum()
```

Sex 0
Length 0
Diameter 0
Height 0
Whole weight 0
Shucked weight 0
Viscera weight 0
Shell weight 0
Rings 0
dtype: int64

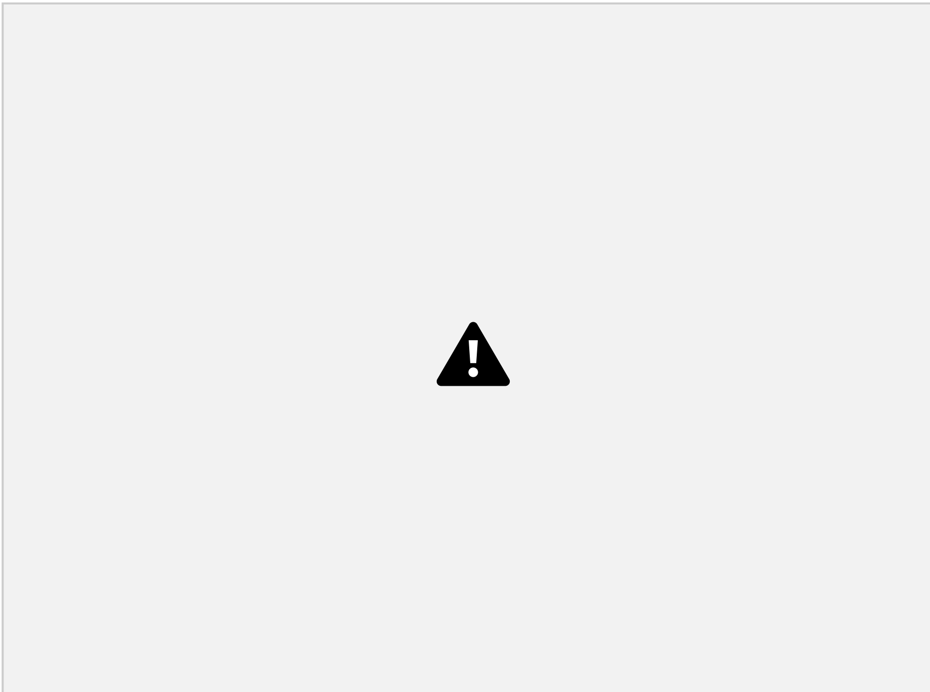
```
d.isna().any().sum()
#no missing values
```

0

5. Find the outliers and replace them outliers

```
#finding outliers

sns.boxplot(d['Diameter'])
```

<AxesSubplot:xlabel='Diameter'>



```
#handling outliers

qnt=d.quantile(q=[0.25,0.75])
qnt
```

```
 Length Diameter Height Whole weight Shucked weight Viscera  weight \
0.25 0.450 0.35 0.115 0.4415 0.186  0.0935
0.75 0.615 0.48 0.165 1.1530 0.502  0.2530

 Shell weight Rings
0.25 0.130 8.0
0.75 0.329 11.0
```

```
iqr=qnt.loc[0.75]-qnt.loc[0.25]

iqr
```

```
Length 0.1650
Diameter 0.1300
Height 0.0500
Whole weight 0.7115
Shucked weight 0.3160
Viscera weight 0.1595
Shell weight 0.1990
Rings 3.0000
dtype: float64
```

```
lower=qnt.loc[0.25]-(1.5*iqr)
```

lower

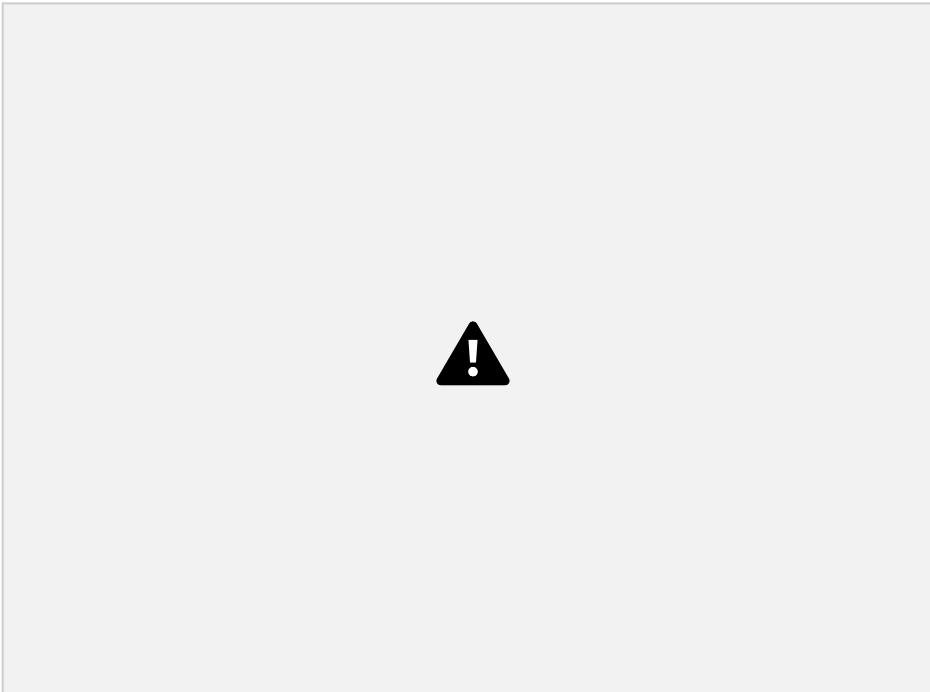
```
Length 0.20250
Diameter 0.15500
Height 0.04000
Whole weight -0.62575
Shucked weight -0.28800
Viscera weight -0.14575
Shell weight -0.16850
Rings 3.50000
dtype: float64
```

```
upper=qnt.loc[0.75]+(1.5*iqr)
upper
```

```
Length 0.86250
Diameter 0.67500
Height 0.24000
Whole weight 2.22025
Shucked weight 0.97600
Viscera weight 0.49225
Shell weight 0.62750
Rings 15.50000
dtype: float64
```

```
# replacing outliers

##Diameter
d['Diameter']=np.where(d['Diameter']<0.155,0.4078,d['Diameter'])
sns.boxplot(d['Diameter'])
```

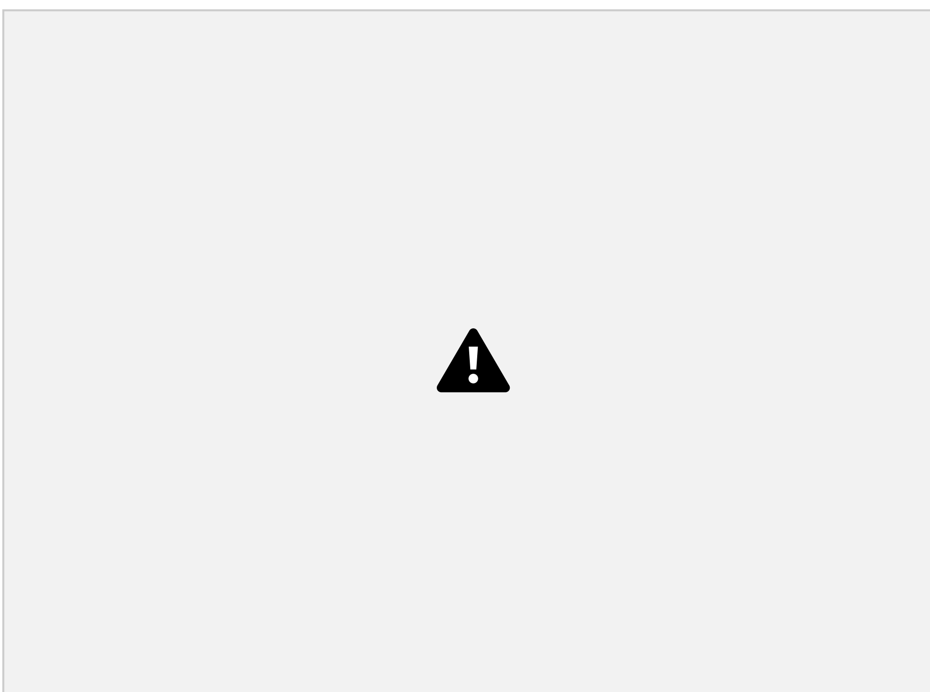
```
<AxesSubplot:xlabel='Diameter'>
```



```
## Length
```

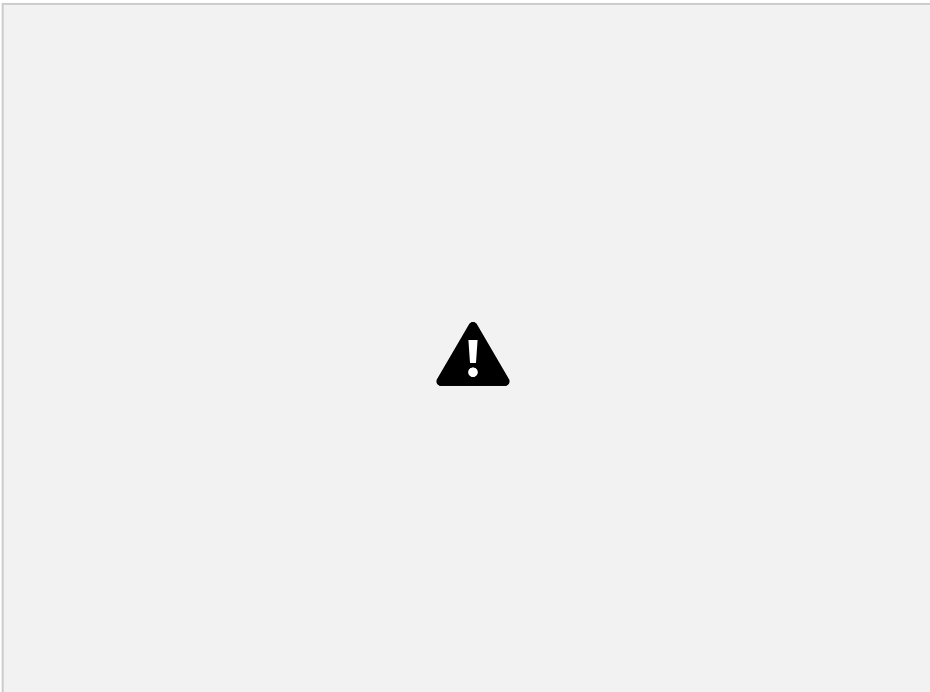
```
sns.boxplot(d['Length'])
```

```
<AxesSubplot:xlabel='Length'>
```



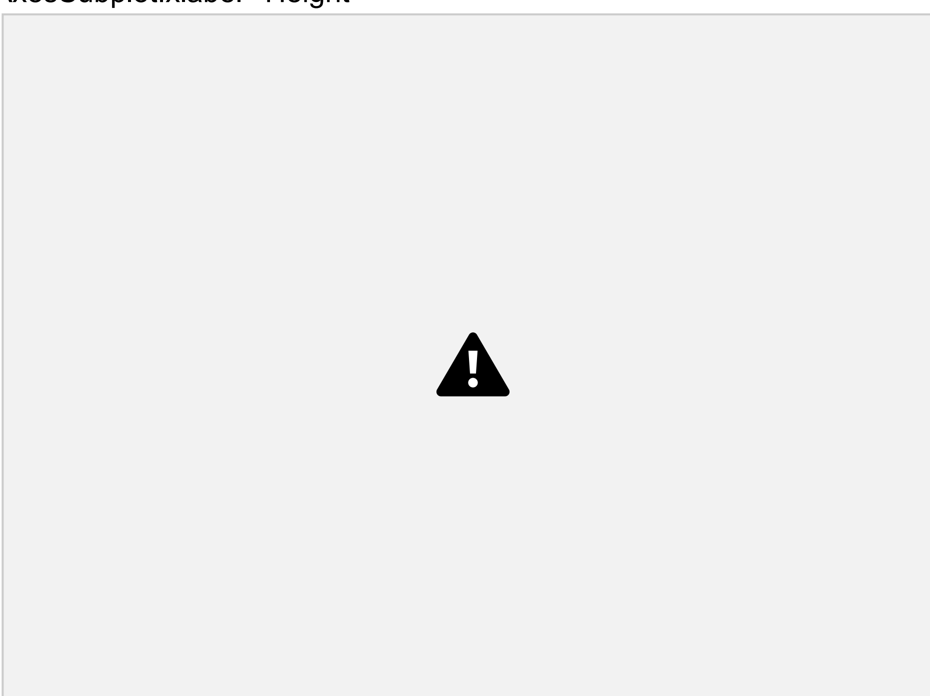
```
d['Length']=np.where(d['Length']<0.23,0.52, d['Length'])
sns.boxplot(d['Length'])
```

<AxesSubplot:xlabel='Length'>



```
## Height
sns.boxplot(d['Height'])
```

<AxesSubplot:xlabel='Height'>



```
d['Height']=np.where(d['Height']<0.04,0.139, d['Height'])
```

```
d['Height']=np.where(d['Height']>0.23,0.139, d['Height'])
```

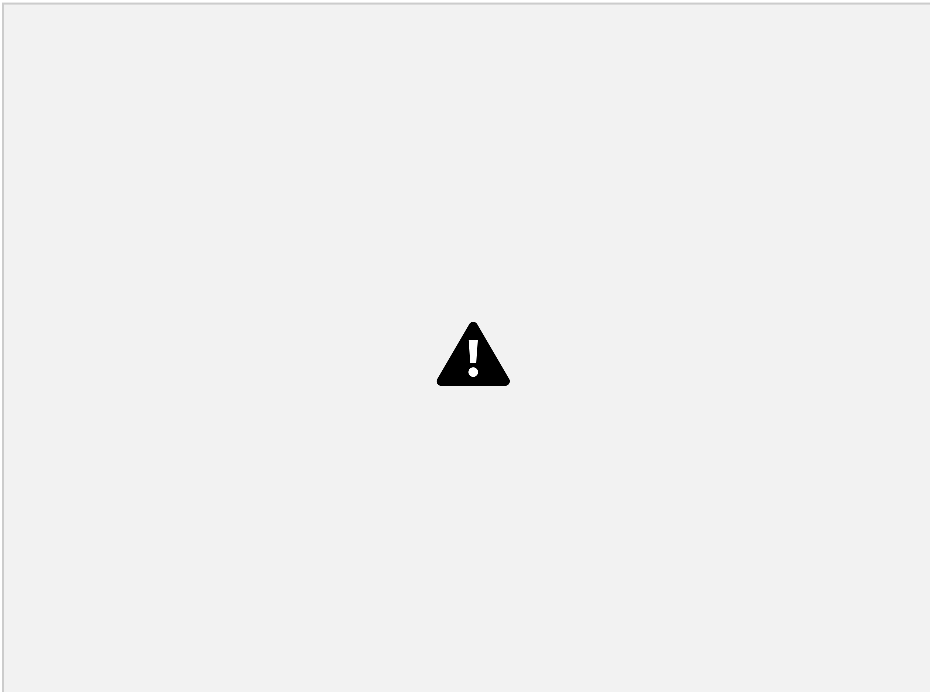
```
sns.boxplot(d['Height'])
```

```
<AxesSubplot:xlabel='Height'>
```

Whole weight

```
sns.boxplot(d['Whole weight'])
```

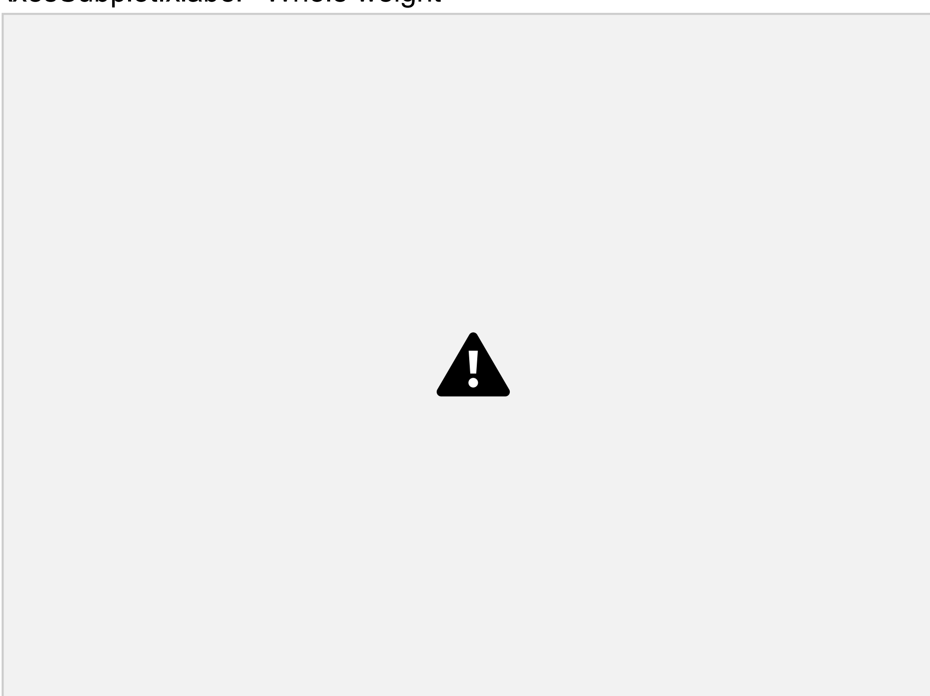
```
<AxesSubplot:xlabel='Whole weight'>
```



```
d['Whole weight']=np.where(d['Whole weight']>0.9,0.82, d['Whole  weight'])
```

```
sns.boxplot(d['Whole weight'])
```

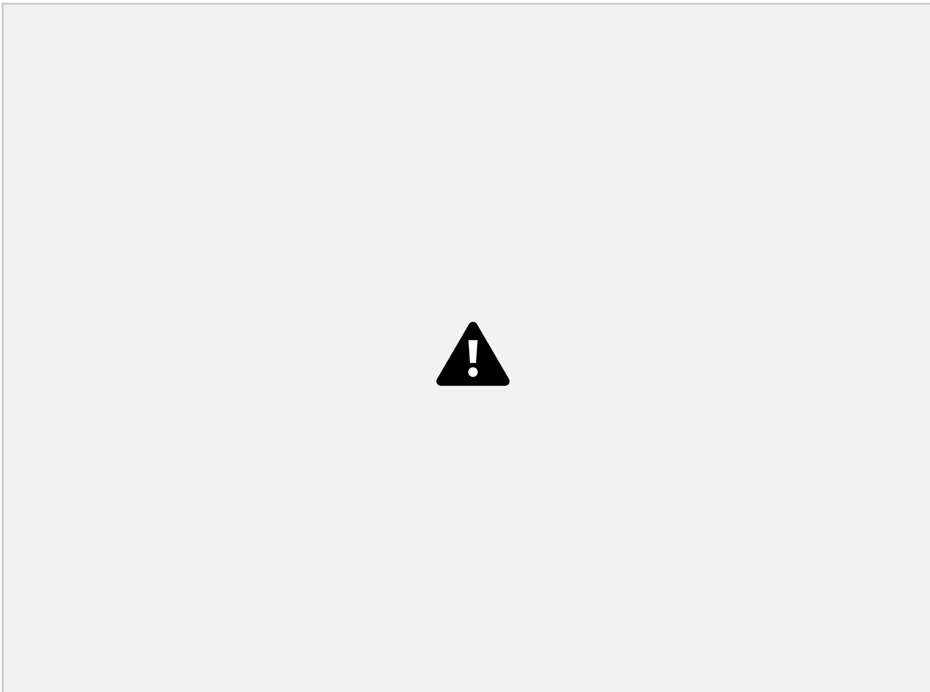
<AxesSubplot:xlabel='Whole weight'>



```
## Shucked weight
```

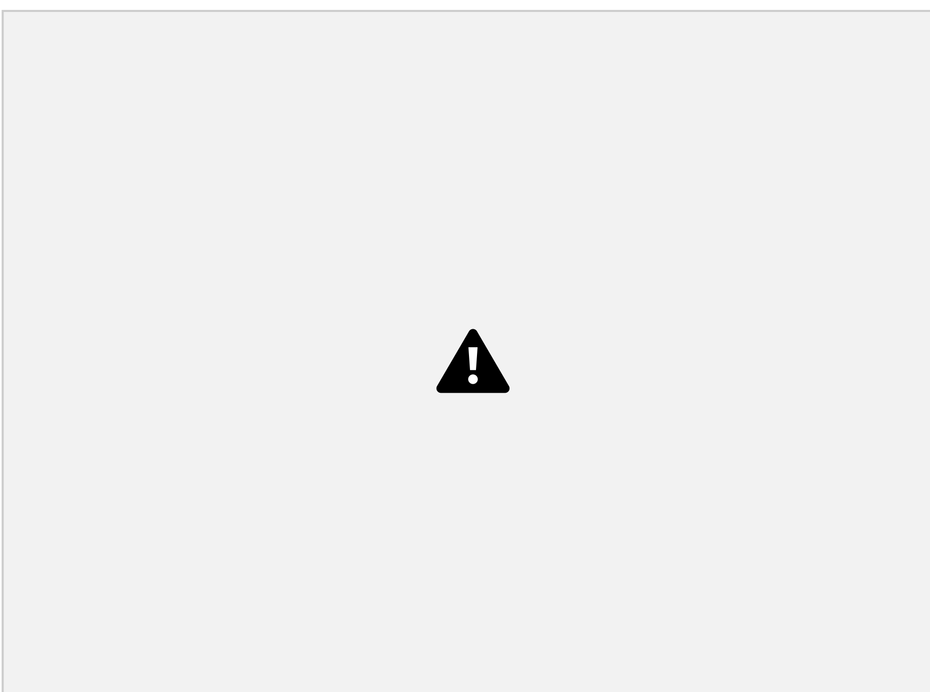
```
sns.boxplot(d['Shucked weight'])
```

<AxesSubplot:xlabel='Shucked weight'>



```
d['Shucked weight']=np.where(d['Shucked weight']>0.93,0.35, d['Shucked weight'])
sns.boxplot(d['Shucked weight'])
```

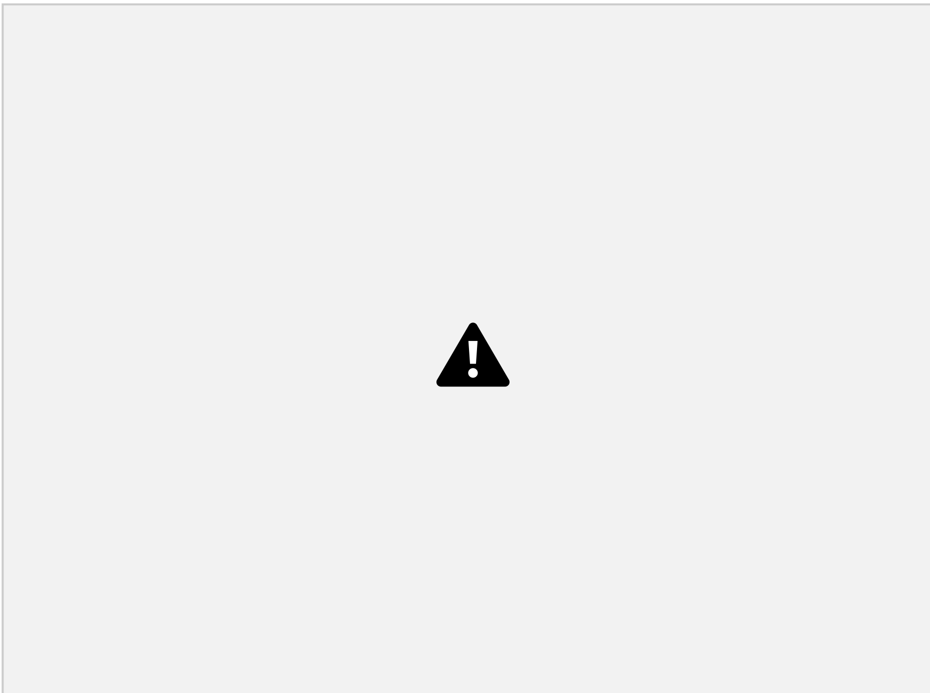
<AxesSubplot:xlabel='Shucked weight'>



Viscera weight

```
sns.boxplot(d['Viscera weight'])
```

```
<AxesSubplot:xlabel='Viscera weight'>
```

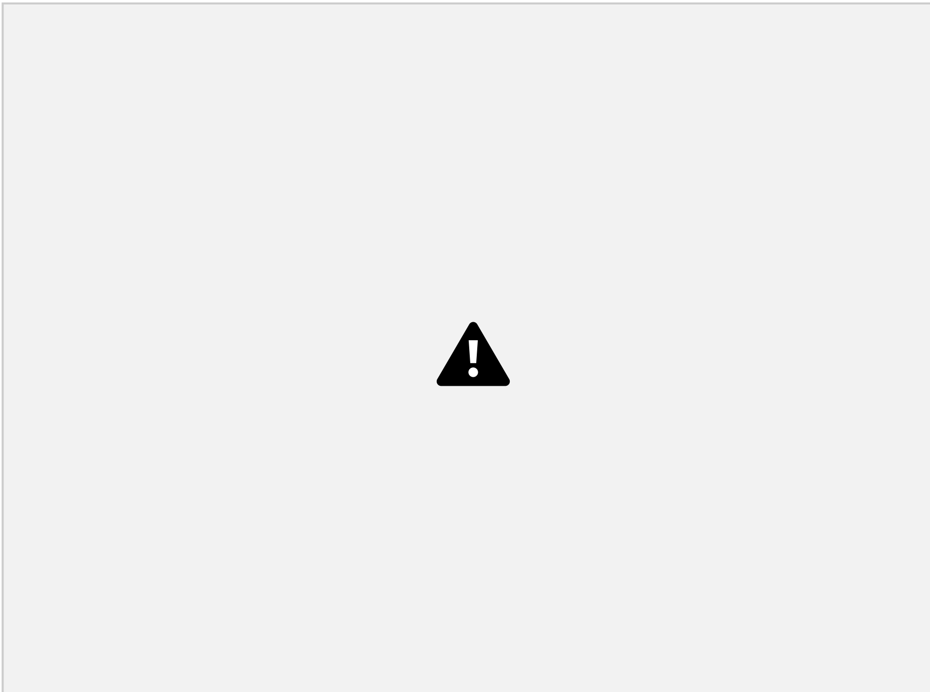


```
d['Viscera weight']=np.where(d['Viscera weight']>0.46,0.18, d['Viscera weight'])
```

```
sns.boxplot(d['Viscera weight'])
```

```
<AxesSubplot:xlabel='Viscera weight'>
```

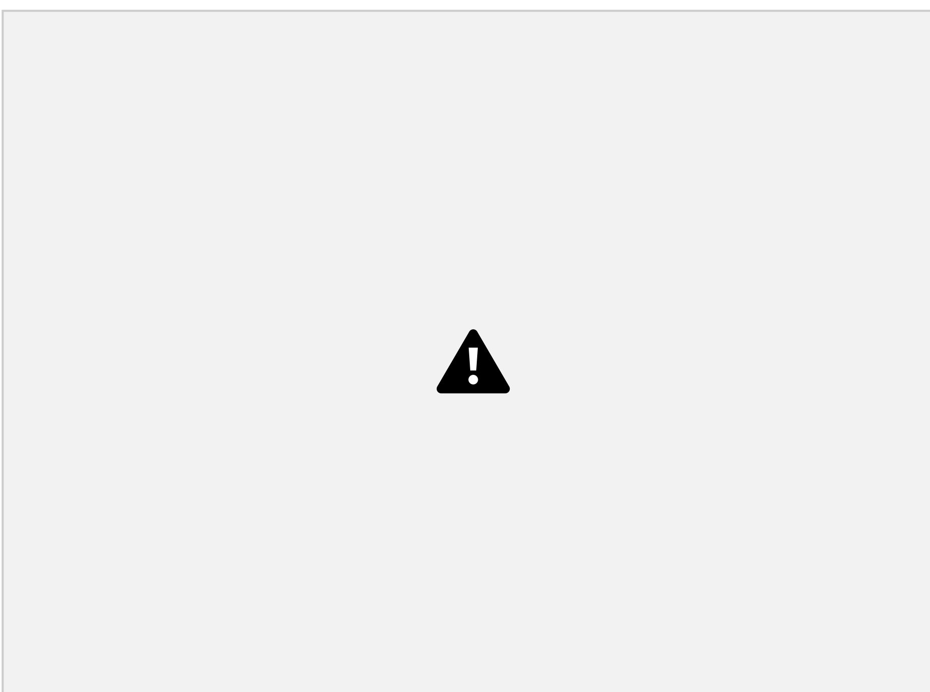
```
d['Shell weight']=np.where(d['Shell weight']>0.61,0.2388, d['Shell  weight'])
```



Shell weight

```
sns.boxplot(d['Shell weight'])
```

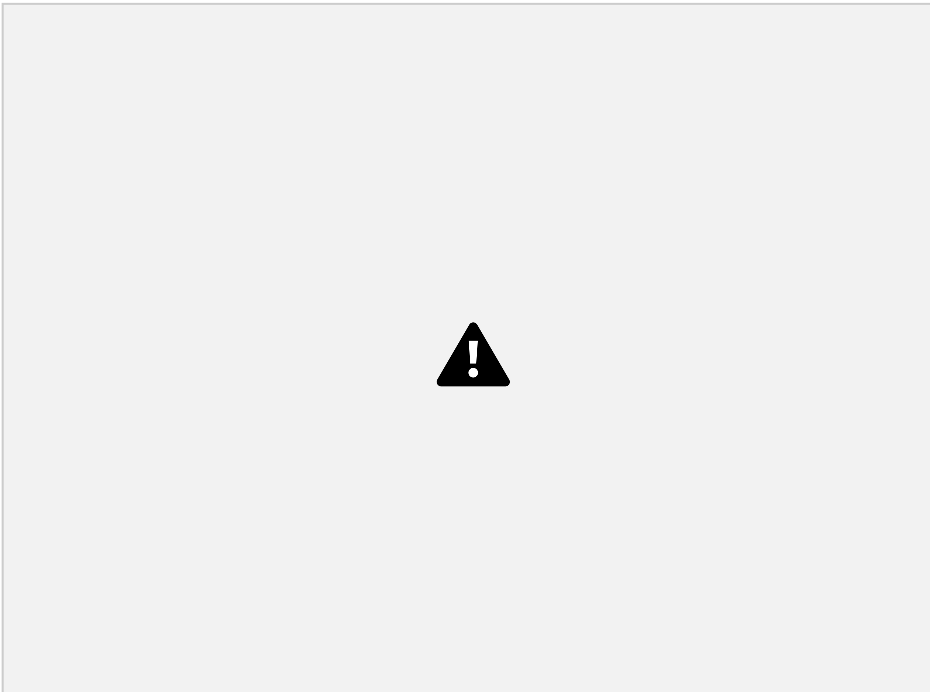
```
<AxesSubplot:xlabel='Shell weight'>
```



```
d['Shell weight']=np.where(d['Shell weight']>0.61,0.2388, d['Shell  weight'])
```

```
sns.boxplot(d['Shell weight'])
```

<AxesSubplot:xlabel='Shell weight'>



6. Check for Categorical columns and perform encoding. *#one hot encoding*

```
d['Sex'].replace({'M':1,'F':0,'I':2},inplace=True)
d
```

Sex Length Diameter Height Whole weight Shucked weight \ 0 1 0.455 0.365 0.095 0.5140 0.2245 1 1 0.350 0.265 0.090 0.2255 0.0995 2 0 0.530 0.420 0.135 0.6770 0.2565 3 1 0.440 0.365 0.125 0.5160 0.2155 4 2 0.330 0.255 0.080 0.2050 0.0895 ... ... 4172 0 0.565 0.450 0.165 0.8870 0.3700 4173 1 0.590 0.440 0.135 0.8200 0.4390 4174 1 0.600 0.475 0.205 0.8200 0.5255 4175 0 0.625 0.485 0.150 0.8200 0.5310 4176 1 0.710 0.555 0.195 0.8200 0.3500

Viscera weight Shell weight Rings
0 0.1010 0.1500 15
1 0.0485 0.0700 7
2 0.1415 0.2100 9
3 0.1140 0.1550 10
4 0.0395 0.0550 7
... ...
4172 0.2390 0.2490 11
4173 0.2145 0.2605 10
4174 0.2875 0.3080 9

4175 0.2610 0.2960 10
4176 0.3765 0.4950 12

[4177 rows x 9 columns]

7. Split the data into dependent and independent variables.

```
x=d.drop(columns= ['Rings'])
y=d['Rings']
x
```

Sex Length Diameter Height Whole weight Shucked weight \ 0 1 0.455 0.365 0.095 0.5140 0.2245 1 1 0.350 0.265 0.090 0.2255 0.0995 2 0 0.530 0.420 0.135 0.6770 0.2565 3 1 0.440 0.365 0.125 0.5160 0.2155 4 2 0.330 0.255 0.080 0.2050 0.0895 ... ... 4172 0 0.565 0.450 0.165 0.8870 0.3700 4173 1 0.590 0.440 0.135 0.8200 0.4390 4174 1 0.600 0.475 0.205 0.8200 0.5255 4175 0 0.625 0.485 0.150 0.8200 0.5310 4176 1 0.710 0.555 0.195 0.8200 0.3500

Viscera weight Shell weight
0 0.1010 0.1500
1 0.0485 0.0700
2 0.1415 0.2100
3 0.1140 0.1550
4 0.0395 0.0550
... ...
4172 0.2390 0.2490
4173 0.2145 0.2605
4174 0.2875 0.3080
4175 0.2610 0.2960
4176 0.3765 0.4950

[4177 rows x 8 columns]

```
y
```

0 15
1 7
2 9
3 10
4 7
..
4172 11
4173 10
4174 9
4175 10
4176 12
Name: Rings, Length: 4177, dtype: int64

8. Scale the independent variables

```
from sklearn.preprocessing import scale #StandardScaler #Scaling the
independent variables

x = scale(x)
x
```

```
array([[-0.0105225 , -0.67088921, -0.50179694, ..., -0.61037964,  -0.7328165 ,
-0.64358742],
 [-0.0105225 , -1.61376082, -1.57304487, ..., -1.22513334,  -1.24343929,
-1.25742181],
 [-1.26630752, 0.00259051, 0.08738942, ..., -0.45300269,  -0.33890749, -0.18321163],
 ...,
 [-0.0105225 , 0.63117159, 0.67657577, ..., 0.86994729,  1.08111018, 0.56873549],
 [-1.26630752, 0.85566483, 0.78370057, ..., 0.89699645,  0.82336724, 0.47666033],
 [-0.0105225 , 1.61894185, 1.53357412, ..., 0.00683308,  1.94673739, 2.00357336]])
```

9. Split the data into training and testing

```
from sklearn.model_selection import train_test_split #spliting data to
train and test

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2)
print(x_train.shape, x_test.shape)
```

```
(3341, 8) (836, 8)
```

10. Build the Model

```
#Multiple Regression

from sklearn.linear_model import LinearRegression

MLR=LinearRegression()
```

11. Train the model

```
MLR.fit(x_train,y_train)
```

```
LinearRegression()
```

12. Test the model

```
#predcition on the test data
y_pred=MLR.predict(x_test)
```

y_pred

```
array([ 9.56232067, 7.19755627, 12.17545567, 6.68796889,  6.81141039,
 11.65253734, 10.10186643, 12.62557907, 8.25152298,  6.65473336,
 6.355173 , 13.85536788, 13.17311525, 11.3295467 ,  12.05336871,
 11.96120439, 7.00090769, 9.18627868, 11.73427744,  11.0171339 ,
 7.6284892 , 8.87680512, 6.48895197, 9.20434217,  10.62571717,
 10.14539453, 10.41366006, 6.09844279, 6.67067929,  11.11967459,
 6.18849609, 7.11321264, 10.02484165, 9.77822264,  10.06320689,
 12.04543114, 6.05593959, 8.03652727, 9.7830606 ,  12.04151177,
 10.70812656, 12.1246719 , 10.42554762, 11.10153426,  10.97225975,
 11.92181641, 11.9025537 , 10.03535105, 10.78987261,  9.68726745,
 10.21396975, 7.70486124, 10.83707079, 12.54430014,  8.53898635,
 9.06169949, 9.45477988, 10.94886012, 6.02923604,  10.2050178 ,
 6.93960582, 7.65565349, 10.03715641, 9.16331448,  10.90389946,
 8.87554261, 11.80272697, 11.08781041, 11.45060668,  11.90064295,
 8.6472278 , 11.96307302, 11.67355714, 9.37441567,  12.55620715,
 8.90059876, 10.30155303, 9.01169321, 9.90038968,  6.55970736,
 10.52780985, 8.26821369, 10.80548611, 11.64263581,  11.5876582 ,
 11.39864742, 12.629618 , 7.98642656, 9.55410995,  9.64777038,
 9.15693567, 11.91204116, 11.03371273, 9.03794167,  10.52382872,
 8.27718202, 11.72780033, 6.27673932, 11.49659814,
8.0189265 ,
 6.85420619, 7.82441681, 10.80471331, 11.17594478,  4.19260417,
 11.42012692, 12.00839382, 7.20915411, 11.17238531,  9.76778112,
 10.9888912 , 9.03583738, 11.24398065, 6.88978627,  9.28240342,
 9.56967271, 11.45558465, 9.21542417, 10.35456282,  13.47590751,
 6.91751936, 6.25947229, 8.90243996, 11.21118499,  11.67897969,
 5.99721159, 7.02957567, 13.93755527, 10.53069448,  7.0495923 ,
 9.72987801, 9.78956478, 7.78386675, 6.57598651,  9.75560118,
 11.45569966, 13.82308926, 10.67651445, 7.96954133,  6.15196629,
 12.10883963, 6.47325303, 13.39250483, 10.56361587,  11.69890618,
 11.99554315, 11.03558685, 8.99302786, 12.9219841 ,  7.58283815,
 9.84877006, 6.91302939, 10.73334169, 13.30017585,  6.94918942,
 8.47038306, 10.07122868, 11.39243766, 11.52698767,  10.98817915,
 8.23702779, 9.70877829, 6.84813838, 10.35866912,  7.81223693,
 10.44291198, 10.5559004 , 8.90471959, 11.20643168,  14.58671746,
 7.85696774, 7.70033032, 8.91527121, 10.38462766,  10.81266114,
 10.61299444, 10.33740135, 9.12087508, 9.09108113,  4.86767713,
 7.88876196, 13.02832116, 5.83797433, 8.87546839,  10.0880134 ,
 6.39218486, 11.22511423, 11.01291911, 11.47371279,  7.89357089,
 12.13033577, 12.70151342, 6.30975703, 12.13339173,  9.03824085,
 10.49580129, 7.83703526, 14.35201795, 9.34878227,  10.27375611,
 7.68348697, 7.73595489, 11.10392999, 8.95867563,  10.56750428,
 10.68276382, 12.87180014, 11.21139587, 10.20702161,  13.40091701,
 6.74438333, 13.08193446, 8.31107712, 9.36896727,  10.70329902,
 7.48994415, 7.77612422, 11.32350381, 10.76970951,  6.32581212,
 8.62993769, 9.82587737, 12.9138691 , 10.65099952,
```

7.05114066,
8.83496628, 9.78640747, 8.78408313, 10.24535895, 14.08372062,
11.59415124, 10.87500042, 10.40000133, 8.3895059 , 15.78729998,
10.8973955 , 9.96939367, 10.92857001, 13.96806652, 9.90891346,
14.40620221, 9.96111 , 10.59752707, 10.64218456, 11.61164488,
9.54930214, 9.18536216, 9.48326306, 12.45086968, 10.31577507,
7.02412218, 8.60074094, 6.55853338, 6.8267182 , 11.27806243,
9.62417698, 9.30763675, 7.71225167, 6.31152615, 10.65890622,
10.68555259, 10.47415439, 5.97415411, 11.24080809, 15.36865222,
8.42572424, 10.47581488, 12.67956958, 9.09770539, 10.21013328,
8.03480179, 10.46345867, 11.20277741, 8.41871839, 11.96332735,
11.02374995, 8.49089066, 10.20382982, 10.98642802, 9.8395771 ,
7.03844752, 10.52435289, 9.23240578, 13.73167758, 9.78632697,
11.87803093, 9.02053589, 10.66567153, 11.47439559, 8.66190024,
7.72719912, 10.02167157, 10.0397404 , 11.44115123, 12.38901892,
10.37369673, 12.45312777, 8.00594776, 12.05307942, 12.79031892,
9.8289306 , 7.53823374, 9.82368247, 6.91506282, 11.59334373,
9.82613846, 13.14555115, 9.75132861, 9.96234844, 12.4633359 ,
8.80422842, 12.13977553, 10.49213303, 8.39859828, 8.09556987,
8.13818713, 10.4195742 , 7.71099146, 6.58020973, 12.41028245,
7.30761739, 12.92665939, 10.86552115, 10.29627044, 8.80648755,
10.45743237, 9.22267413, 12.71777594, 6.77272761, 11.32606369,
7.01994773, 10.7957895 , 11.35523175, 10.65738994, 7.72635768,
10.73264779, 9.75569455, 9.49972763, 10.00130637, 11.2968249 ,
13.29236826, 13.01362707, 9.98872742, 7.68675189, 11.4415783 ,
6.87838442, 8.52730625, 9.59701627, 8.01358992,
8.82689086,
9.89621326, 11.30273005, 11.99665236, 13.55870204, 10.8008616 ,
10.71771219, 10.22492888, 9.59486958, 10.08657927, 9.86177379,
10.87675981, 8.23639723, 6.96563615, 9.64091291, 9.35134204,
11.3713902 , 11.79687973, 8.3415344 , 12.33740515, 10.09429177,
10.90993997, 7.32524091, 12.71760435, 12.30900399, 7.46928688,
10.52889485, 10.25641402, 8.80747026, 6.4886803 , 8.89484225,
11.28645946, 12.74510405, 10.97313574, 9.98264196, 11.66090365,
6.10755118, 7.96874907, 9.97003385, 12.85915978, 8.28489946,
4.16973318, 7.48564612, 9.84484248, 10.28463627, 12.27114797,
15.88134294, 13.38688941, 12.00766564, 10.75216301, 7.25651559,
9.04914777, 10.86390834, 7.88405788, 6.77756046, 7.19857404,
7.64006005, 7.80327296, 9.55032776, 10.11032968, 10.14989236,
8.82595647, 7.75373672, 13.56369047, 11.85046985, 11.13090921,
6.41231543, 11.91717969, 11.97654572, 9.26396909, 7.50970957,
5.81365971, 9.74906399, 10.85942598, 7.16195626, 12.3163721 ,
9.17377753, 7.71998958, 11.93902804, 11.40047029, 10.27245869,
7.28733885, 10.22083405, 9.18541857, 15.05378323, 7.74788611,
6.99603236, 8.07834494, 10.32498143, 7.89610644, 7.98350336,
6.5817457 , 8.45855593, 10.78539824, 12.54515265, 8.29881371,
10.40663019, 9.71374454, 9.50135659, 9.88696265, 11.33754137,

10.16854408, 9.90430622, 11.36009559, 11.05639423, 9.7735533 ,
7.26741236, 14.26411407, 10.01857828, 11.08153429, 6.16506505,
7.44821912, 9.83171637, 6.65065386, 9.36001091, 12.04294071,
10.4429228 , 11.25940606, 11.12909448, 7.14959967, 14.75268071,
12.96303909, 7.4460127 , 11.5798998 , 16.54754561,
12.11606318,
3.78890948, 7.55280133, 11.39984769, 10.14297157, 10.61187478,
10.92330859, 11.54046924, 8.35591136, 14.01948383, 9.77531157,
7.9486812 , 11.59037026, 9.50744425, 10.22691729, 11.05136194,
13.42946609, 11.13055869, 7.83304496, 12.16072796, 13.31369032,
10.90079911, 10.32729206, 11.06192388, 10.91181359, 9.12333601,
11.02347167, 9.19676896, 9.91508986, 6.72640207, 11.2824039 ,
6.25427031, 11.95890501, 12.73309032, 10.93692681, 6.38762064,
8.77457034, 9.48245993, 12.60228753, 9.7951261 , 11.71943102,
11.43103689, 10.47864556, 7.11648334, 9.18849762, 9.85463184,
10.61778407, 13.99241061, 8.52469168, 9.91763113, 6.11796109,
9.4105441 , 12.32967309, 11.62159298, 10.84186575, 13.29722575,
10.84784427, 8.62099845, 12.10934053, 12.1074603 , 11.38429785,
13.09403138, 11.82569349, 8.79052453, 10.45329396, 10.33533737,
9.05396193, 7.45008847, 8.45006844, 9.29116274, 8.85639305,
10.03967069, 12.44702562, 10.1860211 , 11.04303132, 8.42543169,
10.47577603, 7.34594268, 4.92548685, 7.29511613, 8.66020343,
13.98445122, 5.96924161, 11.16329713, 7.76957434, 7.91265325,
11.5243711 , 9.54260488, 7.64487298, 11.40748515, 10.05955511,
10.77546012, 12.66006627, 10.35935742, 9.90194466, 10.39883908,
11.55535519, 10.12679234, 7.62623634, 12.56011913, 9.39720742,
9.44165327, 12.38678545, 11.56860978, 9.72031086, 11.28098618,
11.96246646, 5.7476074 , 8.90596281, 11.31393447, 12.03720382,
9.24517098, 10.94306381, 7.10435581, 13.08251421, 13.74271783,
8.63302031, 12.80742216, 11.75722621, 11.64820719, 7.36041164,
9.64376672, 10.32446093, 6.9910132 , 3.4606273 ,
6.81559896,
13.30215359, 8.01103266, 9.32412959, 7.38120115, 8.15483881,
10.61386941, 6.38482086, 10.04058386, 9.50247092, 8.68157451,
11.05204038, 12.60949839, 6.16580407, 6.4883941 , 11.32828216,
12.16415202, 12.04129788, 11.12820857, 16.1797968 , 9.85268505,
7.14240443, 5.68008775, 7.59506098, 9.7947073 , 5.76704797,
5.96216412, 6.89910569, 12.05203092, 10.21554238, 9.23625509,
6.10742307, 10.36226228, 10.58147607, 11.15985226, 6.60595775,
10.08913388, 9.85757756, 10.53946848, 12.89819746, 12.62870892,
10.69186255, 9.12814143, 12.78351039, 8.89600934, 15.12114616,
10.18671107, 10.03457443, 13.7098309 , 7.57582391, 8.64269111,
7.44466994, 7.84063618, 11.99664542, 5.82567407, 11.90435322,
9.74401253, 6.21990249, 8.25688733, 11.08915141, 6.50885175,
6.84277558, 10.66697608, 6.4173908 , 8.47428217, 9.32510784,
8.80601731, 10.58530079, 10.00829671, 9.65178478, 12.23060707,
11.26724374, 11.51220403, 9.65550988, 12.3726616 , 11.86086808,

```
 6.52606529, 11.77380153, 6.17661644, 10.57105746,  10.90926988,
 12.43692852, 8.00928114, 8.08213542, 6.53993069,  9.56518065,
 13.32808412, 12.45759861, 12.88301556, 9.52876194,  9.06593465,
 8.13736772, 12.38757752, 11.21990793, 13.36851954,  12.08606712,
 9.15570209, 9.96620173, 11.27427733, 7.23101251,  6.58667927,
 11.920947 , 13.49465942, 7.93522736, 9.68881488,  11.84012367,
 9.848229 , 11.84404276, 7.9725836 , 5.68477384,  7.23609829,
 13.33338893, 7.49832872, 12.07564515, 13.82284372,  10.50482871,
 7.91098681, 12.28372563, 10.18787007, 10.93847527,  13.76725212,
 9.5595163 , 11.17140435, 10.62550836, 11.32866135,
9.70819822,
 10.78808921, 10.76237601, 10.15947041, 9.1145544 ,  12.02838288,
 9.46424999, 7.90935047, 10.17114239, 6.31523371,  11.95339882,
 7.07959422, 11.14934112, 9.55718527, 7.22580872,  9.17148587,
 10.08614925, 12.34158345, 9.17672079, 8.21557651,  10.12395726,
 11.25273283, 9.65749865, 10.66622507, 7.45560855,  11.89636085,
 9.51715616, 8.97747036, 6.57603259, 4.52630698,  7.14197678,
 12.90820924, 10.0190957 , 8.49040488, 11.00376924,  12.30636449,
 7.51267037, 8.68156192, 10.96964327, 6.36407856,  9.32676715,
 8.20027834, 8.12155662, 14.80647383, 13.06915566,  8.97710018,
 11.29185158, 7.67435384, 10.69321897, 3.80658924,  10.91150916,
 8.62020669, 5.99090939, 9.71812511, 12.59389307,  8.69993329,
 8.76714763, 8.93235901, 11.65481558, 11.32456191,  10.21615279,
 7.19681793, 10.16507184, 9.08679183, 10.17958316,  12.48512277,
 9.94874027, 9.68716181, 9.475427 , 9.69144292,  7.755378 ,
 8.52358358, 8.02769212, 8.93442862, 10.22721518,  12.89307961,
 16.11019162, 14.01724005, 6.23021499, 11.65160152,  7.84886117,
 7.11722872, 12.08156893, 11.43641084, 11.14439485,  11.18092429,
 6.74939069, 9.19267552, 10.43072254, 7.50969754,  11.75372709,
 12.21332067, 12.29928177, 10.04061937, 8.62397343,  8.00680224,
 9.58958453, 10.34723651, 10.0148802 , 10.27230306,  14.02064032,
 6.51080709, 12.34834102, 10.21599246, 5.32952315,  12.07628387,
 10.66632177, 5.75233828, 11.09736572, 9.36845413,  11.06439481,
 8.50871786])
```

```
#prediction in the train data
pred=MLR.predict(x_train)
pred
```

```
array([ 9.84586178, 10.70270235, 9.09697411, ..., 12.66223706,  13.87025441,
10.07468021])
```

```
from sklearn.metrics import r2_score
```

```
acc=r2_score(y_test,y_pred)
```

```
acc
```

```
0.453594229684136
```

```
#test this model

MLR.predict([[1,0.455,0.365,0.095,0.5140,0.2245,0.1010,0.150]])
```

array([9.87947217])

13. Measure the performance using Metrics

```
from sklearn import metrics
from sklearn.metrics import mean_squared_error

np.sqrt(mean_squared_error(y_test,y_pred))
```

2.2497338061968057

LASSO

```
from sklearn.linear_model import Lasso, Ridge

#intialising model

lso=Lasso(alpha=0.01,normalize=True)

#fit the model
lso.fit(x_train,y_train)
```

Lasso(alpha=0.01, normalize=True)

```
#predcition on test data

lso_pred=lso.predict(x_test)

#coef
coef=lso.coef_
coef
```

array([-0. , 0. , 0. , 0.45254135, 0.14236687,
0. , 0. , 0.8939626])

```
#accuracy
from sklearn import metrics
from sklearn.metrics import mean_squared_error
metrics.r2_score(y_test,lso_pred)
```

0.35863807123829816

```
#error

np.sqrt(mean_squared_error(y_test,lso_pred))
```

2.4373903322278783

RIDGE

```
rg=Ridge(alpha=0.01,normalize=True)
```

```
#fit
rg.fit(x_train,y_train)
```

Ridge(alpha=0.01, normalize=True)

```
#predcition

rg_pred=rg.predict(x_test)
rg_pred
```

```
array([ 9.52873331, 7.20781215, 12.14767865, 6.71019935,  6.81666869,
 11.62637555, 10.09181694, 12.6749307 , 8.26447249,  6.66887585,
 6.35061288, 13.76824928, 13.11005043, 11.34102509,  12.06965102,
 11.90221671, 6.99402114, 9.10501689, 11.69542511,  11.06740391,
 7.61600088, 8.91141481, 6.53286096, 9.21589982,  10.62252993,
 10.13245706, 10.3881983 , 6.10040118, 6.63766288,  11.19456021,
 6.19015842, 7.10307288, 10.01677544, 9.78530638,  9.99979883,
 11.96245675, 6.04584576, 8.08086386, 9.87919743,  11.94788868,
 10.76350134, 12.13808769, 10.3687511 , 11.08734445,  10.96757218,
 11.9868591 , 11.91002434, 9.99308349, 10.82244284,  9.75195986,
 10.31497792, 7.71543726, 10.9239886 , 12.47764192,  8.63980155,
 9.04202297, 9.59301953, 10.96940323, 6.01179164,  10.34239072,
 6.94516652, 7.67378406, 10.08090979, 9.16452635,
10.85156881,
 8.86701969, 11.72488318, 11.1532381 , 11.39252704,  11.80544378,
 8.63677842, 11.93299921, 11.6697933 , 9.35957161,  12.51656076,
 8.88978106, 10.26314448, 9.03344775, 9.85527096,  6.53860977,
 10.46607091, 8.27135385, 10.76404062, 11.64819114,  11.53579226,
 11.30419 , 12.51373392, 7.96998678, 9.58933524,  9.69829262,
 9.16733839, 11.86493059, 11.05363678, 9.02814165,  10.47667435,
 8.28118469, 11.82642894, 6.31065532, 11.43743562,  8.00478401,
 6.85991831, 7.88367277, 10.81220802, 11.11626132,  4.31971474,
 11.43569451, 12.01228173, 7.20935545, 11.21330363,  9.79292778,
 10.90025378, 9.05082304, 11.20199879, 6.88910567,  9.30820423,
 9.61153253, 11.43368534, 9.40079789, 10.32026921,  13.47026416,
 6.89192738, 6.235617 , 8.90034232, 11.22941928,  11.74463691,
 5.95768691, 7.03970162, 13.81373218, 10.6025767 ,  7.11472428,
 9.79174312, 9.76053889, 7.79903344, 6.68799729,  9.90585749,
 11.44965137, 13.66993969, 10.64587814, 7.94354096,  6.15284893,
 12.10824342, 6.58998767, 13.30297373, 10.51208458,  11.72579271,
 12.04471829, 11.12229286, 8.99611623, 12.88811295,  7.52716061,
 9.87668908, 6.98047296, 10.78660847, 13.1242953 ,  6.96361216,
 8.45629794, 10.14628829, 11.415616 , 11.54044604,  11.03268298,
```

8.23240394, 9.72189777, 6.84813187, 10.36562056, 7.78445346,
10.41993628, 10.64069242, 8.87292438, 11.14556664, 14.41896117,
7.78743617, 7.73127166, 8.96546071, 10.47580321, 10.90739172,
10.66779222, 10.35291713, 9.14229011, 9.12965678, 4.97159883,
7.91128019, 13.00892904, 5.78672977, 8.85571482, 10.18606182,
6.35572872, 11.22132352, 10.94438653, 11.42327403,
7.87818447,
12.05610161, 12.71529149, 6.29985787, 12.10989933, 9.04186013,
10.46489133, 7.83430253, 14.23492422, 9.33090916, 10.31115121,
7.69210658, 7.71989269, 11.13035082, 8.95810013, 10.58713329,
10.74878685, 12.86199511, 11.24491514, 10.19525923, 13.37054132,
6.73543063, 12.99474177, 8.28697398, 9.53487315, 10.68338002,
7.54342948, 7.77697753, 11.25941608, 10.71541139, 6.3385795 ,
8.67664959, 9.81731184, 12.93864733, 10.73639597, 7.00979027,
8.84650669, 10.00304308, 8.70321773, 10.32528579, 13.93131709,
11.62906364, 10.83407728, 10.41279568, 8.3961383 , 15.56524418,
10.88154662, 9.95323659, 10.98052955, 13.87169685, 10.00534277,
14.32722398, 9.93808486, 10.59307046, 10.58414236, 11.65480336,
9.58189099, 9.27683398, 9.49886591, 12.2961644 , 10.48396998,
7.0178077 , 8.58923538, 6.57957064, 6.84757095, 11.205874 ,
9.63153059, 9.28745715, 7.72387254, 6.27337897, 10.72947939,
10.69329745, 10.41916227, 5.95419307, 11.21054361, 15.2180724 ,
8.40420349, 10.62300701, 12.52856505, 9.0790375 , 10.276452 ,
8.03380356, 10.4505996 , 11.2110249 , 8.68842869, 11.93286187,
11.02805785, 8.49175264, 10.17345329, 11.17041799, 9.88121541,
7.04159138, 10.58069464, 9.23592935, 13.7026291 , 9.79853787,
11.96944717, 9.03999342, 10.59908531, 11.44384213, 8.66830822,
7.74373953, 10.17187378, 10.06520008, 11.40538169, 12.31665238,
10.3849862 , 12.43024703, 7.96517043, 11.93197069, 12.62357305,
9.89667041, 7.55267731, 9.93270812, 6.89921735, 11.5618212 ,
9.79877593, 13.00425153, 9.66795514, 10.03852292, 12.48138032,
8.80630391, 12.12199342, 10.5593597 , 8.45983801,
8.11083783,
8.1624298 , 10.4738589 , 7.71881857, 6.5559061 , 12.3937018 ,
7.32257022, 12.78720051, 10.87084556, 10.35476208, 8.84203277,
10.44499388, 9.22826593, 12.68428708, 6.80354359, 11.35542272,
7.06358656, 10.80160915, 11.46965392, 10.80774272, 7.72924176,
10.69935259, 9.78743053, 9.49566126, 9.98398246, 11.33917762,
13.10880133, 12.90385207, 9.90634531, 7.71827817, 11.4001669 ,
6.89864286, 8.53619782, 9.64289444, 7.99996298, 8.80597576,
9.87674646, 11.30692402, 12.04599782, 13.55279359, 10.71756379,
10.67047481, 10.36171127, 9.66359033, 10.06411403, 9.99386576,
10.93030519, 8.29031551, 6.97843067, 9.71591238, 9.37401159,
11.34597617, 11.78166764, 8.37504337, 12.35686448, 10.06609321,
10.90551622, 7.34835826, 12.69807184, 12.30182991, 7.71862635,
10.55050488, 10.21812841, 8.79800197, 6.47395936, 8.91846792,
11.40229901, 12.67644235, 10.91905469, 9.97606935, 11.75366092,

6.07956042, 7.98325282, 10.07621049, 12.74466478, 8.30672896,
4.3174385 , 7.48106846, 9.84008581, 10.38485403, 12.19614147,
15.743999 , 13.38985018, 11.93874745, 10.71884265, 7.26629606,
9.05229315, 10.88643604, 7.81248042, 6.76623825, 7.21792674,
7.66875379, 7.83002983, 9.5336033 , 10.2218477 , 10.13282617,
8.82195623, 7.80958 , 13.45091283, 11.86770763, 11.19663322,
6.38176425, 11.88133255, 11.99253544, 9.2867554 , 7.53114038,
5.78192817, 9.7167695 , 10.88719219, 7.17652712, 12.31611016,
9.23479384, 7.781589 , 11.95975006, 11.43932028, 10.23024873,
7.3025269 , 10.23246188, 9.30683408, 14.93737877, 7.78095861,
7.01845919, 8.08657719, 10.32151628, 7.89645063,
8.01397958,
6.60435981, 8.44284551, 10.84324439, 12.54558913, 8.26890616,
10.38295524, 9.88421585, 9.60780589, 9.95196538, 11.20556396,
10.21594288, 9.85308623, 11.34623389, 11.16655818, 9.72387669,
7.28116384, 14.13229067, 10.04020691, 11.10471024, 6.17045227,
7.4256634 , 9.8018655 , 6.66123323, 9.27615054, 11.93510794,
10.54093859, 11.3362301 , 11.21169938, 7.20349816, 14.56530511,
12.96520458, 7.43592602, 11.54394487, 16.359947 , 11.95600173,
3.92594506, 7.53546622, 11.33606498, 10.25397434, 10.7363397 ,
10.98283913, 11.5141192 , 8.40851152, 13.97113133, 9.88721936,
8.01265833, 11.56019044, 9.60312836, 10.20302896, 10.98670784,
13.34632932, 11.10475114, 7.79814721, 12.08241037, 13.15737043,
10.81474833, 10.30779869, 11.17048413, 10.97059256, 9.12081341,
10.961023 , 9.24906564, 9.99562155, 6.75966813, 11.30127987,
6.26770585, 11.91893484, 12.67969667, 10.88555595, 6.38307545,
8.83801889, 9.44484003, 12.64259672, 9.82884419, 11.70183681,
11.42244419, 10.5153038 , 7.16946931, 9.21556728, 9.90043586,
10.67187749, 13.90846849, 8.54315388, 9.95405706, 6.09881693,
9.43705271, 12.27474647, 11.62282526, 10.71675921, 13.14127277,
10.79666732, 8.66569768, 12.12051218, 12.0272078 , 11.30800928,
13.05229767, 11.81421676, 8.80107235, 10.49842859, 10.33060942,
9.11145308, 7.48954071, 8.43805409, 9.27811851, 8.85816045,
10.03718297, 12.40610793, 10.23422166, 10.94455862, 8.4332166 ,
10.54381626, 7.3117628 , 5.05121283, 7.31447175, 8.6423952 ,
13.86048891, 5.97004721, 11.24024109, 7.74886611, 7.92162251,
11.53443848, 9.53756901, 7.63985485, 11.37217939,
10.13806553,
10.78405117, 12.6264759 , 10.40294961, 9.92060085, 10.39114878,
11.53443078, 10.24192429, 7.61552208, 12.4794433 , 9.36293116,
9.49777584, 12.30746514, 11.49825595, 9.74674497, 11.22210063,
11.91743732, 5.72167291, 9.01769648, 11.29708758, 12.00989866,
9.2657777 , 10.97023963, 7.09810325, 13.07912529, 13.62112447,
8.63542258, 12.8468817 , 11.67704554, 11.63876609, 7.38574522,
9.62706943, 10.42358173, 7.00897596, 3.93585906, 6.81846365,
13.25026631, 8.06496306, 9.30792817, 7.37259729, 8.19594582,
10.56470777, 6.3679716 , 10.15603746, 9.53301265, 8.65109224,

11.16576608, 12.59267106, 6.12709279, 6.4872351 , 11.30852322,
12.10695348, 12.01773779, 11.18675771, 15.9819291 , 9.81796099,
7.13962209, 5.68639473, 7.55715779, 9.83347063, 5.75667486,
5.9372433 , 6.9063169 , 12.09781776, 10.24705163, 9.23563964,
6.0967128 , 10.39479799, 10.61916849, 11.2056428 , 6.59447033,
10.22497832, 9.85587027, 10.68352493, 12.82690877, 12.66157843,
10.71561832, 9.16194229, 12.65971823, 8.88949605, 14.91299876,
10.21867197, 10.17929574, 13.57728247, 7.60462758, 8.62830208,
7.42557186, 7.85798479, 12.01000175, 5.81550741, 11.94421982,
9.93515387, 6.2184606 , 8.27628797, 11.08561786, 6.62400234,
6.8866491 , 10.79694194, 6.45800079, 8.48632924, 9.28204693,
8.84621284, 10.60984767, 10.14424223, 9.68845973, 12.28839202,
11.28358217, 11.45390855, 9.65930025, 12.40489478, 11.762936 ,
6.54691315, 11.67236171, 6.15738336, 10.5281044 , 10.97459361,
12.35021903, 8.04563625, 8.08486895, 6.5162803 , 9.58670506,
13.39575672, 12.42811688, 12.86042357, 9.49798484,
9.08226611,
8.15046127, 12.28727176, 11.29542547, 13.20302439, 12.07197665,
9.17596342, 9.94304769, 11.30273637, 7.25951099, 6.5643887 ,
11.9184521 , 13.45302819, 7.96002145, 9.66264447, 11.79330515,
9.92509393, 11.87409915, 7.91485381, 5.72539104, 7.26357929,
13.25106024, 7.46967669, 12.10000363, 13.71401959, 10.4579484 ,
7.94970996, 12.19214558, 10.17323875, 10.94173941, 13.65298726,
9.72519324, 11.17376316, 10.52840304, 11.32613463, 9.78985771,
10.82186572, 10.82685438, 10.35433658, 9.0879156 , 11.97972749,
9.59802242, 7.88673096, 10.26199378, 6.31337526, 11.915047 ,
7.11726886, 11.16563404, 9.57079682, 7.25444008, 9.14234641,
10.09195393, 12.26321603, 9.21518891, 8.20474263, 10.10413095,
11.20176884, 9.75439287, 10.62803408, 7.45928527, 11.76882693,
9.51951382, 8.97632454, 6.59017358, 4.66819822, 7.18039884,
12.87580932, 10.07597512, 8.51334786, 10.92933545, 12.28647609,
7.53875291, 8.7014604 , 10.98401017, 6.3197358 , 9.42679645,
8.21093349, 8.12780768, 14.6082573 , 12.96985881, 8.97398262,
11.26175384, 7.65452905, 10.73499716, 3.94145048, 10.91673972,
8.68735073, 5.95822639, 9.78432727, 12.60735432, 8.71904108,
8.76919582, 8.91613768, 11.61756357, 11.34178796, 10.21638751,
7.1699912 , 10.13339837, 9.11336647, 10.26884985, 12.47544423,
10.05250502, 9.66370444, 9.60391583, 9.65697043, 7.80696451,
8.51791394, 8.09710006, 8.91558115, 10.2542057 , 12.75629966,
15.90227003, 13.86667808, 6.21982894, 11.67023867, 7.88666797,
7.10762423, 12.08519085, 11.41810551, 11.11293381, 11.12847704,
6.75210361, 9.18255906, 10.46190322, 7.52921426,
11.63338753,
12.1319043 , 12.25088954, 10.04210832, 8.66427214, 8.03325333,
9.5791297 , 10.42225221, 9.99748449, 10.32126435, 13.86311286,
6.53181101, 12.2483629 , 10.17645864, 5.32037694, 12.11288597,
10.62452758, 5.74489148, 11.10614667, 9.36651779, 11.09674777,

```
 8.53191527])
```

#coef

```
rg.coef_
```

```
array([-0.33034681, -0.75080284, 0.25026028, 1.00167015,  0.93545612,
 -1.39455501, -0.0307678 , 1.77821467])
```

#accuracy

```
metrics.r2_score(y_test,rg_pred)
```

```
0.45306390505079364
```

#error

```
np.sqrt(mean_squared_error(y_test,rg_pred))
```

```
2.2508253026332983
```