Assignment -3
Python Programming

Assignment Date	**29 September 2022**
Student Name	**AJAY**
Student Roll Number	**720819106004**
Maximum Marks	**2 Marks**

```
import pandas as pd
```

2.Loading the Dataset

```
df=pd.read_csv(r"C:\Users\Arjun\Downloads\abalone.csv")
df
```

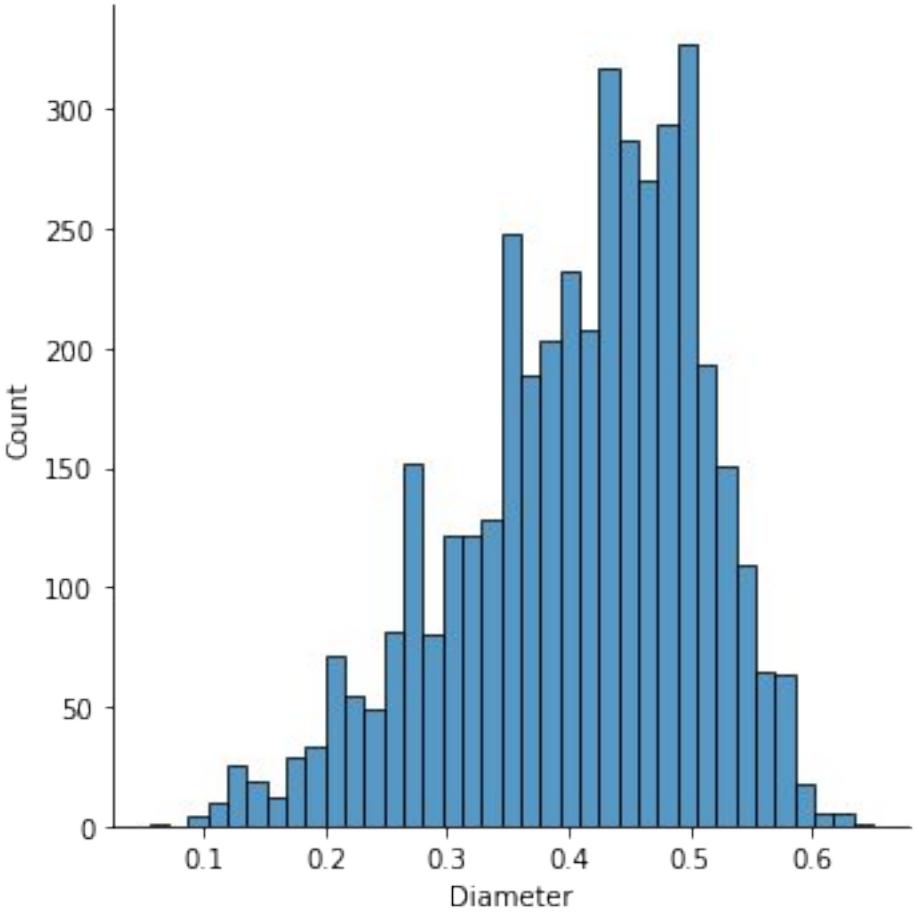
```
     Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0      M   0.455     0.365   0.095        0.5140          0.2245
1      M   0.350     0.265   0.090        0.2255          0.0995
2      F   0.530     0.420   0.135        0.6770          0.2565
3      M   0.440     0.365   0.125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

3.Performing Visualizations.

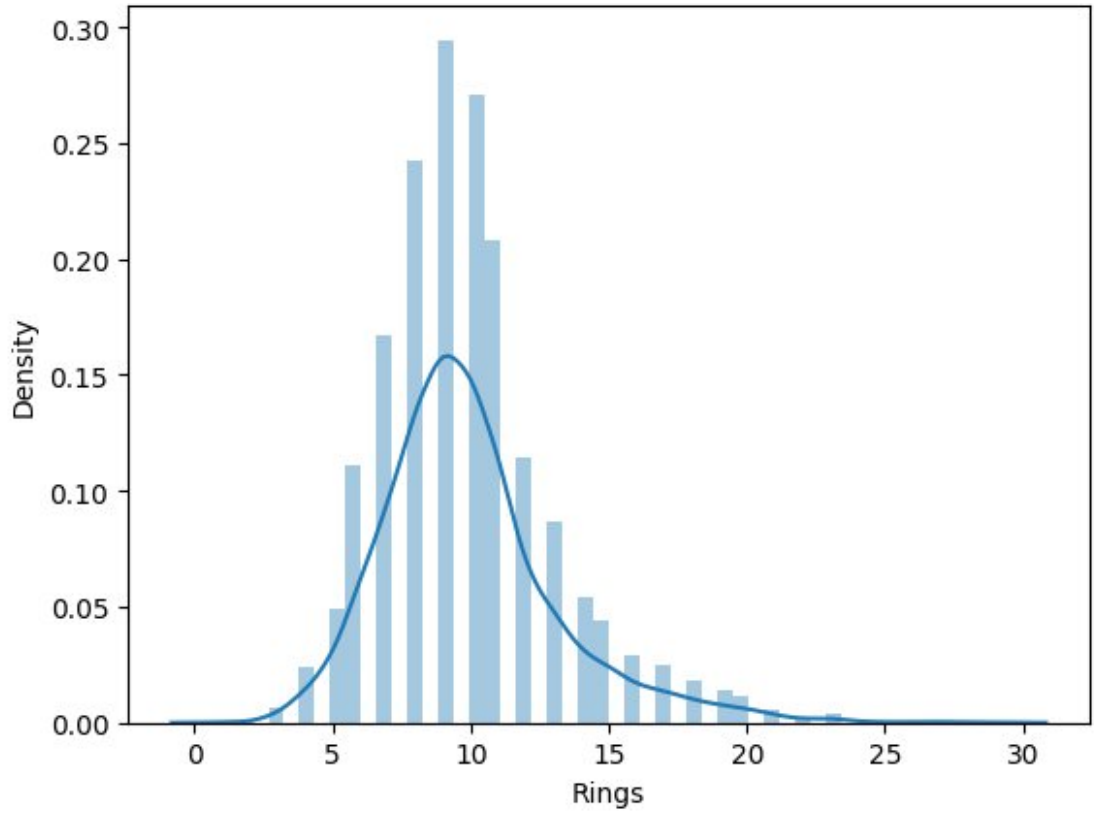
3.1 UNI-variant analysis

```
import seaborn as sns

sns.displot(df.Diameter)

<seaborn.axisgrid.FacetGrid at 0x29f9a84b280>
```

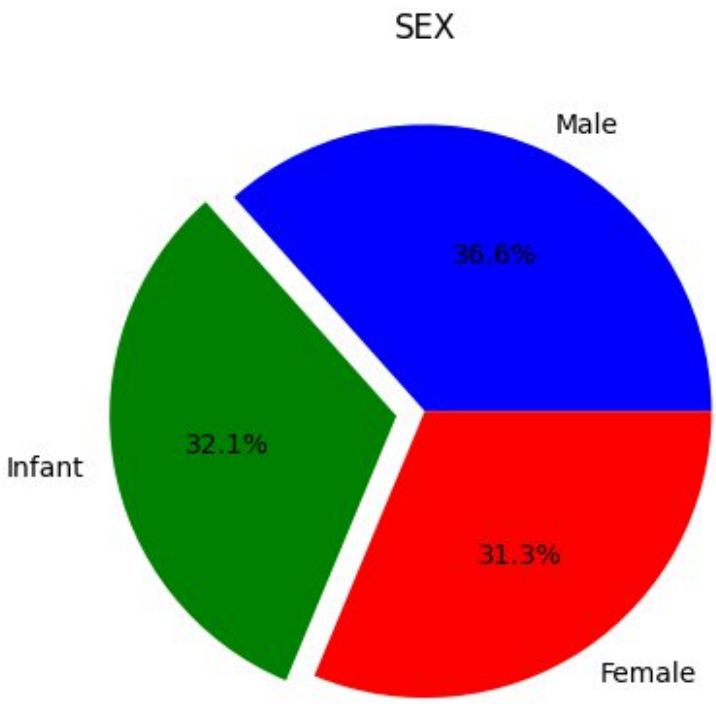



```
sns.distplot(df.Rings)

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-
level function with similar flexibility) or `histplot` (an axes-level
function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='Rings', ylabel='Density'>
```




```
import matplotlib.pyplot as plt
plt.pie(df.Sex.value_counts(),[0,0.1,0],labels=["Male","Infant","Female"],aut
opct="%1.1f%%",colors=["blue","green","red"])
plt.title("SEX")
plt.show()
```

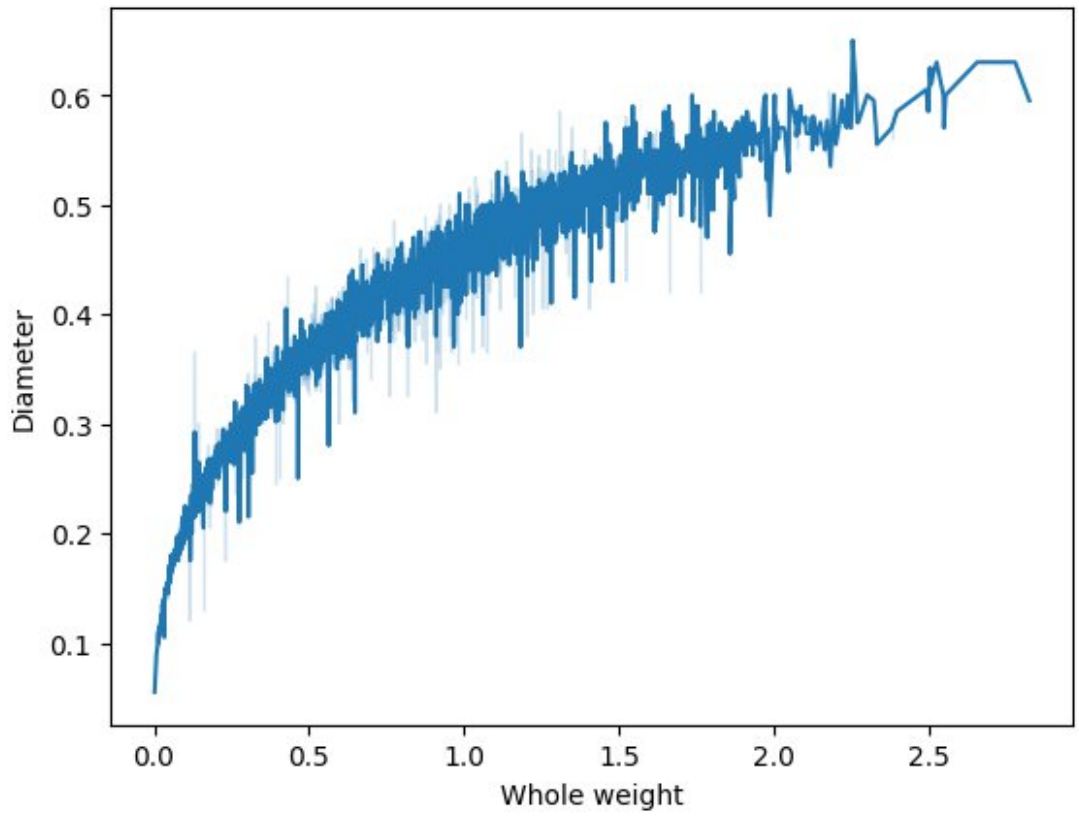


3.2 BI-variant analysis

```
sns.lineplot(df["Whole weight"],df.Diameter)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

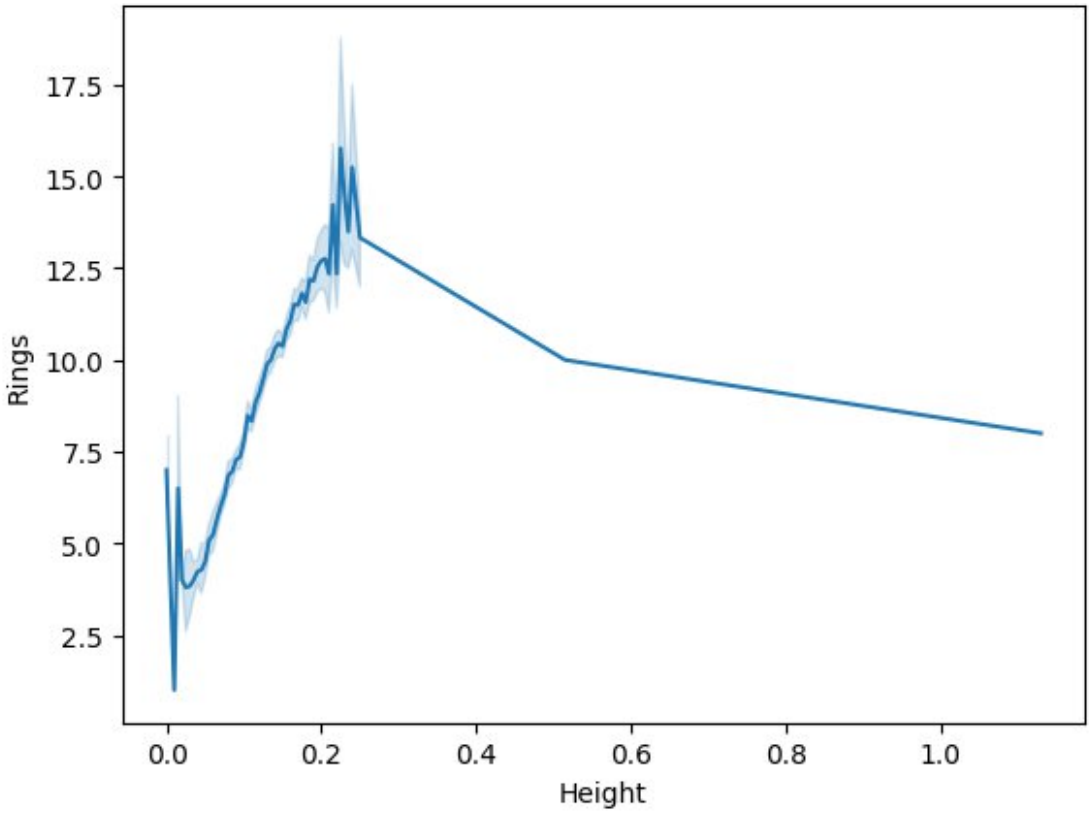
```
<AxesSubplot:xlabel='Whole weight', ylabel='Diameter'>
```




```
sns.lineplot(df.Height,df.Rings)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Height', ylabel='Rings'>
```

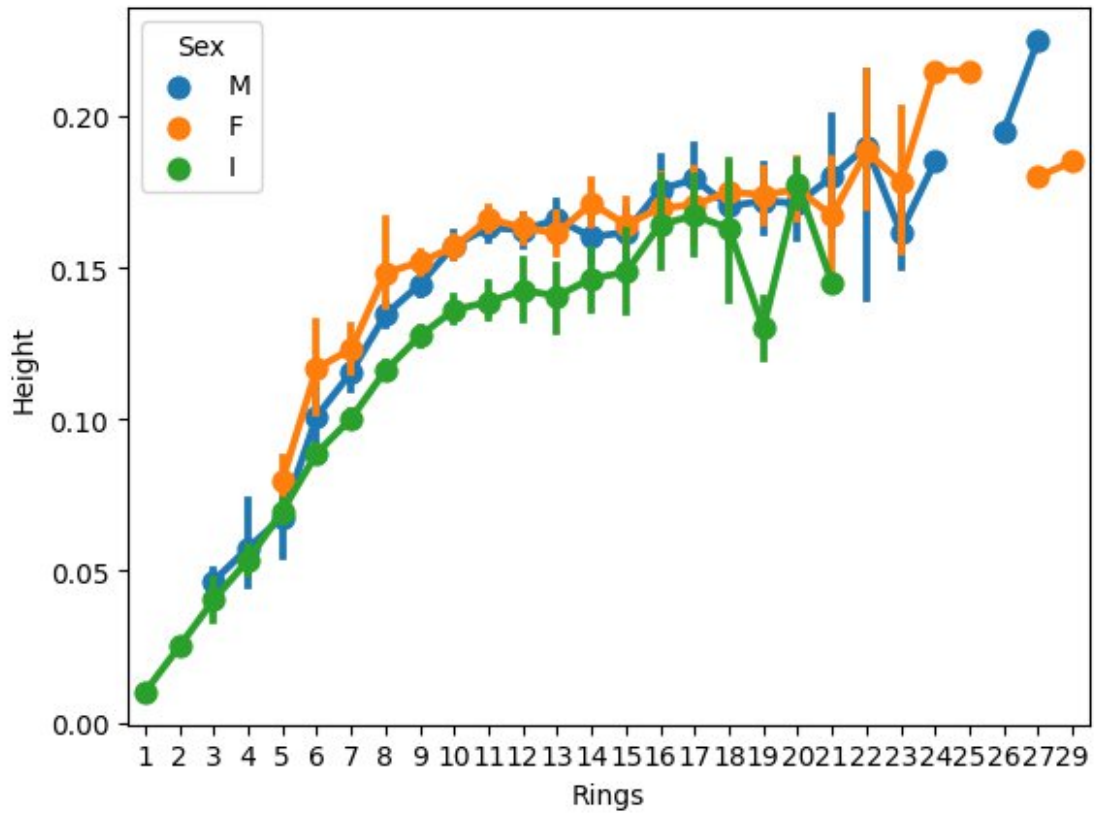


3.3 MULTI-variate analysis

```
sns.pointplot(df.Rings ,df.Height ,data=df,hue='Sex')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Rings', ylabel='Height'>
```



4. Perform descriptive statistics on the dataset

```
df.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
```

5. Check for Missing values and deal with them.

```
df.isnull().sum()
```

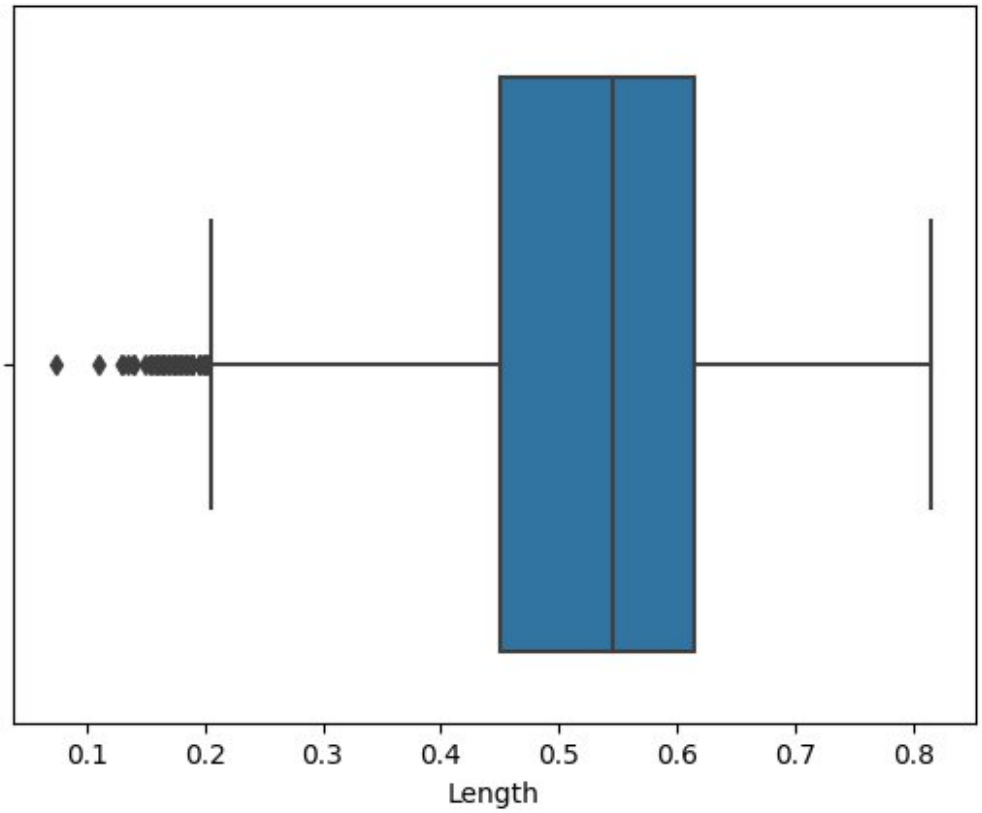
```
Sex               0
Length            0
Diameter          0
Height            0
Whole weight      0
Shucked weight    0
Viscera weight    0
Shell weight      0
Rings             0
dtype: int64
```

6. Find the outliers and replaing the outliers

```
sns.boxplot(df.Length)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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



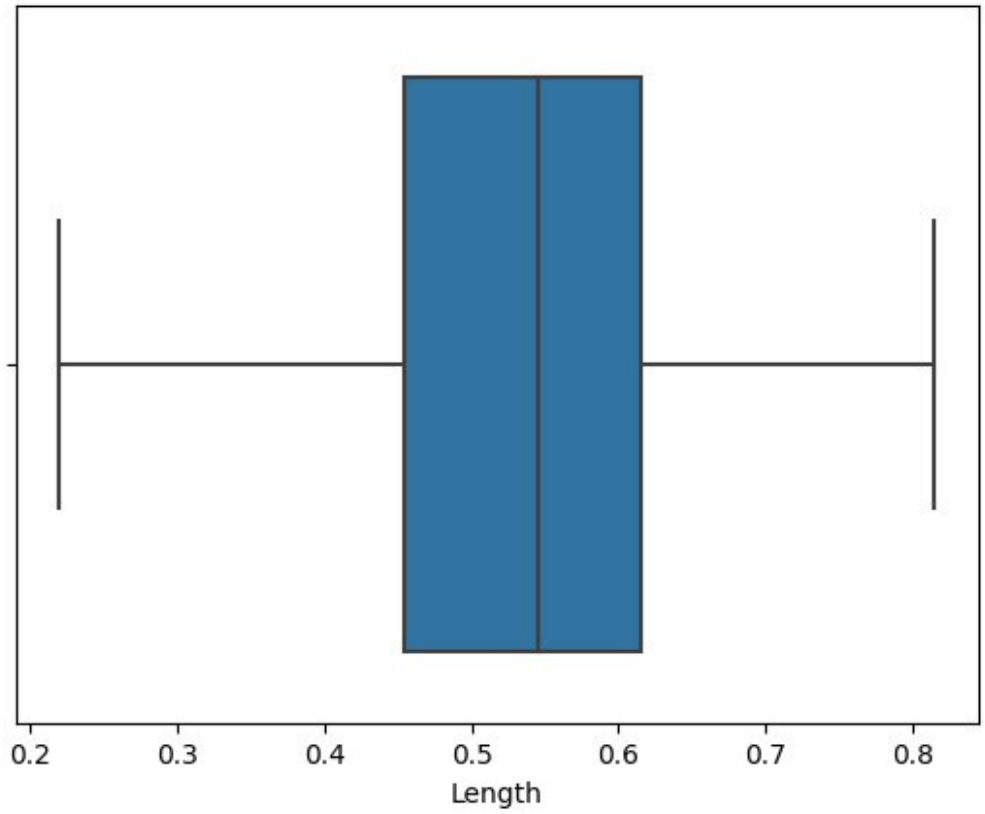

```
a01=df.Length.quantile(0.015)
```

```
import numpy as np
df["Length"]=np.where(df.Length<=a01,df.Length.median(),df.Length)
```

```
sns.boxplot(df.Length)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

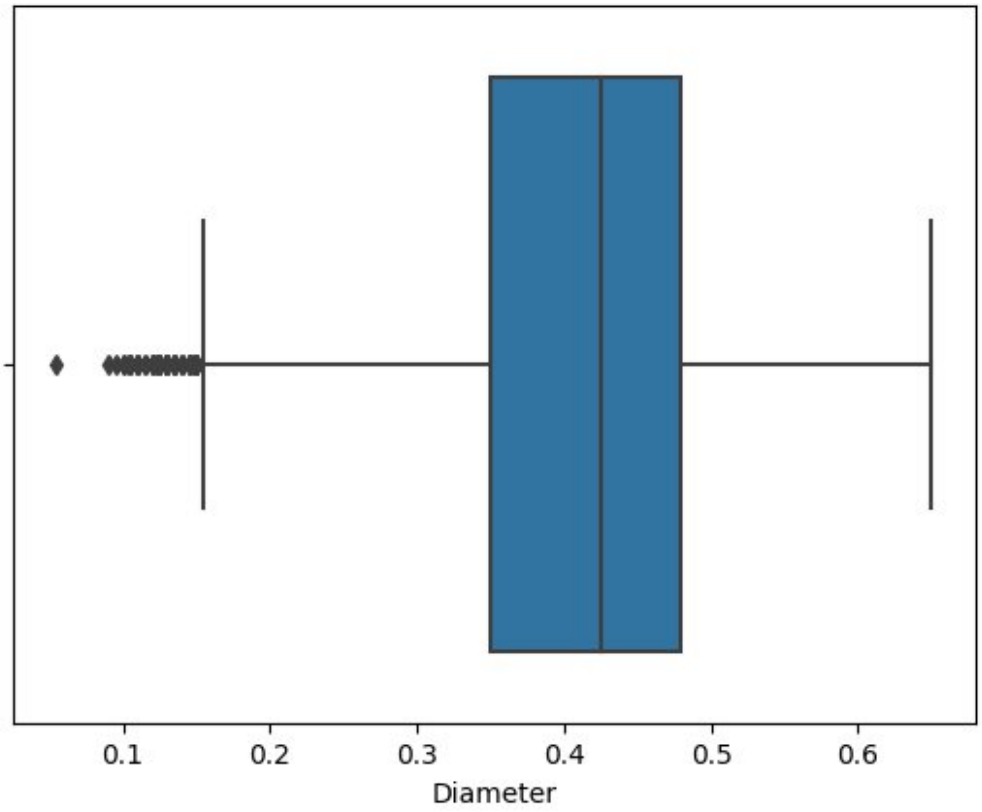
```
<AxesSubplot:xlabel='Length'>
```




```
sns.boxplot(df.Diameter)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

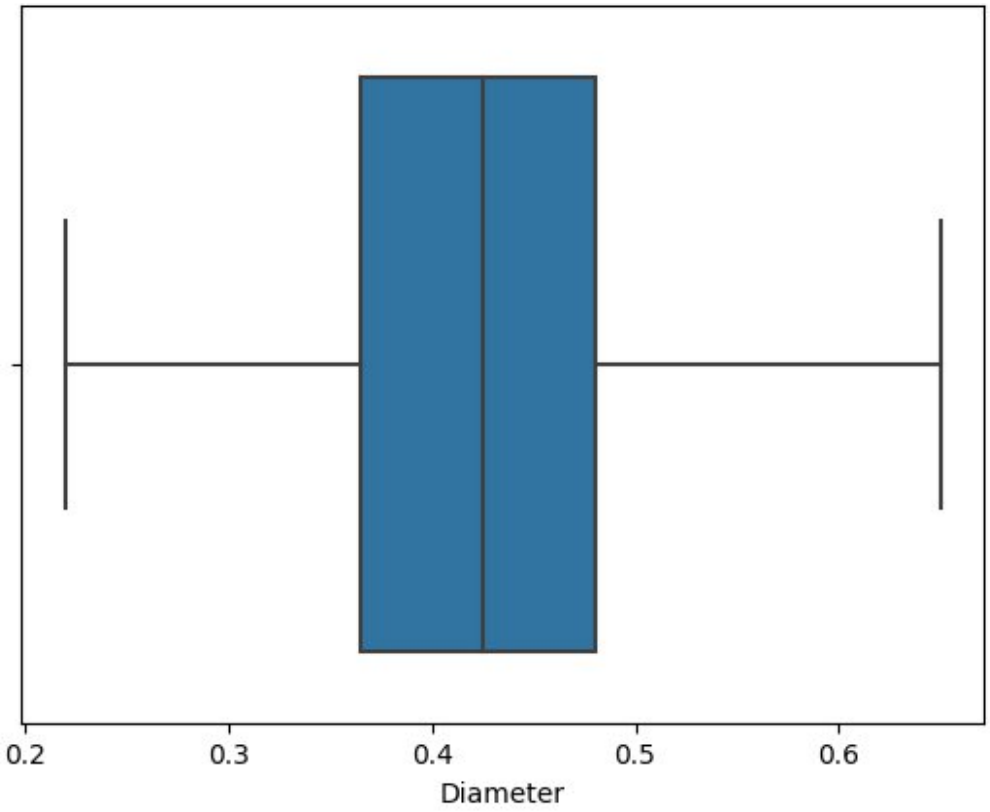



```
a02=df.Length.quantile(0.01)
df["Diameter"]=np.where(df.Diameter<=a01,df.Diameter.median(),df.Diameter)
```

```
sns.boxplot(df.Diameter)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

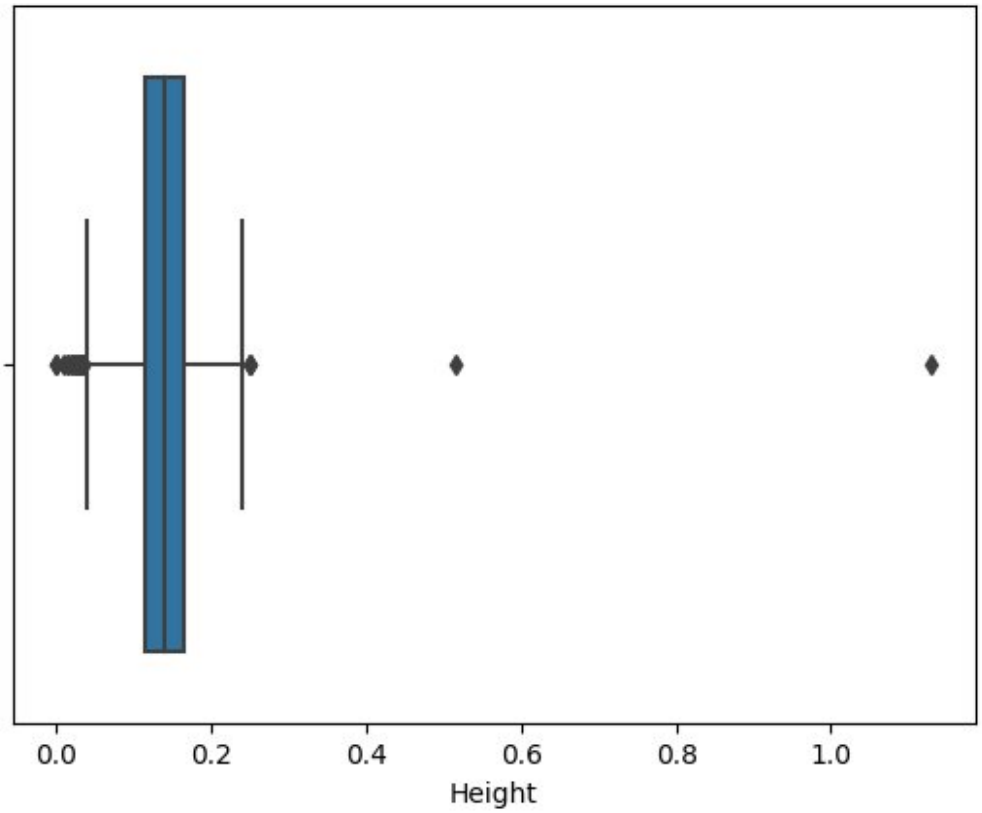
```
<AxesSubplot:xlabel='Diameter'>
```




```
sns.boxplot(df.Height)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

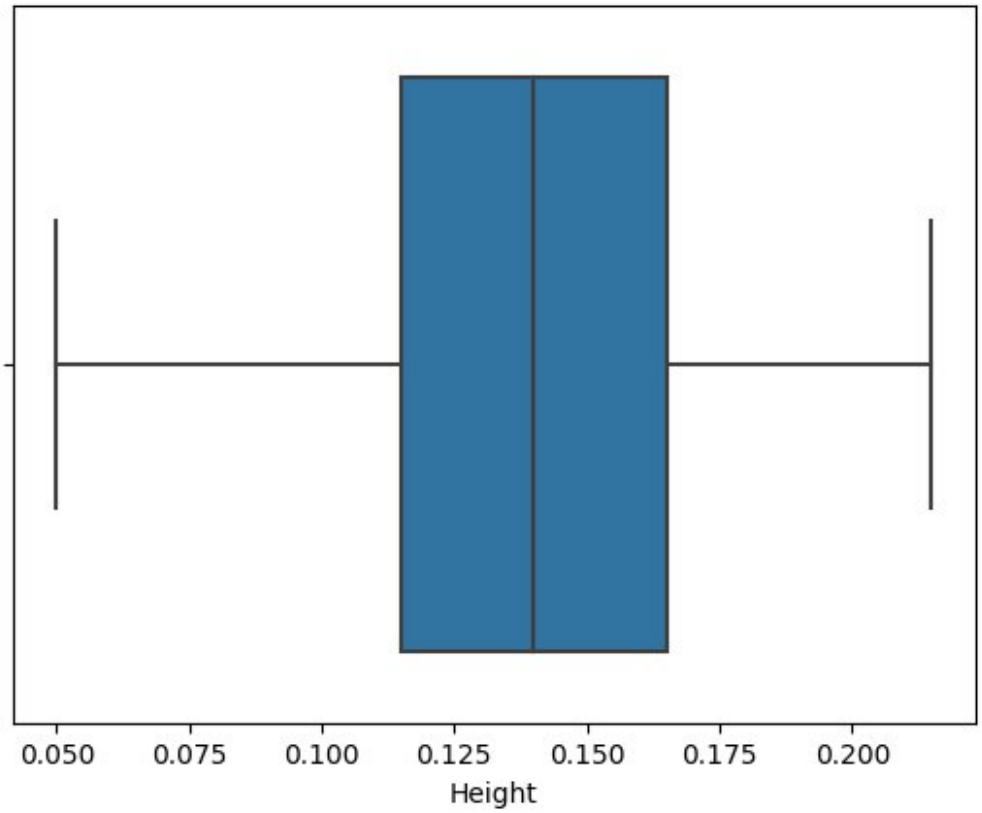



```
a09=df["Height"].quantile(0.01)
df["Height"]=np.where(df.Height<=a09,df.Height.median(),df.Height)
a10=df.Height.quantile(0.99)
df["Height"]=np.where(df.Height>=a10,df.Height.median(),df.Height)
```

```
sns.boxplot(df.Height)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

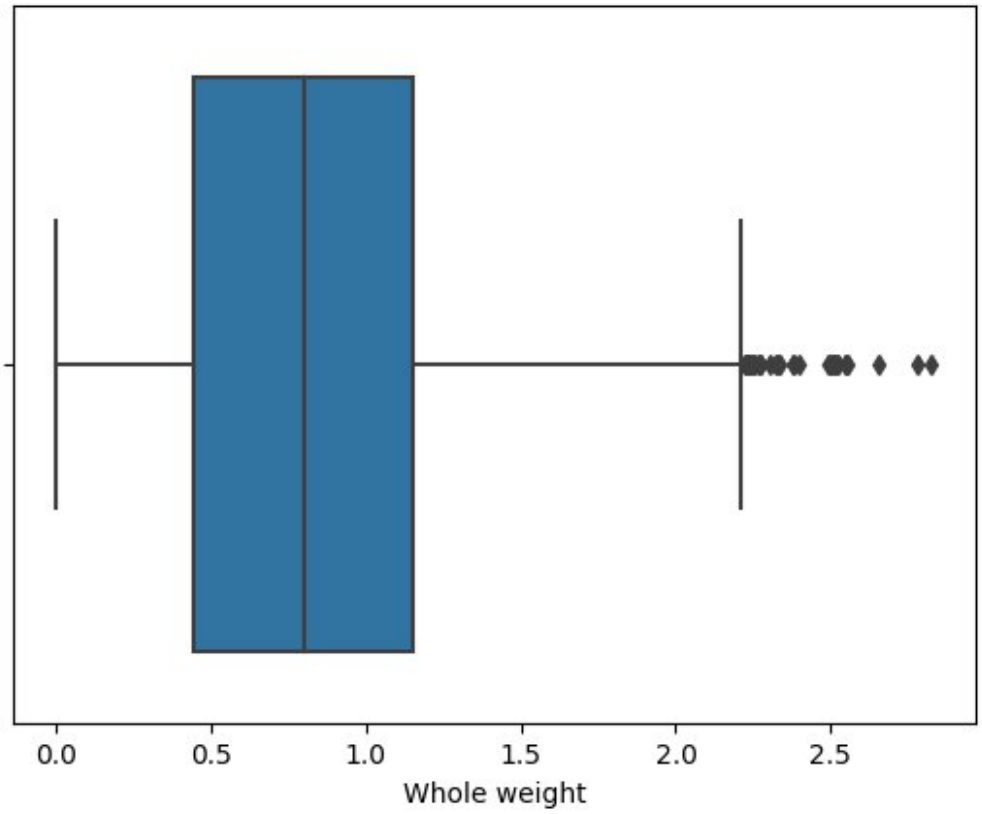
```
<AxesSubplot:xlabel='Height'>
```




```
sns.boxplot(df["Whole weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

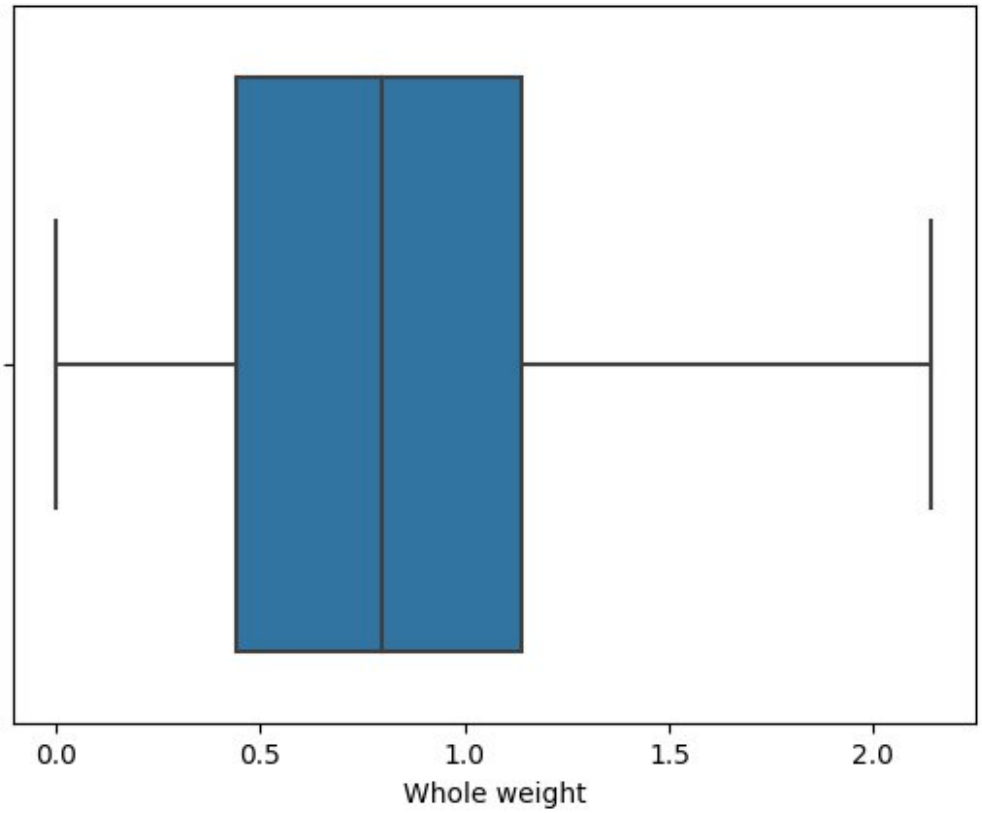



```
a03=df["Whole weight"].quantile(0.99)
df["Whole weight"]=np.where(df["Whole weight"]>=a03,df["Whole
weight"].median(),df["Whole weight"])
```

```
sns.boxplot(df["Whole weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

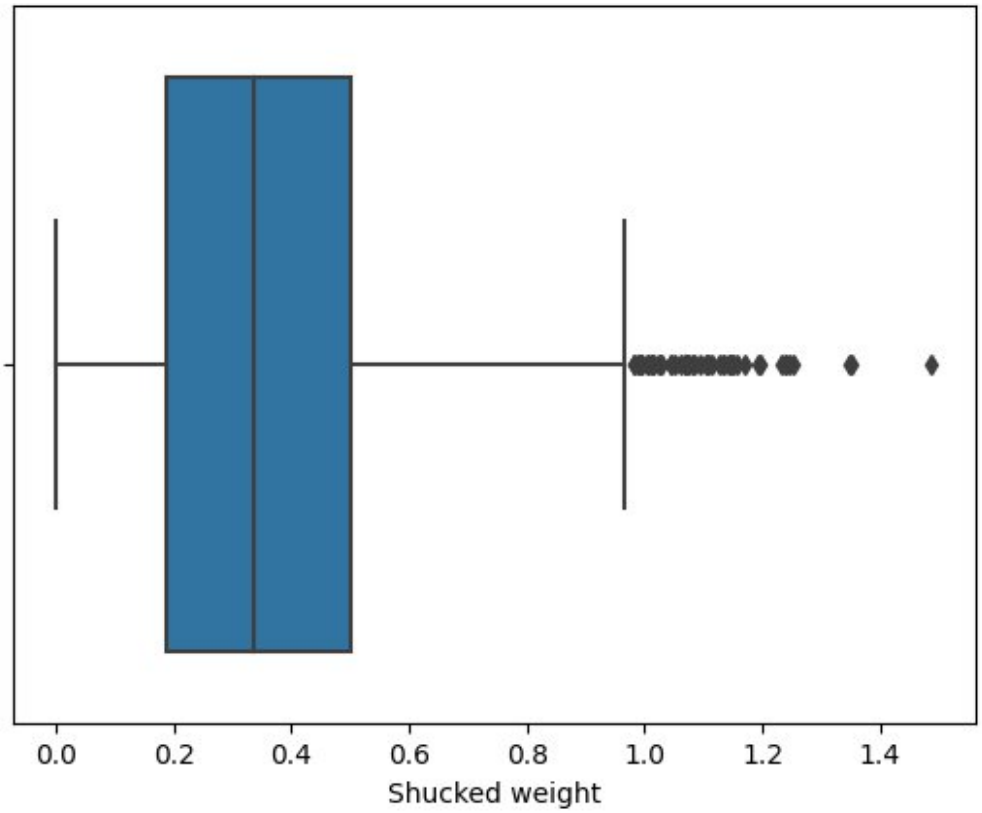
```
<AxesSubplot:xlabel='Whole weight'>
```




```
sns.boxplot(df["Shucked weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

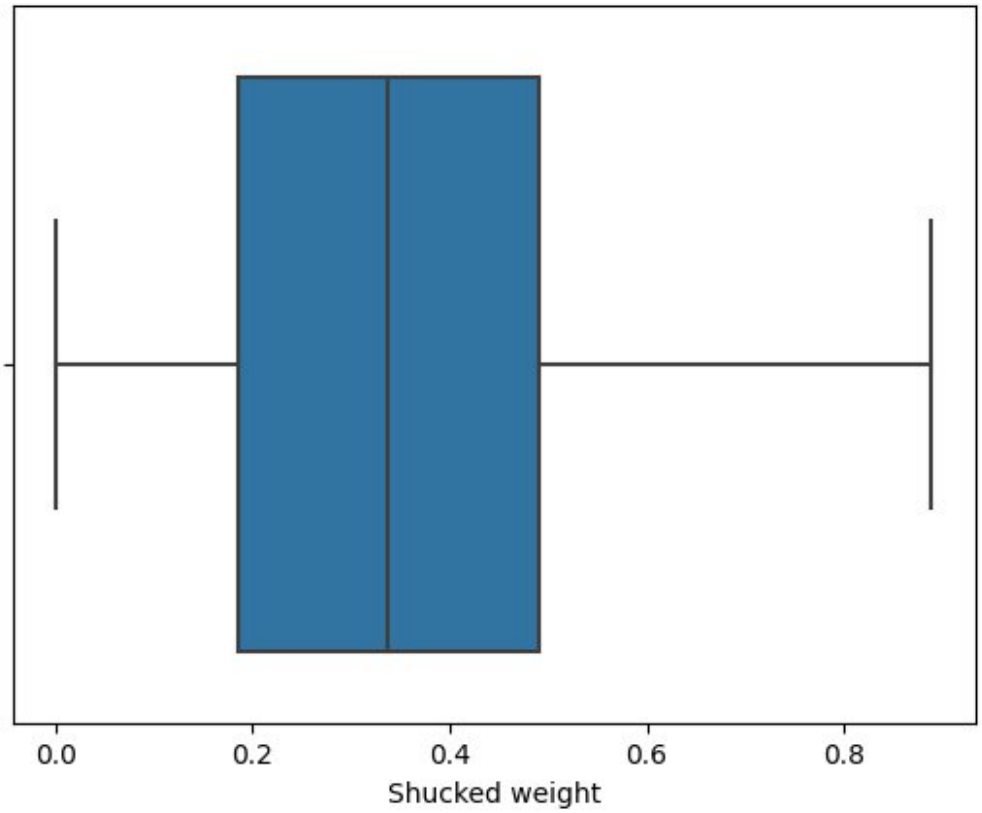



```
a04=df["Shucked weight"].quantile(0.98)
df["Shucked weight"]=np.where(df["Shucked weight"]>=a04,df["Shucked
weight"].median(),df["Shucked weight"])
```

```
sns.boxplot(df["Shucked weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

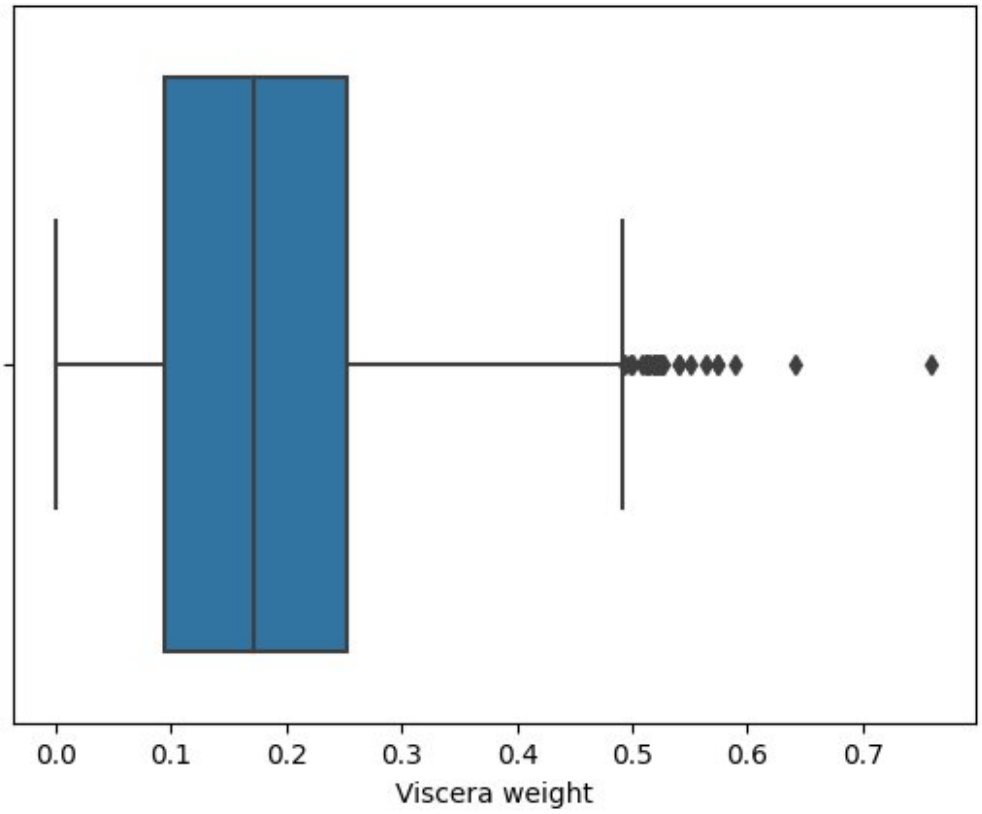
```
<AxesSubplot:xlabel='Shucked weight'>
```




```
sns.boxplot(df["Viscera weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

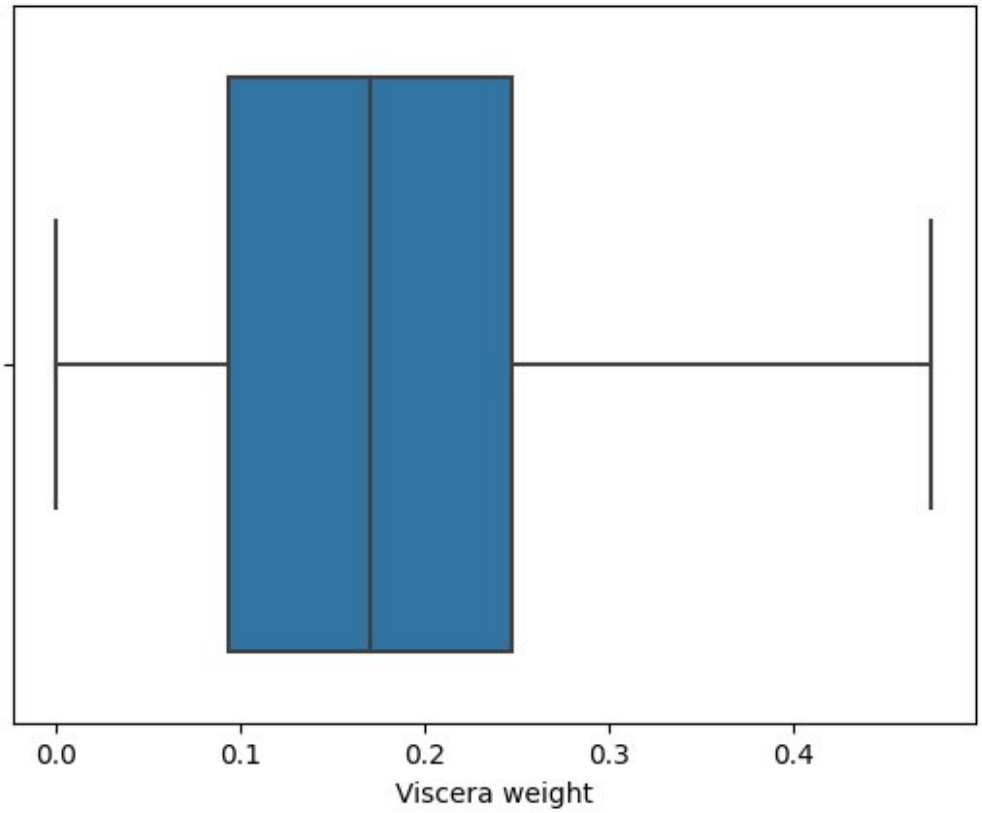



```
a05=df["Viscera weight"].quantile(0.99)
df["Viscera weight"]=np.where(df["Viscera weight"]>=a05,df["Viscera
weight"].median(),df["Viscera weight"])
```

```
sns.boxplot(df["Viscera weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

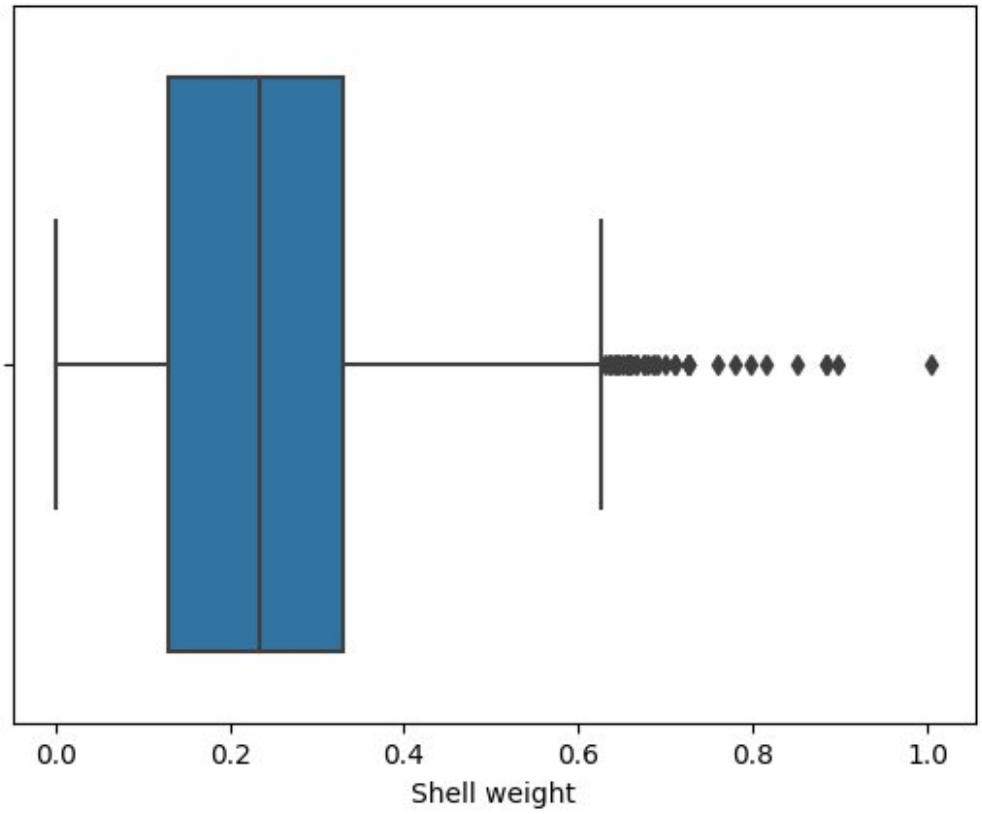
```
<AxesSubplot:xlabel='Viscera weight'>
```




```
sns.boxplot(df["Shell weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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

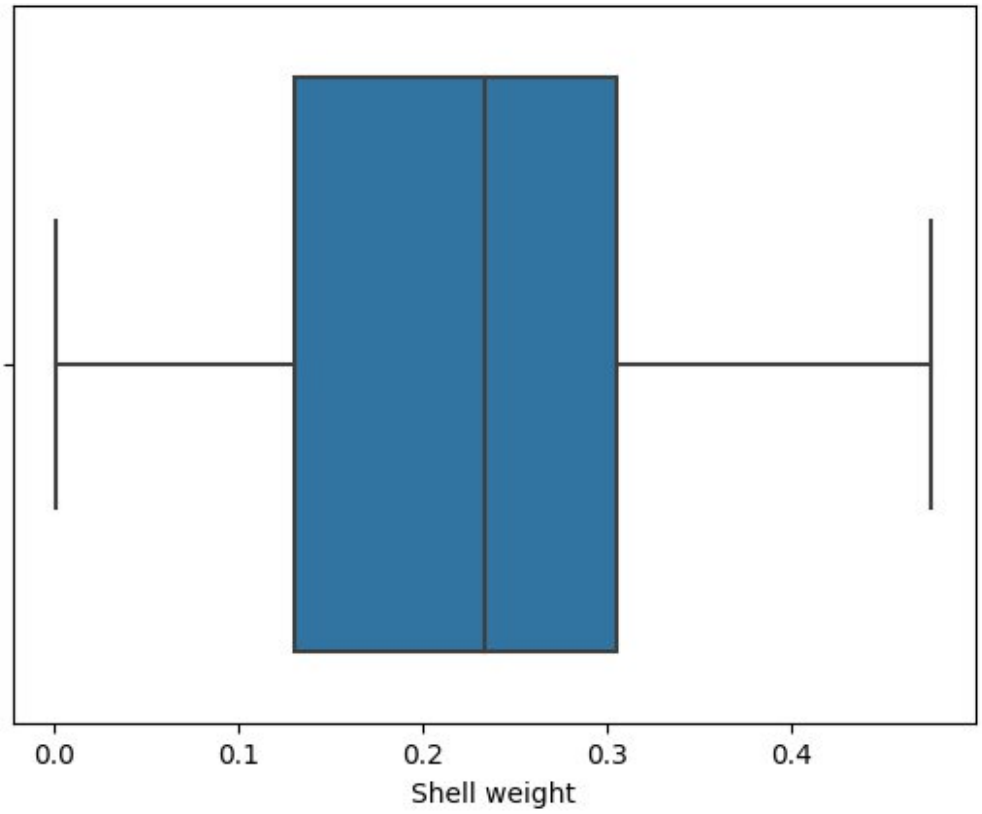



```
a06=df["Shell weight"].quantile(0.99)
df["Shell weight"]=np.where(df["Shell weight"]>=a05,df["Shell
weight"].median(),df["Shell weight"])
```

```
sns.boxplot(df["Shell weight"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

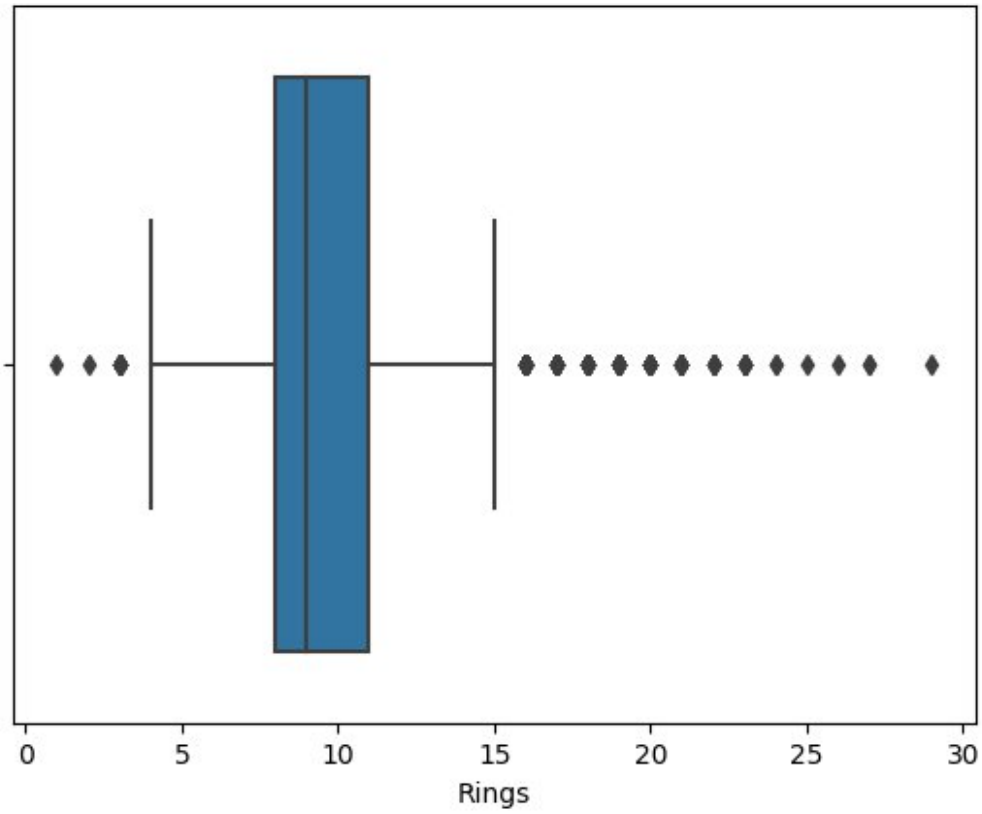
```
<AxesSubplot:xlabel='Shell weight'>
```




```
sns.boxplot(df["Rings"])
```

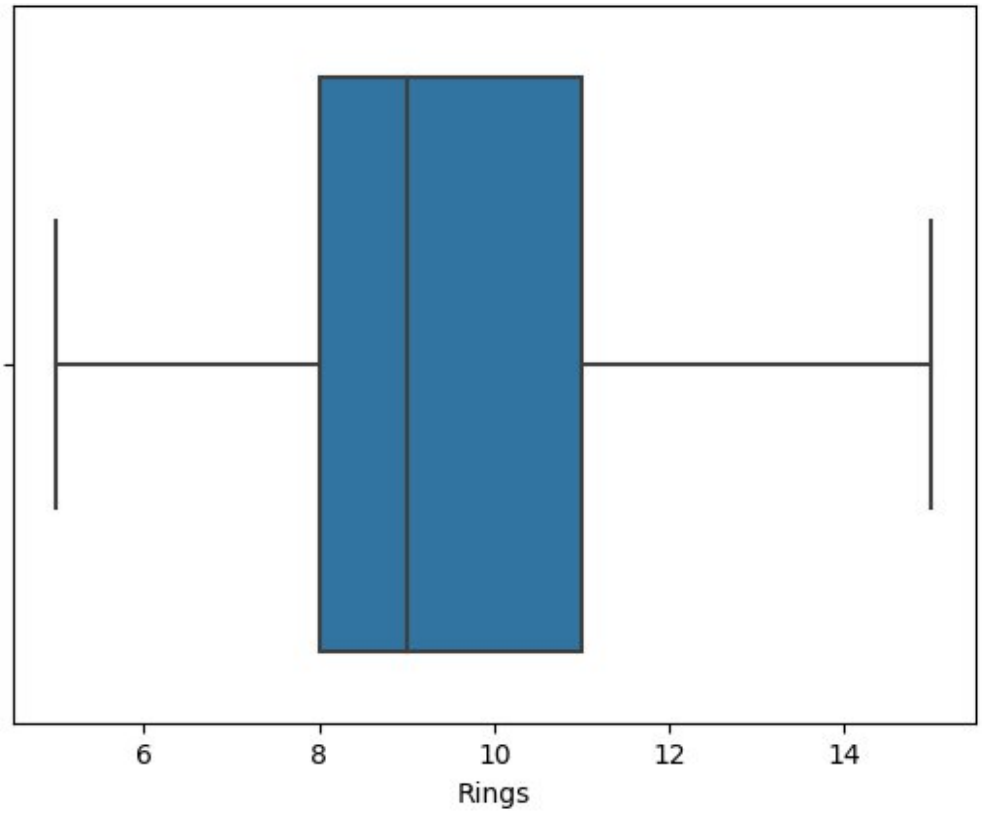
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

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




```
a07=df["Rings"].quantile(0.01)
df["Rings"]=np.where(df.Rings<=a07,df.Rings.median(),df.Rings)

a08=df.Rings.quantile(0.95)
df["Rings"]=np.where(df.Rings>=a08,df.Rings.median(),df.Rings)

sns.boxplot(df["Rings"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(

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



Adding the target variable

```
df["Age"]=df.Rings+1.5
```

```
df
```

```
     Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0      M   0.455     0.365   0.095        0.5140          0.2245
1      M   0.350     0.265   0.090        0.2255          0.0995
2      F   0.530     0.420   0.135        0.6770          0.2565
3      M   0.440     0.365   0.125        0.5160          0.2155
4      I   0.330     0.255   0.080        0.2050          0.0895
...   ..     ...       ...     ...           ...             ...
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          0.3360

      Viscera weight  Shell weight  Rings   Age
0             0.1010        0.1500   15.0  16.5
1             0.0485        0.0700    7.0   8.5
2             0.1415        0.2100    9.0  10.5
```

```
3             0.1140        0.1550   10.0  11.5
4             0.0395        0.0550    7.0   8.5
...              ...           ...    ...   ...
4172          0.2390        0.2490   11.0  12.5
4173          0.2145        0.2605   10.0  11.5
4174          0.2875        0.3080    9.0  10.5
4175          0.2610        0.2960   10.0  11.5
4176          0.3765        0.2340   12.0  13.5

[4177 rows x 10 columns]
```

7.Check for Categorical columns and perform encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.Sex=le.fit_transform(df.Sex)

df.head()
```

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

   Viscera weight  Shell weight  Rings   Age
0          0.1010         0.150   15.0  16.5
1          0.0485         0.070    7.0   8.5
2          0.1415         0.210    9.0  10.5
3          0.1140         0.155   10.0  11.5
4          0.0395         0.055    7.0   8.5
```

8.Split the data into dependent and independent variables

```
x=df.drop(["Age"],axis="columns")
x
```

```
      Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0       2   0.455     0.365   0.095        0.5140          0.2245
1       2   0.350     0.265   0.090        0.2255          0.0995
2       0   0.530     0.420   0.135        0.6770          0.2565
3       2   0.440     0.365   0.125        0.5160          0.2155
4       1   0.330     0.255   0.080        0.2050          0.0895
...   ...     ...       ...     ...           ...             ...
4172    0   0.565     0.450   0.165        0.8870          0.3700
```

```
4173    2   0.590     0.440   0.135        0.9660          0.4390
4174    2   0.600     0.475   0.205        1.1760          0.5255
4175    0   0.625     0.485   0.150        1.0945          0.5310
4176    2   0.710     0.555   0.195        1.9485          0.3360

      Viscera weight  Shell weight  Rings
0              0.1010        0.1500   15.0
1              0.0485        0.0700    7.0
2              0.1415        0.2100    9.0
3              0.1140        0.1550   10.0
4              0.0395        0.0550    7.0
...               ...           ...    ...
4172           0.2390        0.2490   11.0
4173           0.2145        0.2605   10.0
4174           0.2875        0.3080    9.0
4175           0.2610        0.2960   10.0
4176           0.3765        0.2340   12.0

[4177 rows x 9 columns]
```

```
y=df.Age
y
```

```
0       16.5
1        8.5
2       10.5
3       11.5
4        8.5
        ...
4172    12.5
4173    11.5
4174    10.5
4175    11.5
4176    13.5
Name: Age, Length: 4177, dtype: float64
```

9.Scaling the independent variables

```
from sklearn.preprocessing import scale
x_scaled=pd.DataFrame(scale(x),columns=x.columns)
```

10. Split the data into training and testing

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.3,rando
m_state=10)
```

11.Build the model

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

12.Train the mdoel

```
lr.fit(x_scaled,y)
```

```
LinearRegression()
```

13.Test the model

```
lr.predict(x_test)
```

```
array([ 8.5, 11.5,  9.5, ..., 12.5, 10.5, 12.5])
```

```
lr.score(x_train,y_train)
```

```
1.0
```

```
lr.score(x_test,y_test)
```

```
1.0
```