In [13]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
```

In [14]:
```python
df=pd.read_csv('abalone.csv')
df.head()
```

Out[14]:

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

In [15]:
```python
df.shape
```

Out[15]:
```
(4177, 9)
```

In [16]:
```python
df.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
```

In [17]:
```python
df.isnull().any()
```

Out[17]:
```
Sex               False
Length            False
Diameter          False
Height            False
Whole weight      False
Shucked weight    False
Viscera weight    False
Shell weight      False
Rings             False
dtype: bool
```

In [18]:
```python
df.describe()
```

Out[18]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	8.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	9.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	11.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000

In [19]:
```python
df.Sex.value_counts()
```

Out[19]:
```
M    1528
I    1342
F    1307
Name: Sex, dtype: int64
```

In [20]:
```python
df.Diameter.value_counts()
```

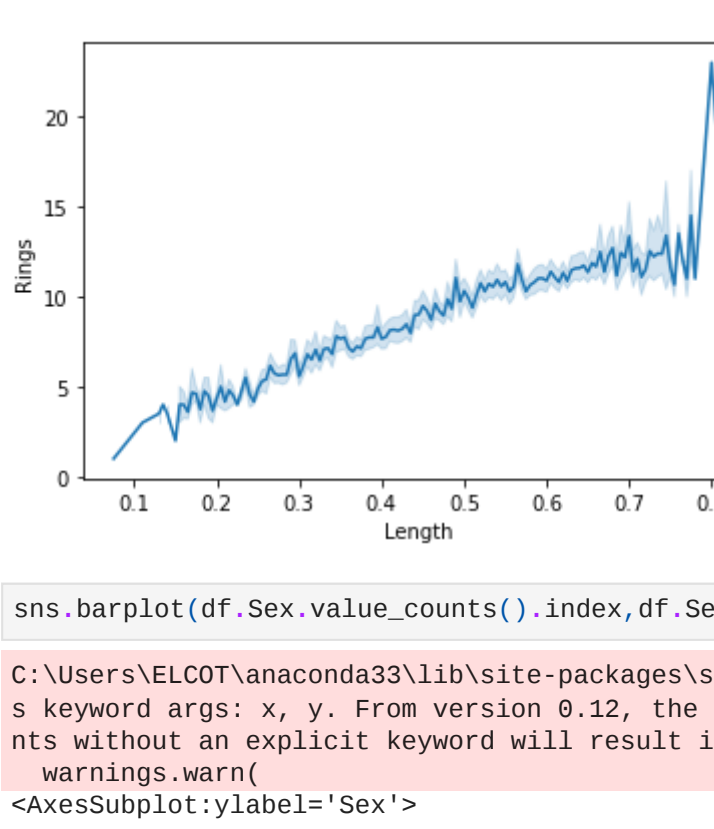
Out[20]:
```
0.450    139
0.475    120
0.400    111
0.500    110
0.470    100
        ... 
0.610      1
0.650      1
0.620      1
0.095      1
0.615      1
Name: Diameter, Length: 111, dtype: int64
```

In [21]:
```python
df.Rings.value_counts()
```

Out[21]:
```
9     689
10    634
8     568
11    487
7     391
12    267
6     259
13    203
14    126
5     115
15    103
16     67
17     58
4      57
18     42
19     32
20     26
3      15
21     14
23      9
22      6
27      2
24      2
1       1
26      1
29      1
2       1
25      1
Name: Rings, dtype: int64
```

In [22]:
```python
sns.displot(df.Length)
```

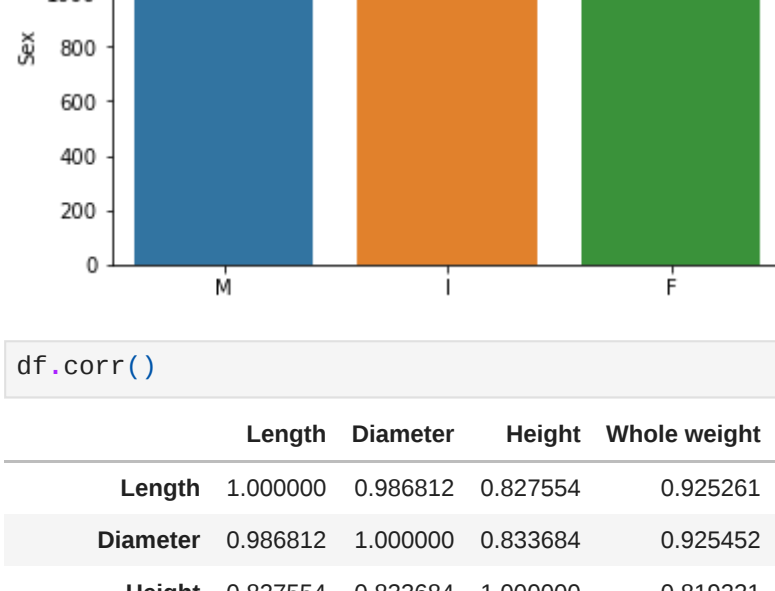
Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x218aee00d90>
```



In [23]:
```python
sns.lineplot(df.Length,df.Rings)
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other argume
nts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

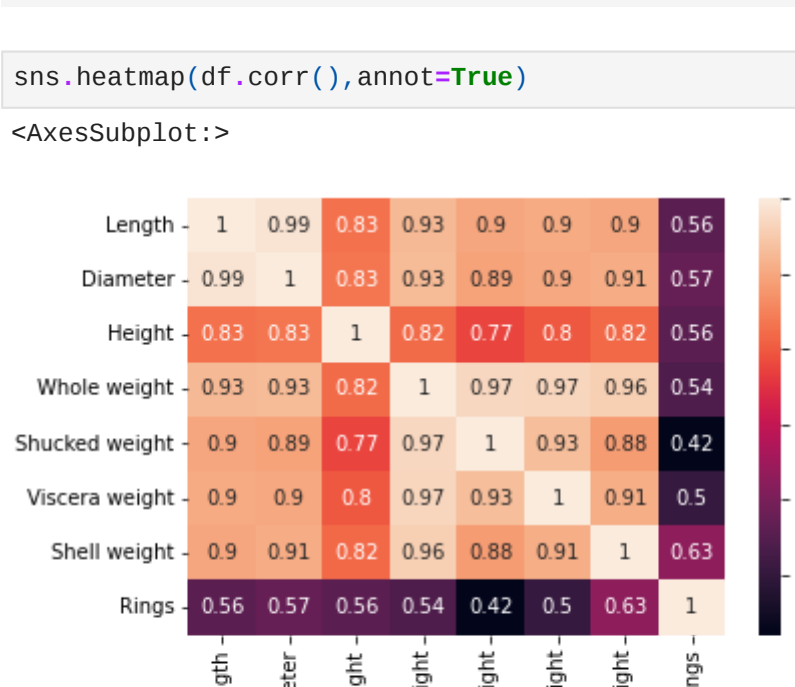
Out[23]:
```
<AxesSubplot:xlabel='Length', ylabel='Rings'>
```



In [24]:
```python
sns.barplot(df.Sex.value_counts().index,df.Sex.value_counts())
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables a
s keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other argume
nts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[24]:
```
<AxesSubplot:ylabel='Sex'>
```



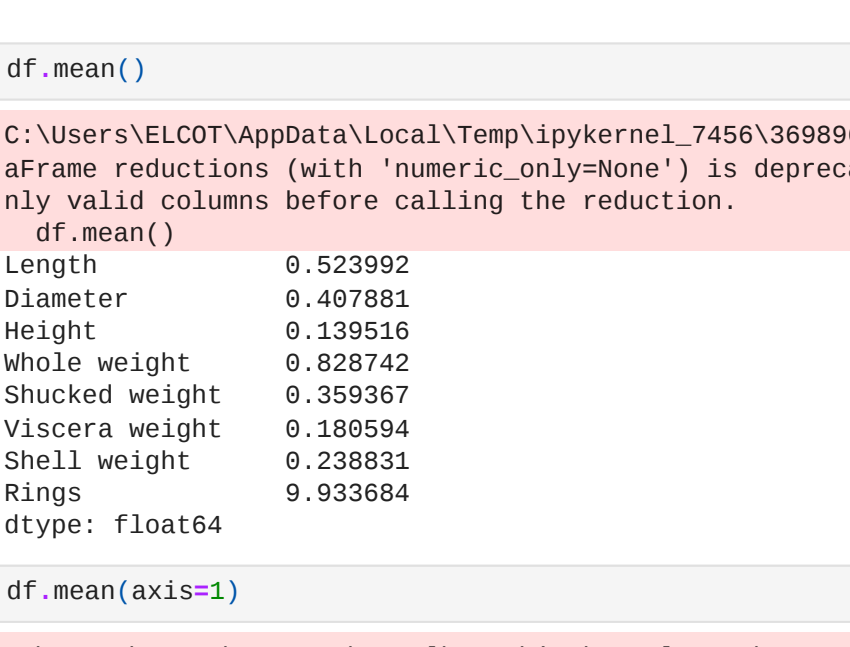
In [25]:
```python
df.corr()
```

Out[25]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
Length	1.000000	0.986812	0.827554	0.925261	0.897914	0.903018	0.897706	0.556720
Diameter	0.986812	1.000000	0.833684	0.925452	0.893162	0.899724	0.905330	0.574660
Height	0.827554	0.833684	1.000000	0.819221	0.774972	0.798319	0.817338	0.557467
Whole weight	0.925261	0.925452	0.819221	1.000000	0.969405	0.966375	0.955355	0.540390
Shucked weight	0.897914	0.893162	0.774972	0.969405	1.000000	0.931961	0.882617	0.420884
Viscera weight	0.903018	0.899724	0.798319	0.966375	0.931961	1.000000	0.907656	0.503819
Shell weight	0.897706	0.905330	0.817338	0.955355	0.882617	0.907656	1.000000	0.627574
Rings	0.556720	0.574660	0.557467	0.540390	0.420884	0.503819	0.627574	1.000000

In [26]:
```python
sns.heatmap(df.corr(),annot=True)
```

Out[26]:
```
<AxesSubplot:>
```



In [27]:
```python
df.head()
```

Out[27]:

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

In [28]:
```python
df.mean()
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_7456\3698961737.py:1: FutureWarning: Dropping of nuisance columns in Dat
aFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select o
nly valid columns before calling the reduction.
  df.mean()
```

Out[28]:
```
Length            0.523992
Diameter          0.407881
Height            0.139516
Whole weight      0.828742
Shucked weight    0.359367
Viscera weight    0.180594
Shell weight      0.238831
Rings             9.933684
dtype: float64
```

In [29]:
```python
df.mean(axis=1)
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_7456\3676274908.py:1: FutureWarning: Dropping of nuisance columns in Dat
aFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select o
nly valid columns before calling the reduction.
  df.mean(axis=1)
```

Out[29]:
```
0       2.113062
1       1.018563
2       1.421250
3       1.491313
4       1.006750
          ...   
4172    1.740625
4173    1.630625
4174    1.572125
4175    1.680312
4176    2.153188
Length: 4177, dtype: float64
```

In [30]:
```python
df.median()
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_7456\530051474.py:1: FutureWarning: Dropping of nuisance columns in Data
Frame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select on
ly valid columns before calling the reduction.
  df.median()
```

Out[30]:
```
Length            0.5450
Diameter          0.4250
Height            0.1400
Whole weight      0.7995
Shucked weight    0.3360
Viscera weight    0.1710
Shell weight      0.2340
Rings             9.0000
dtype: float64
```

In [31]:
```python
df.mode()
```

Out[31]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.550	0.45	0.15	0.2225	0.175	0.1715	0.275	9.0
1	NaN	0.625	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [32]:
```python
df["Rings"].std()
```

Out[32]:
```
3.2241690320681133
```

In [33]:
```python
df["Rings"].var()
```

Out[33]:
```
10.395265947347035
```

In [34]:
```python
df["Rings"].skew()
```

Out[34]:
```
1.114101898355677
```

In [35]:
```python
df["Rings"].kurt()
```

Out[35]:
```
2.3306874268535847
```

In [36]:
```python
null=pd.isnull(df)
null.head()
```

Out[36]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False

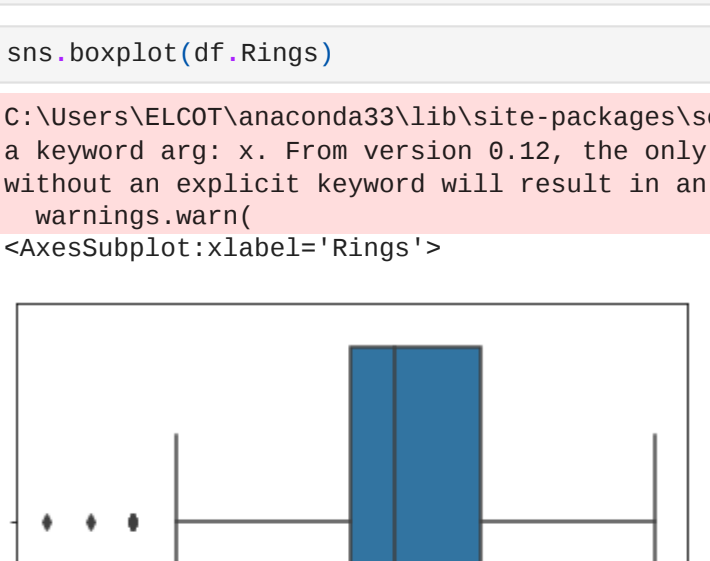
In [37]:
```python
pd.isnull(df).sum().sum()
```

Out[37]:
```
0
```

In [38]:
```python
sns.boxplot(df.Rings)
```

```
C:\Users\ELCOT\anaconda33\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(
```

Out[38]:
```
<AxesSubplot:xlabel='Rings'>
```



In [39]:
```python
q1=df.Rings.quantile(0.25)
q3=df.Rings.quantile(0.75)
```

In [40]:
```python
IQR=q3-q1
```

In [41]:
```python
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

In [42]:
```python
upper_limit
```

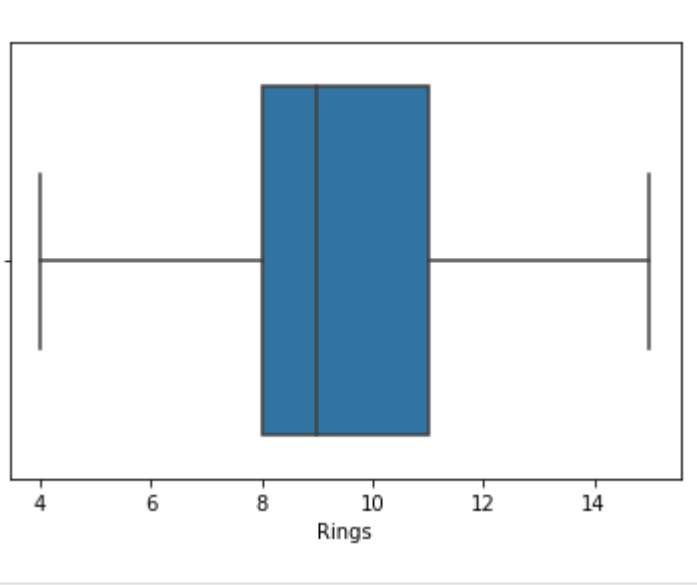
Out[42]:
```
15.5
```

In [43]:
```python
df=df[df.Rings<upper_limit]
```

In [44]:
```python
sns.boxplot(df.Rings)
```

```
C:\Users\ELCOT\anaconda33\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(
```

Out[44]:
```
<AxesSubplot:xlabel='Rings'>
```



In [45]:
```python
lower_limit
```

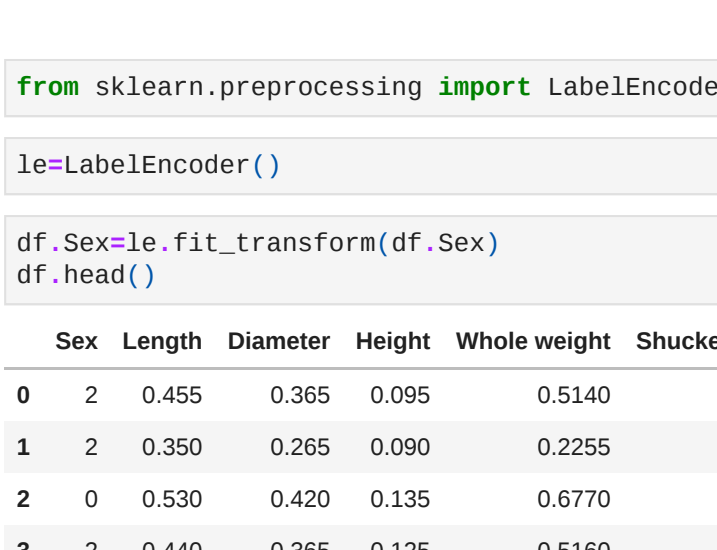
Out[45]:
```
3.5
```

In [46]:
```python
df=df[df.Rings>lower_limit]
```

In [47]:
```python
sns.boxplot(df.Rings)
```

```
C:\Users\ELCOT\anaconda33\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(
```

Out[47]:
```
<AxesSubplot:xlabel='Rings'>
```



In [48]:
```python
df.shape
```

Out[48]:
```
(3899, 9)
```

In [49]:
```python
df.head()
```

Out[49]:

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

In [53]:
```python
from sklearn.preprocessing import LabelEncoder
```

In [54]:
```python
le=LabelEncoder()
```

In [55]:
```python
df.Sex=le.fit_transform(df.Sex)
df.head()
```

Out[55]:

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

In [59]:
```python
y=df['Rings']
y
```

Out[59]:
```
0       15
1        7
2        9
3       10
4        7
        ..
4172    11
4173    10
4174     9
4175    10
4176    12
Name: Rings, Length: 3899, dtype: int64
```

In [60]:
```python
X=df.drop(columns=['Rings'],axis=1)
X.head()
```

Out[60]:

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

In [61]:
```python
from sklearn.preprocessing import scale
```

In [62]:
```python
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()
```

Out[62]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
0	1.157932	-0.551043	-0.404229	-1.035557	-0.608006	-0.585646	-0.697589	-0.604476
1	1.157932	-1.433200	-1.423098	-1.156858	-1.206406	-1.146009	-1.179895	-1.213621
2	-1.299758	0.079070	0.156149	-0.065146	-0.269916	-0.442193	-0.325524	-0.147618
3	1.157932	-0.677065	-0.404229	-0.307749	-0.603858	-0.625992	-0.578161	-0.566405
4	-0.070913	-1.601230	-1.524985	-1.399461	-1.248927	-1.190838	-1.262576	-1.327835

In [63]:
```python
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,random_state=0)
```

In [64]:
```python
X_train.shape
```

Out[64]:
```
(2729, 8)
```

In [65]:
```python
y_train.shape
```

Out[65]:
```
(2729,)
```

In [66]:
```python
X_test.shape
```

Out[66]:
```
(1170, 8)
```

In [67]:
```python
y_test.shape
```

Out[67]:
```
(1170,)
```

In [72]:
```python
from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier(max_depth=3)
```

In [73]:
```python
model.fit(X_train,y_train)
```

Out[73]:
```
DecisionTreeClassifier(max_depth=3)
```

In [74]:
```python
test_pred=model.predict(X_test)
test_pred
```

Out[74]:
```
array([11,  8,  6, ..., 11, 10,  6], dtype=int64)
```

In [75]:
```python
train_pred=model.predict(X_train)
```

In [76]:
```python
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
```

In [77]:
```python
print('Training Accuracy:',accuracy_score(y_train,train_pred))
```

```
Training Accuracy: 0.3023085379259802
```

In [78]:
```python
print('Testing Accuracy:',accuracy_score(y_test,test_pred))
```

```
Testing Accuracy: 0.28034188034188035
```

In [79]:
```python
pd.crosstab(y_test,test_pred)
```

Out[79]:

col_0	4	6	7	8	9	10	11
Rings							
4	9	7	0	0	0	0	0
5	6	25	8	0	0	0	0
6	0	22	47	10	0	2	0
7	1	13	60	42	3	0	0
8	0	6	30	64	51	9	3
9	0	4	13	54	79	44	7
10	0	0	7	21	58	68	25
11	0	1	6	20	41	50	26
12	0	0	4	11	28	24	14
13	0	0	1	8	20	21	12
14	0	0	1	3	17	11	12
15	0	0	0	5	8	18	10

In [81]:
```python
print(classification_report(y_test,test_pred))
```

```
              precision    recall  f1-score   support

           4       0.56      0.56      0.56        16
           5       0.00      0.00      0.00        39
           6       0.28      0.27      0.28        81
           7       0.34      0.50      0.41       119
           8       0.27      0.39      0.32       163
           9       0.26      0.39      0.31       201
          10       0.28      0.38      0.32       179
          11       0.24      0.18      0.21       144
          12       0.00      0.00      0.00        81
          13       0.00      0.00      0.00        62
          14       0.00      0.00      0.00        44
          15       0.00      0.00      0.00        41

    accuracy                           0.28      1170
   macro avg       0.19      0.22      0.20      1170
weighted avg       0.22      0.28      0.24      1170
```

```
C:\Users\ELCOT\anaconda33\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisi
on and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ELCOT\anaconda33\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisi
on and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ELCOT\anaconda33\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisi
on and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]:

In [ ]: