In [ ]:
```
import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import stats
```

In [ ]:
```
df = pd.read_csv("/content/abalone.csv")
```

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

Out[ ]:

	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 [ ]:
```
df.tail()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
4172	F	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	M	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	M	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	F	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	M	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

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

Out[ ]:

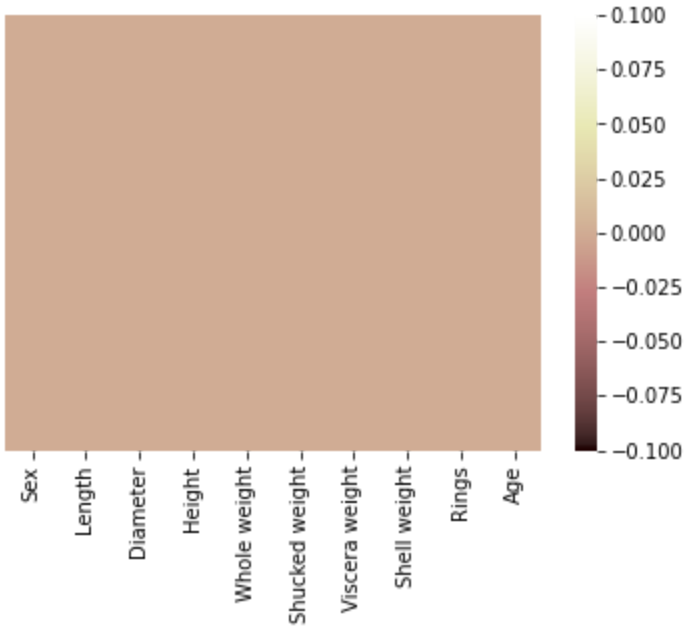
	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell we
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005

In [ ]:
```
df.isnull().sum()
```

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

In [ ]:
```
sns.heatmap(df.isnull(),yticklabels=False,cmap='pink')
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f57bf3ff290>
```



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

Out[ ]:

	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 [ ]:
```
df['Sex'].value_counts()
```

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

In [ ]:
```
df['Sex'].unique()
```

Out[ ]:
```
array(['M', 'F', 'I'], dtype=object)
```

In [ ]:
```
df['Sex'] = df['Sex'].map({'M': 0, 'I': 1, 'F':2})
```

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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	0	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 [ ]:
```
df.tail()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
4172	2	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	0	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	0	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	2	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	0	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

ADDING AGE COLUMN

In [ ]:
```
df['Age'] = df['Rings'] + 2.5
```

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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	17.5
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	9.5
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	11.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	12.5
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	9.5

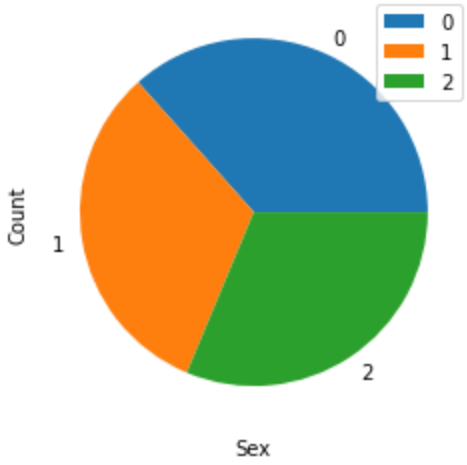
In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weig
ht',
       'Viscera weight', 'Shell weight', 'Rings', 'Age'],
      dtype='object')
```

Data visualization

In [ ]:
```
df['Sex'].value_counts().plot(kind='pie')
plt.legend()
plt.xlabel('Sex')
plt.ylabel('Count')
```

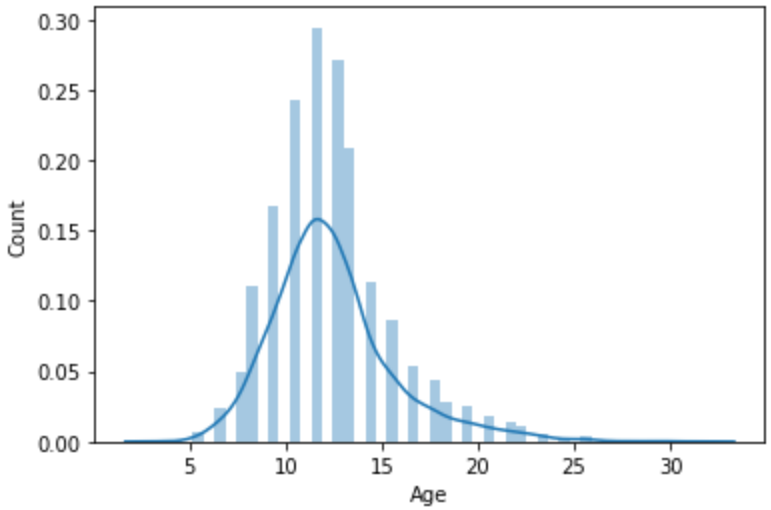
Out[ ]:
```
Text(0, 0.5, 'Count')
```



In [ ]:
```
sns.distplot(df['Age'])
plt.xlabel('Age')
plt.ylabel('Count')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function fo
r histograms).
  warnings.warn(msg, FutureWarning)
```

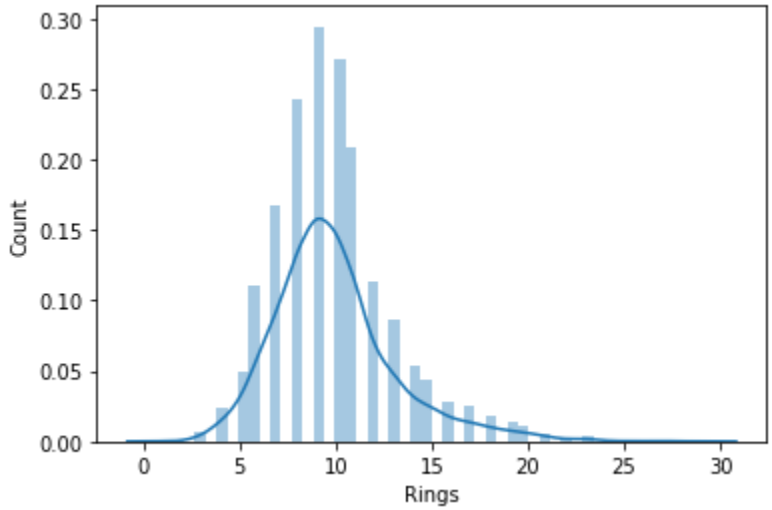
Out[ ]: Text(0, 0.5, 'Count')



In [ ]:
```
sns.distplot(df['Rings'])
plt.xlabel('Rings')
plt.ylabel('Count')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: Futur
eWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function fo
r histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: Text(0, 0.5, 'Count')



In [ ]:

Bi-variate analysis

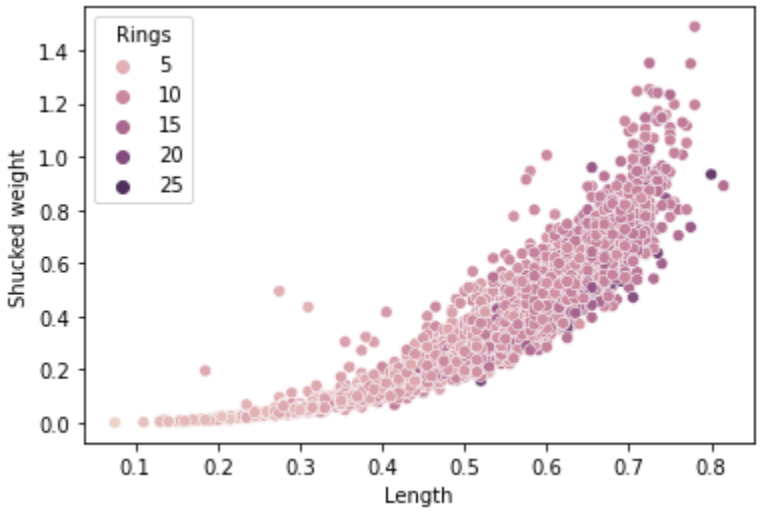
In [ ]:
```
df.head()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	17.5
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	9.5
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	11.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	12.5
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	9.5

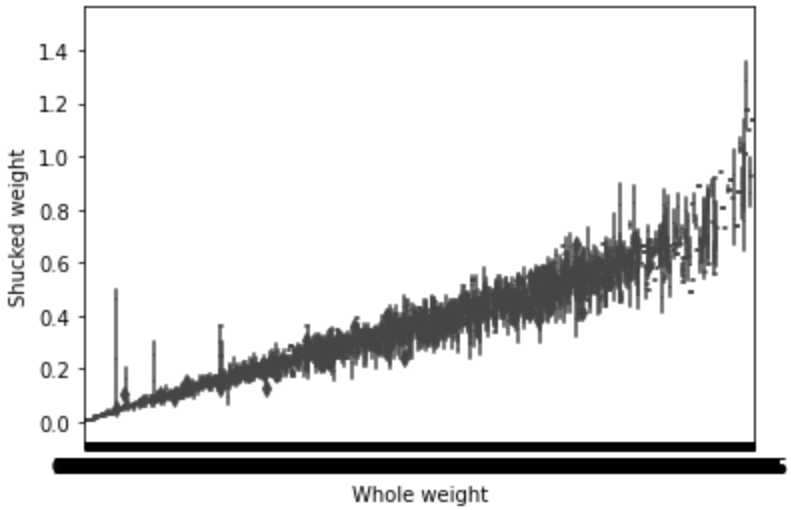
In [ ]:
```
sns.scatterplot(data=df, x='Length', y='Shucked weight', hue='Rings',)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f57befe3ed0>
```



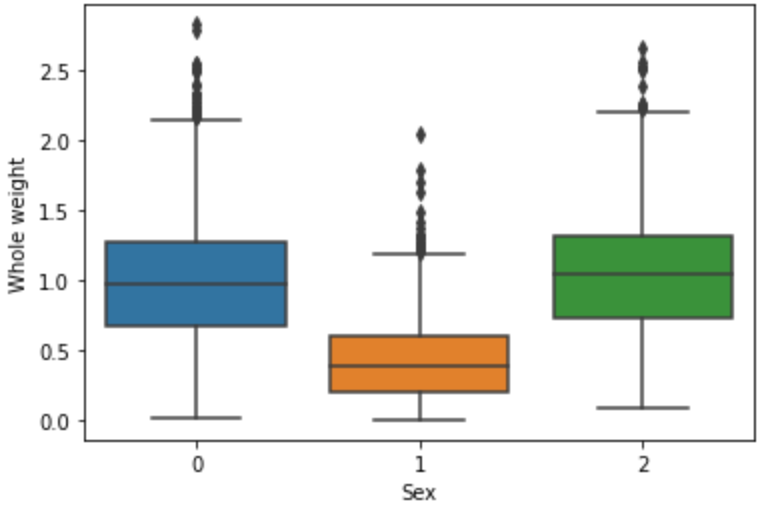
In [ ]:
```
sns.boxplot(data=df, x='Whole weight', y='Shucked weight')
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f57bef71090>



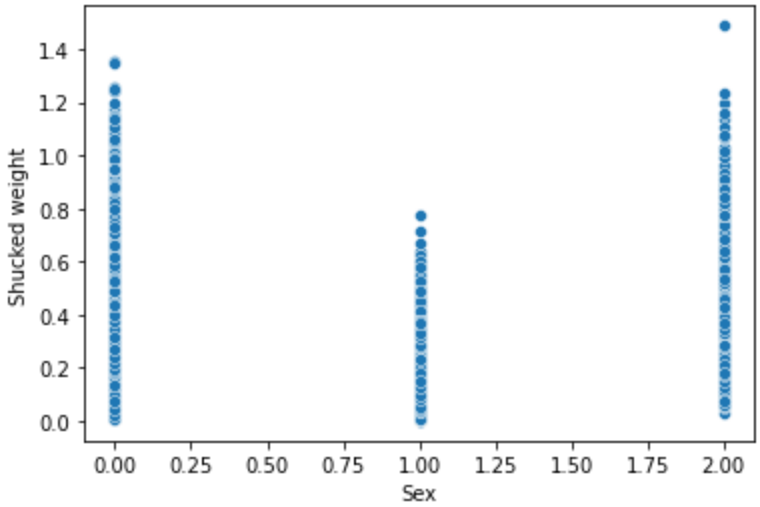
In [ ]:
```
sns.boxplot(data=df, x='Sex', y='Whole weight')
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f57beb39810>



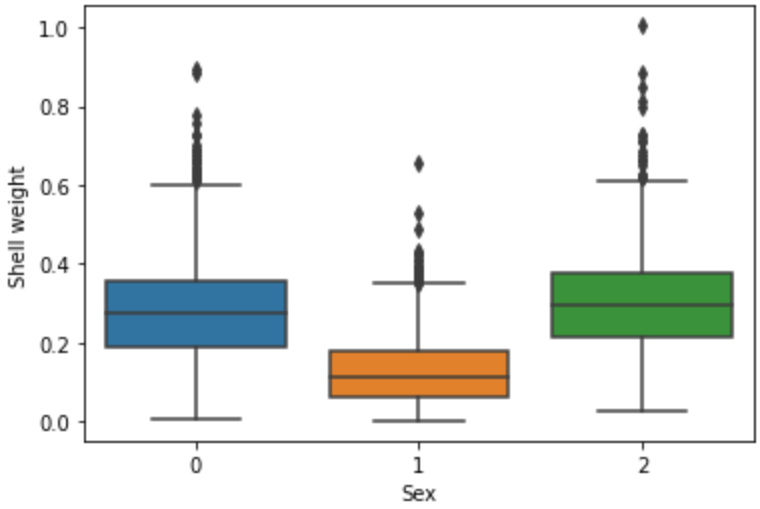
In [ ]:
```
sns.scatterplot(data=df, x='Sex', y='Shucked weight')
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f57a17fffd0>



In [ ]:
```
sns.boxplot(data=df, x='Sex', y='Shell weight')
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f57beb61b50>
```



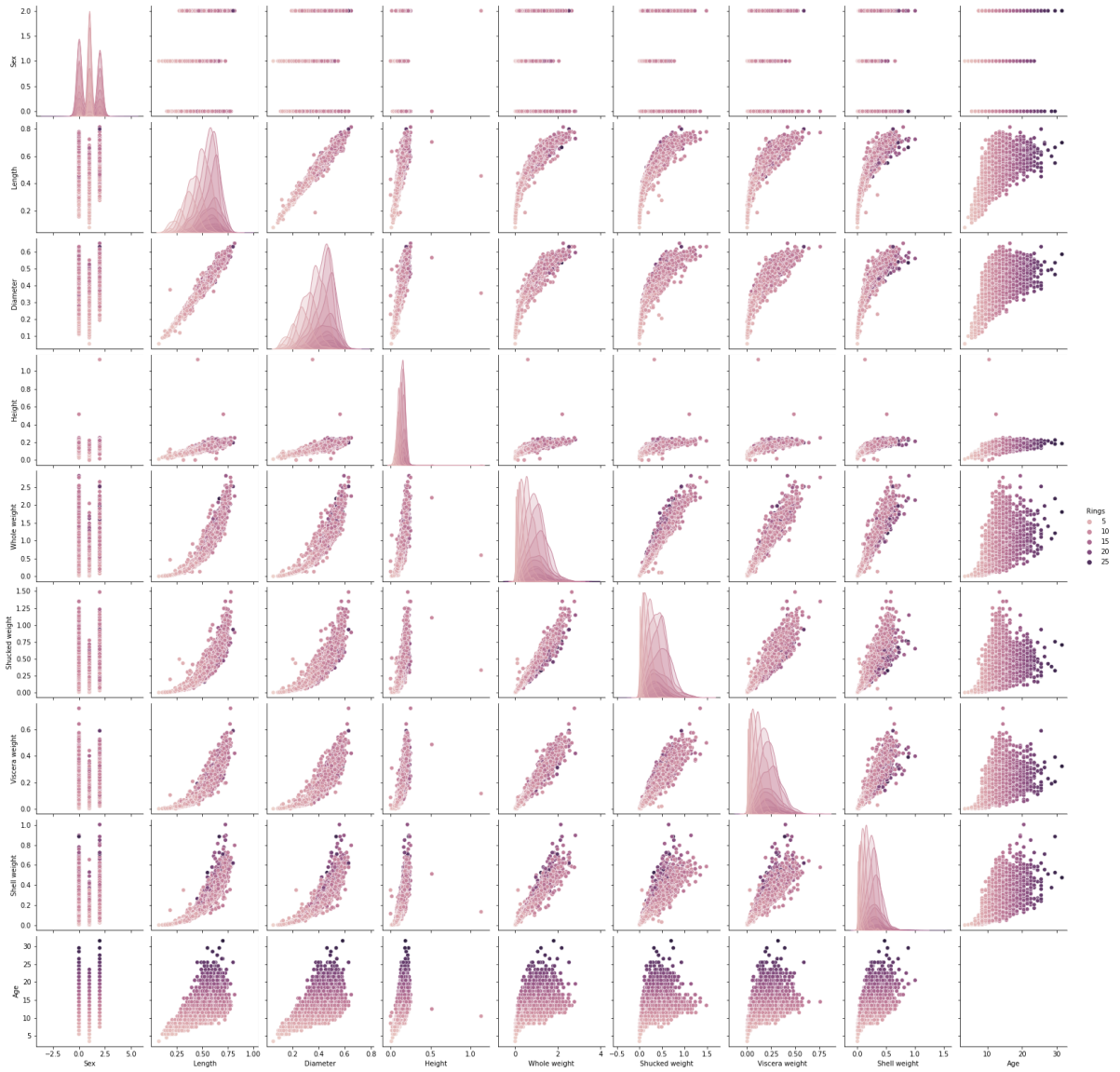
In [ ]:
```

```

__Univariate analysis__

In [ ]:
```
sns.pairplot(data=df, hue='Rings')
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7f57a111acd0>
```



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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Visc we
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000
mean	0.947091	0.523992	0.407881	0.139516	0.828742	0.359367	0.180
std	0.822240	0.120093	0.099240	0.041827	0.490389	0.221963	0.109
min	0.000000	0.075000	0.055000	0.000000	0.002000	0.001000	0.000
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	0.093
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	0.171
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	0.253
max	2.000000	0.815000	0.650000	1.130000	2.825500	1.488000	0.760

In [ ]:
```
df.corr()['Age']
```

Out[ ]:
```
Sex                0.034627
Length             0.556720
Diameter           0.574660
Height             0.557467
Whole weight       0.540390
Shucked weight     0.420884
Viscera weight     0.503819
Shell weight       0.627574
Rings              1.000000
Age                1.000000
Name: Age, dtype: float64
```

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

Out[ ]:
```
(4177, 10)
```

In [ ]:
```

```

__Checking outliers for the data__

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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	17.5
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	9.5
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	11.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	12.5
4	1	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	9.5

In [ ]:
```
df.drop('Age',axis=1,inplace=True)
```

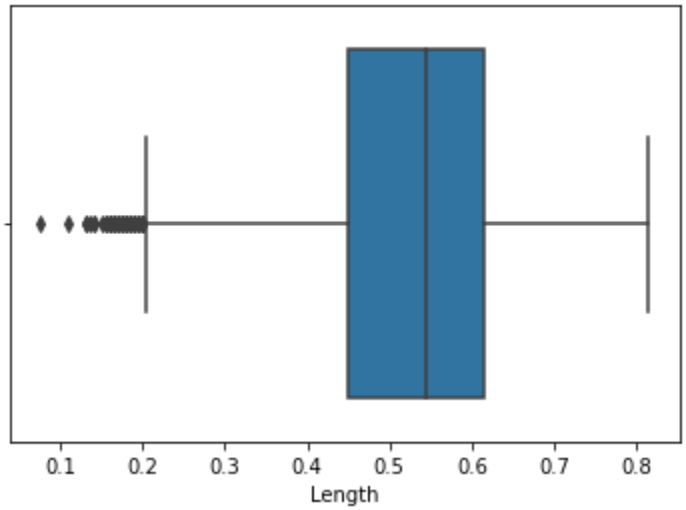
In [ ]:
```
df.head()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	0	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 [ ]:
```
sns.boxplot(x=df['Length'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579f241250>
```



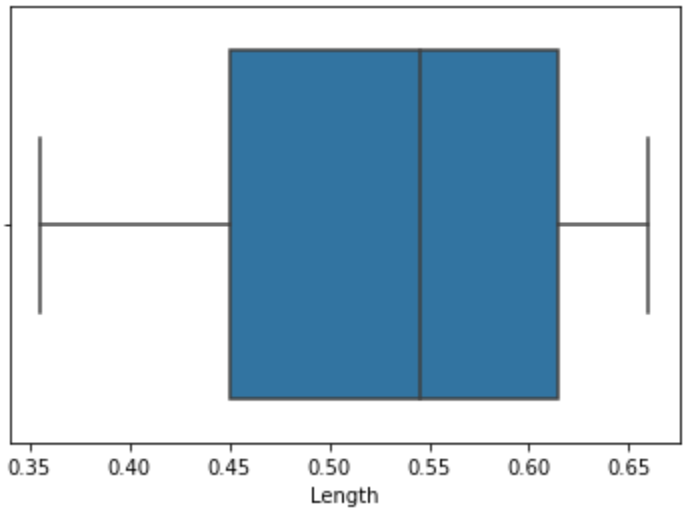
In [ ]:
```
tenth_per = np.percentile(df['Length'], 10)
nine_per  = np.percentile(df['Length'], 90)

df['Length'] = np.where(df['Length'] < tenth_per, tenth_per, df['Length'])
df['Length'] = np.where(df['Length'] > nine_per, nine_per, df['Length'])
```

IQR

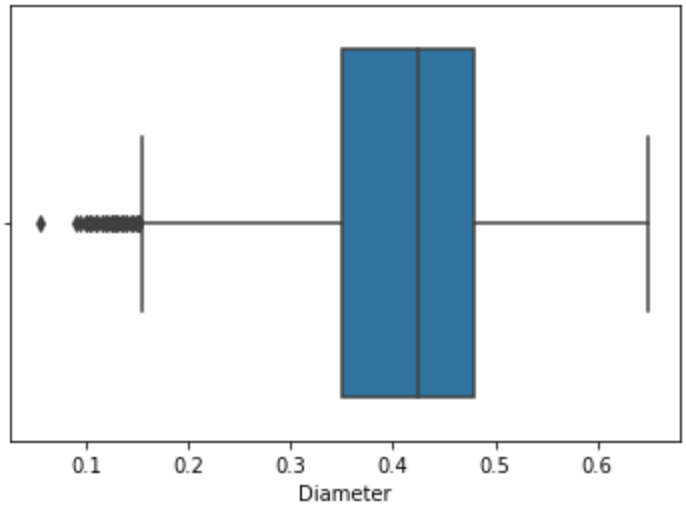
In [ ]:
```
sns.boxplot(x=df['Length'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579f24e390>
```



In [ ]:
```
sns.boxplot(x=df['Diameter'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579f1e90d0>
```

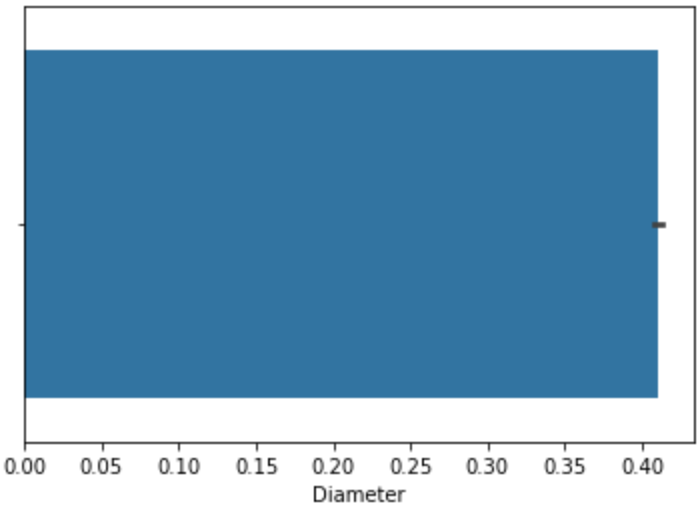


In [ ]:
```
tenth_per = np.percentile(df['Diameter'], 10)
nine_per  = np.percentile(df['Diameter'], 90)

df['Diameter'] = np.where(df['Diameter'] < tenth_per, tenth_per, df['Diamete
df['Diameter'] = np.where(df['Diameter'] > nine_per, nine_per, df['Diameter'
```

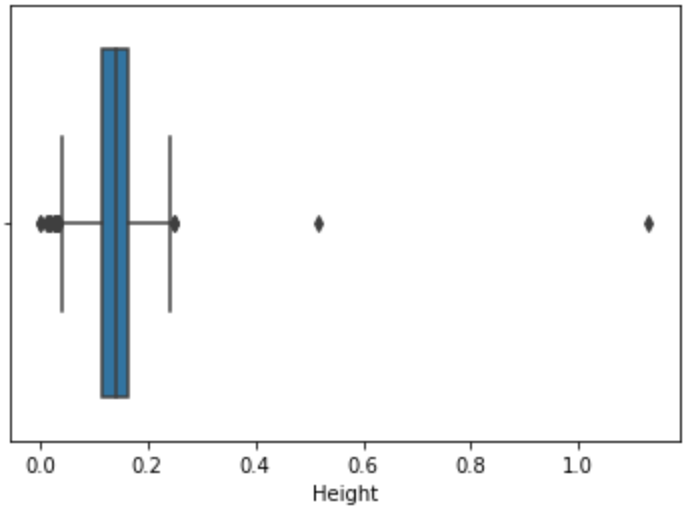
In [ ]:
```
sns.barplot(x=df['Diameter'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d9c2250>
```



In [ ]:
```
sns.boxplot(x=df['Height'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d9b9410>
```

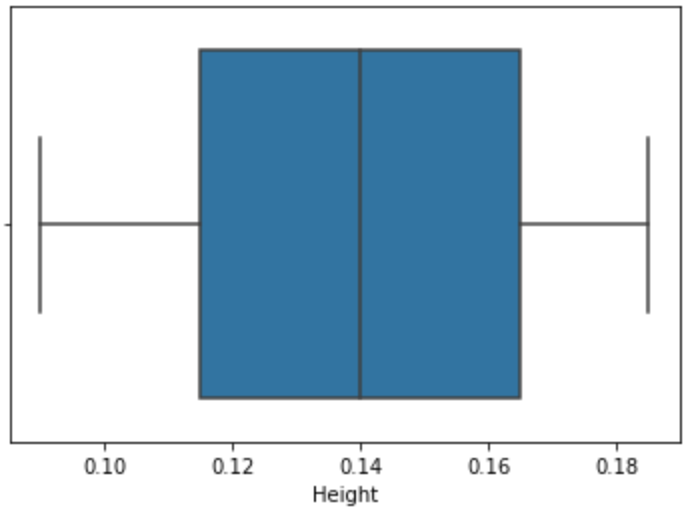


In [ ]:
```
tenth_per = np.percentile(df['Height'], 10)
nine_per  = np.percentile(df['Height'], 90)

df['Height'] = np.where(df['Height'] < tenth_per, tenth_per, df['Height'])
df['Height'] = np.where(df['Height'] > nine_per, nine_per, df['Height'])
```

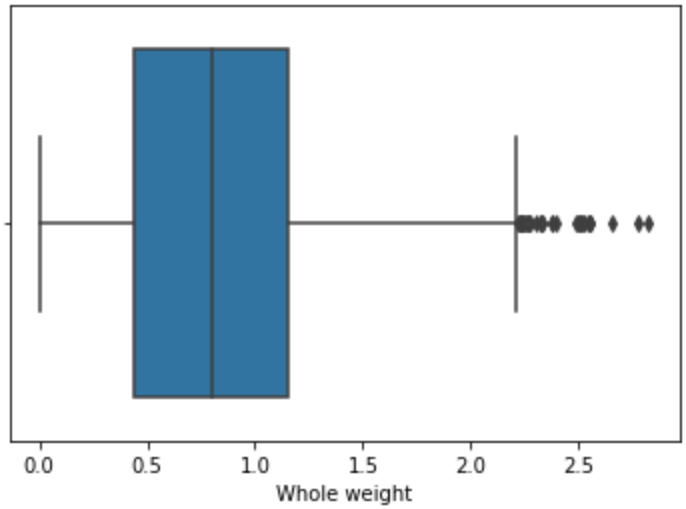
In [ ]:
```
sns.boxplot(x=df['Height'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d918ed0>
```



In [ ]:
```
sns.boxplot(x=df['Whole weight'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d869410>
```

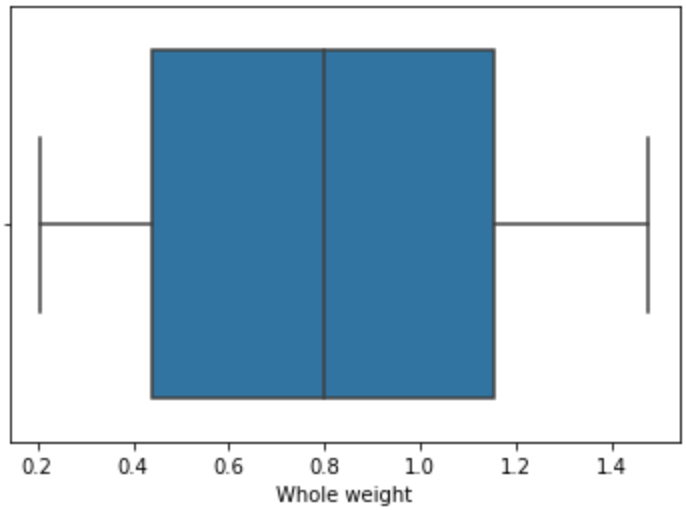


In [ ]:
```
tenth_per = np.percentile(df['Whole weight'], 10)
nine_per  = np.percentile(df['Whole weight'], 90)

df['Whole weight'] = np.where(df['Whole weight'] < tenth_per, tenth_per, df[
df['Whole weight'] = np.where(df['Whole weight'] > nine_per, nine_per, df['W
```

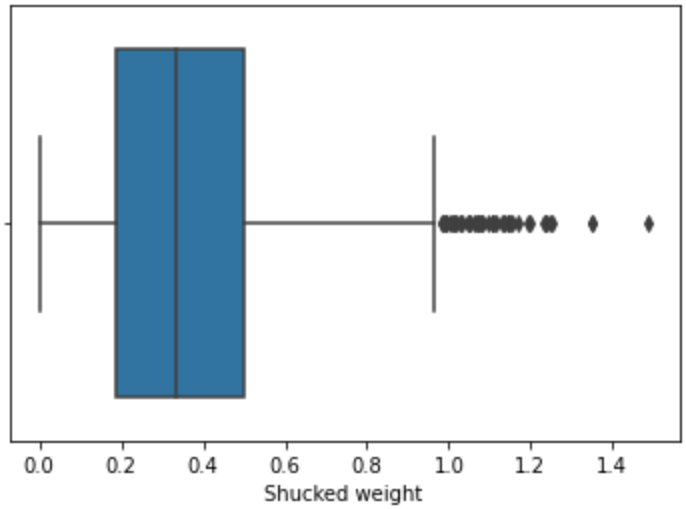
In [ ]:
```
sns.boxplot(x=df['Whole weight'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d992ad0>
```



In [ ]:
```
sns.boxplot(x=df['Shucked weight'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d7ccf10>
```

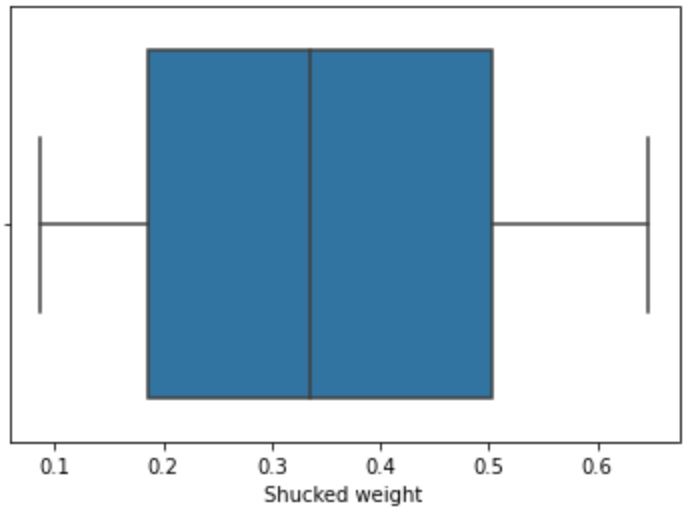


In [ ]:
```
tenth_per = np.percentile(df['Shucked weight'], 10)
nine_per  = np.percentile(df['Shucked weight'], 90)

df['Shucked weight'] = np.where(df['Shucked weight'] < tenth_per, tenth_per,
df['Shucked weight'] = np.where(df['Shucked weight'] > nine_per, nine_per, d
```

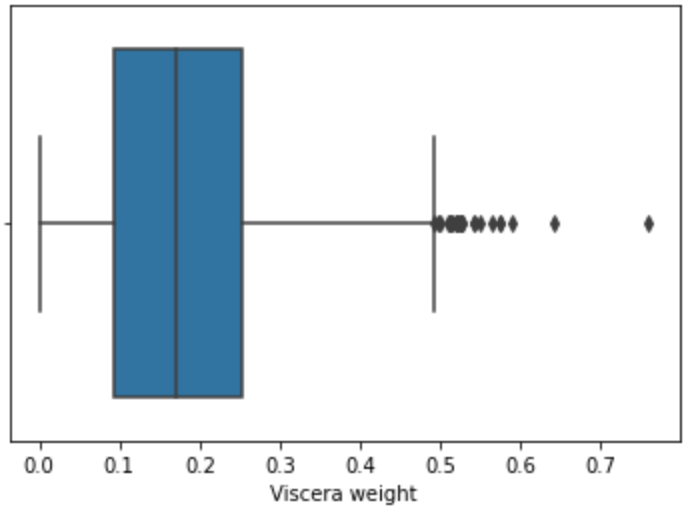
In [ ]:
```
sns.boxplot(x=df['Shucked weight'])
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d740e90>
```



In [ ]:
```
sns.boxplot(x=df['Viscera weight'])
```

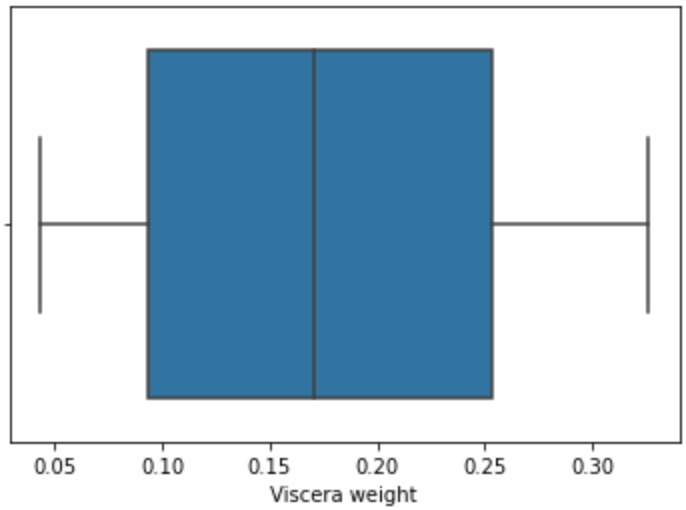
Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d7b99d0>
```



In [ ]:
```
tenth_per = np.percentile(df['Viscera weight'], 10)
nine_per  = np.percentile(df['Viscera weight'], 90)

df['Viscera weight'] = np.where(df['Viscera weight'] < tenth_per, tenth_per,
df['Viscera weight'] = np.where(df['Viscera weight'] > nine_per, nine_per, d
sns.boxplot(x=df['Viscera weight'])
```

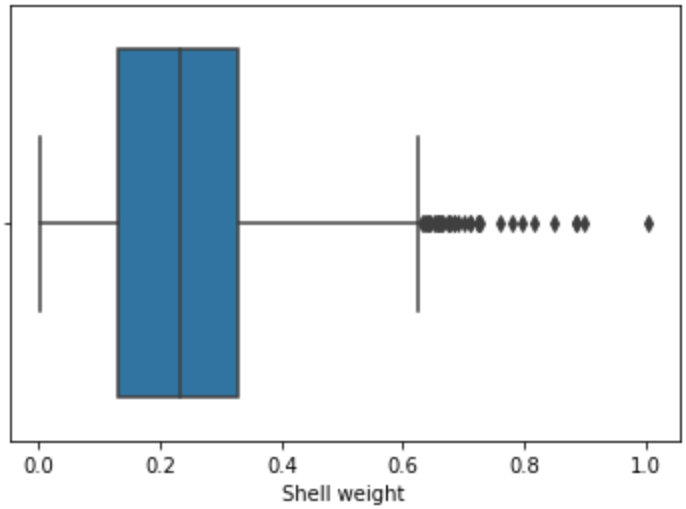
Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d685210>
```



In [ ]:
```
sns.boxplot(df['Shell weight'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretatio
n.
  FutureWarning
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d5ff150>
```

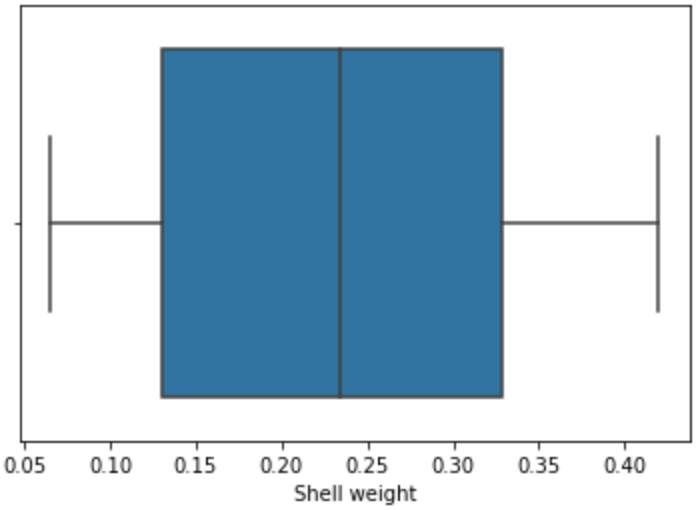


In [ ]:
```
tenth_per = np.percentile(df['Shell weight'], 10)
nine_per  = np.percentile(df['Shell weight'], 90)

df['Shell weight'] = np.where(df['Shell weight'] < tenth_per, tenth_per, df[
df['Shell weight'] = np.where(df['Shell weight'] > nine_per, nine_per, df['S
sns.boxplot(df['Shell weight'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretatio
n.
  FutureWarning
```

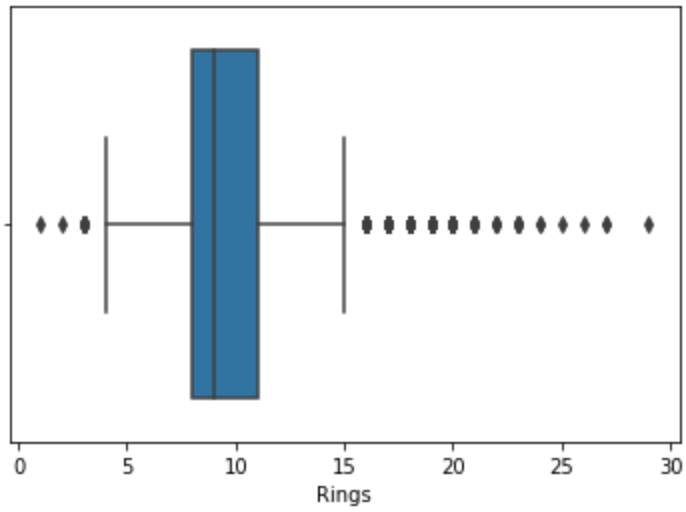
Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f579d569a10>
```



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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretatio
n.
  FutureWarning
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f579d556f50>

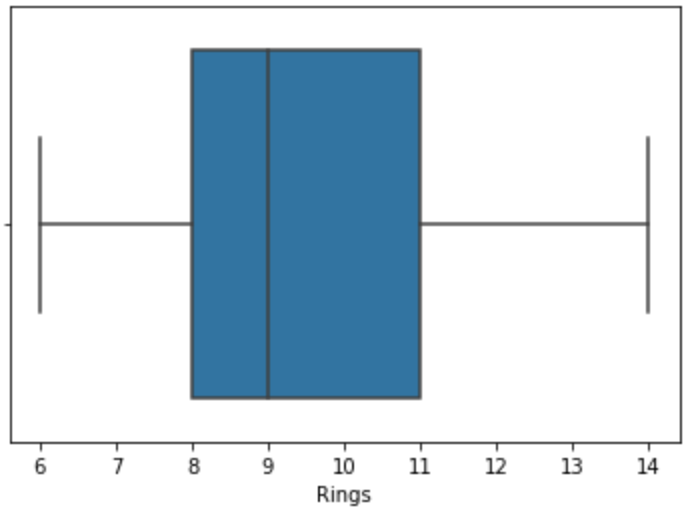


In [ ]:
```
tenth_per = np.percentile(df['Rings'], 10)
nine_per  = np.percentile(df['Rings'], 90)

df['Rings'] = np.where(df['Rings'] < tenth_per, tenth_per, df['Rings'])
df['Rings'] = np.where(df['Rings'] > nine_per, nine_per, df['Rings'])
sns.boxplot(df['Rings'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretatio
n.
  FutureWarning
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f579d6b5d10>



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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Visc we
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000
mean	0.947091	0.527491	0.410466	0.139701	0.807958	0.348481	0.175
std	0.822240	0.099873	0.083713	0.031559	0.418877	0.185356	0.093
min	0.000000	0.355000	0.265000	0.090000	0.205000	0.086500	0.043
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	0.093
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	0.171
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	0.253
max	2.000000	0.660000	0.522000	0.185000	1.478200	0.647000	0.326

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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	14.0
1	0	0.355	0.265	0.090	0.2255	0.0995	0.0485	0.070	7.0
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9.0
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10.0
4	1	0.355	0.265	0.090	0.2050	0.0895	0.0433	0.065	7.0

Outlier treatment

In [ ]:
```
df['Age'] = df['Rings'] + 2.5
```

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

Out[ ]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	14.0	16.5
1	0	0.355	0.265	0.090	0.2255	0.0995	0.0485	0.070	7.0	9.5
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9.0	11.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10.0	12.5
4	1	0.355	0.265	0.090	0.2050	0.0895	0.0433	0.065	7.0	9.5

In [ ]:
```
X = df.drop('Age', axis=1)
y = df['Age']
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test  = train_test_split(X, y, test_size=0.3, randc
```

Out[ ]: (2923, 9)

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

Out[ ]: (1254, 9)

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

Out[ ]: (2923,)

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

Out[ ]: (1254,)

In [ ]:
```
from sklearn.linear_model import LinearRegression
model1 = LinearRegression()
model1.fit(X_train,y_train)
```

Out[ ]: LinearRegression()

In [ ]:
```
y_pred1 = model1.predict(X_test)
```

In [ ]:
```
y_pred1
```

Out[ ]: array([12.5, 9.5, 12.5, ..., 8.5, 16.5, 12.5])

In [ ]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_scor
```

In [ ]:
```
print(mean_absolute_error( y_test, y_pred1))
print(mean_squared_error(y_test, y_pred1))
```

```
7.592721418808088e-16
1.4544229767711159e-30
```

In [ ]:
```
print(r2_score( y_test,y_pred1))
```

```
1.0
```

In [ ]:
```
from sklearn.ensemble import RandomForestRegressor
model2 = RandomForestRegressor(n_estimators=500)
model2.fit(X_train, y_train)
```

Out[ ]: RandomForestRegressor(n_estimators=500)

In [ ]:
```
y_pred2 = model2.predict(X_test)
```

In [ ]:
```
y_pred2
```

Out[ ]: array([12.5, 9.5, 12.5, ..., 8.5, 16.5, 12.5])

In [ ]:
```
print(r2_score( y_test,y_pred2))
```

In [ ]: