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

```
df = pd.read_csv("/content/abalone.csv")
```

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shel
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	

```
df.tail()
```

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

```
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
```

```
df.describe()
```

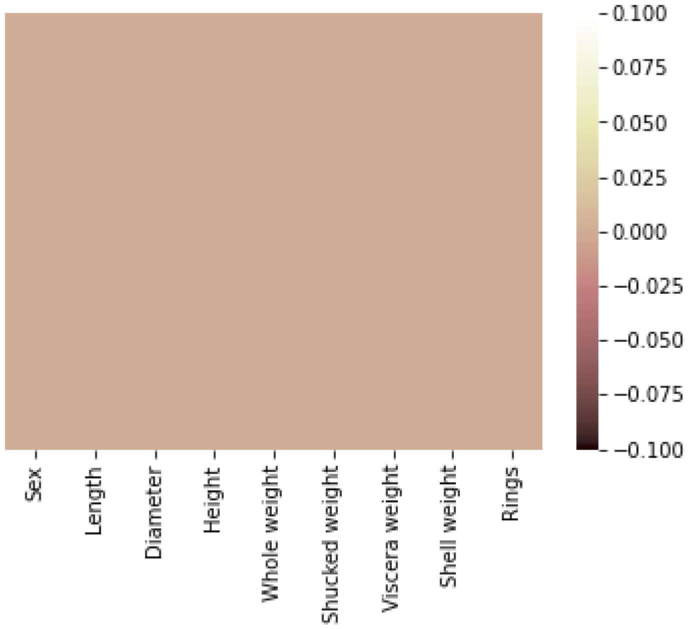
	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	

```
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
```

```
sns.heatmap(df.isnull(),yticklabels=False,cmap='pink')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2253b4f8d0>
```



```
df.corr()
```

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Ring
Length	1.000000	0.986812	0.827554	0.925261	0.897914	0.903018	0.897706	0.55672
Diameter	0.986812	1.000000	0.833684	0.925452	0.893162	0.899724	0.905330	0.57466
Height	0.827554	0.833684	1.000000	0.819221	0.774972	0.798319	0.817338	0.55746
Whole weight	0.925261	0.925452	0.819221	1.000000	0.969405	0.966375	0.955355	0.54039
Shucked weight	0.897914	0.893162	0.774972	0.969405	1.000000	0.931961	0.882617	0.42088
Viscera	0.903018	0.899724	0.798319	0.966375	0.931961	1.000000	0.907656	0.50381

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

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

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

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

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

```
df.head()
```

	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

```
df.tail()
```

	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
4175	2	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10

ADDING AGE COLUMN

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

```
df.head()
```

	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

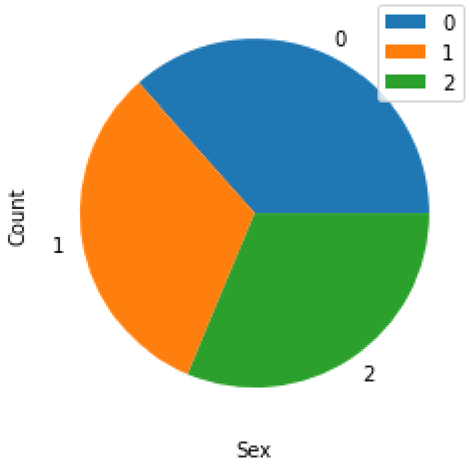
```
df.columns
```

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

Data visualization

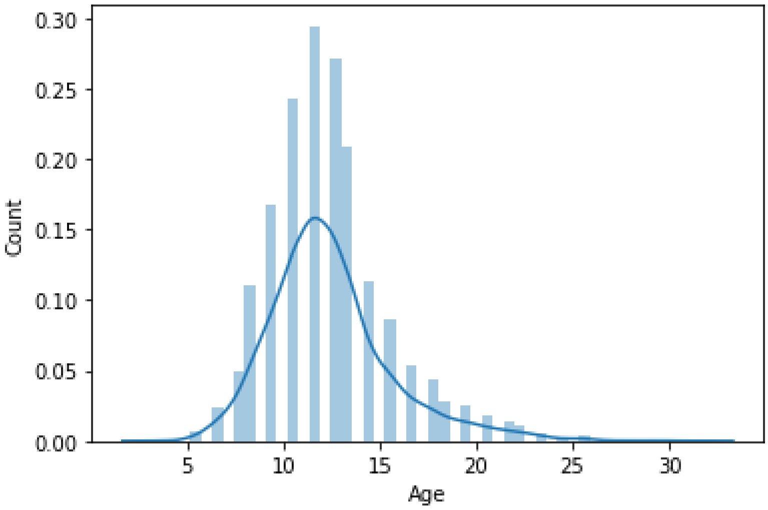
```
df['Sex'].value_counts().plot(kind='pie')
plt.legend()
plt.xlabel('Sex')
plt.ylabel('Count')
```

```
    Text(0, 0.5, 'Count')
```



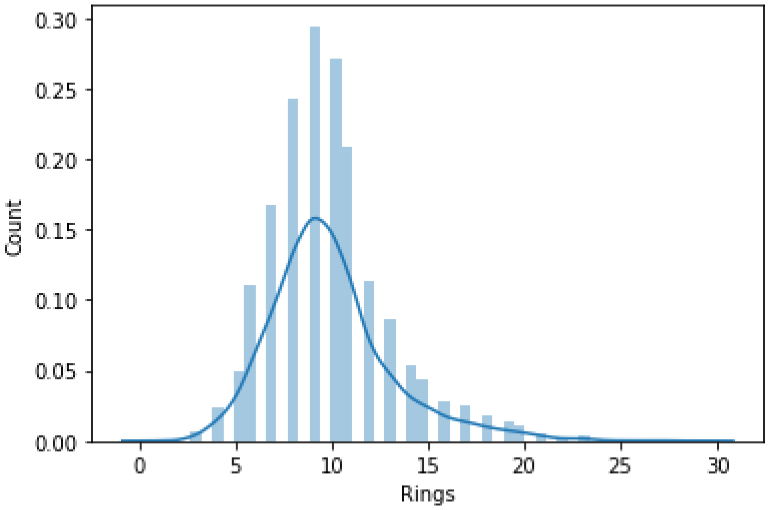
```
sns.distplot(df['Age'])
plt.xlabel('Age')
plt.ylabel('Count')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
Text(0, 0.5, 'Count')
```



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

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
Text(0, 0.5, 'Count')
```



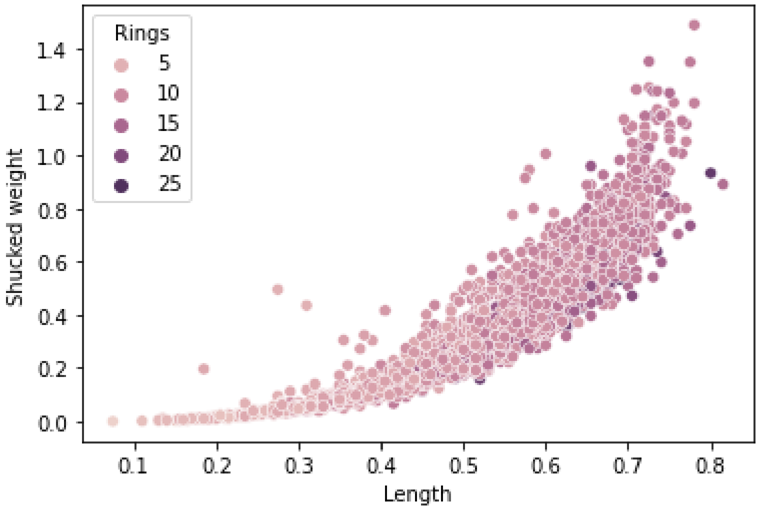
Bi-variate analysis

```
df.head()
```

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

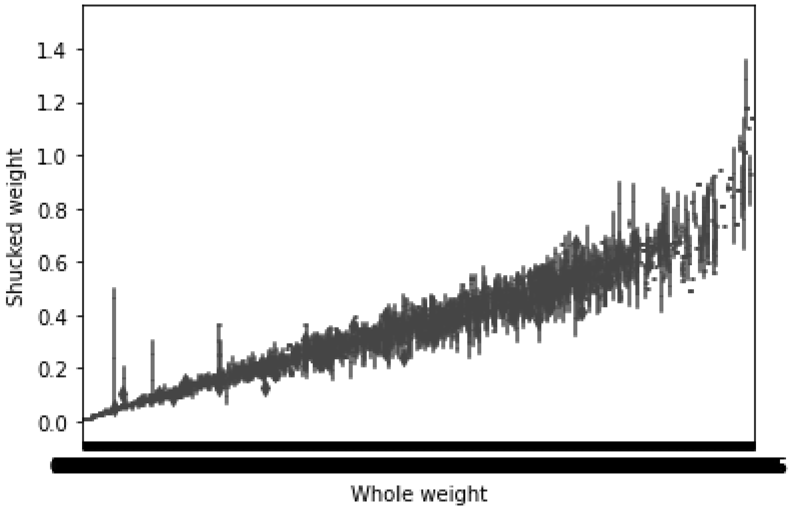
```
sns.scatterplot(data=df, x='Length', y='Shucked weight', hue='Rings',)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2250a2b710>
```



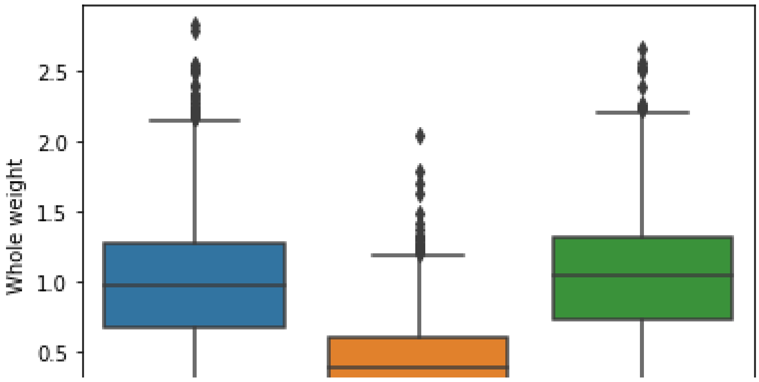
```
sns.boxplot(data=df, x='Whole weight', y='Shucked weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f225031a550>
```



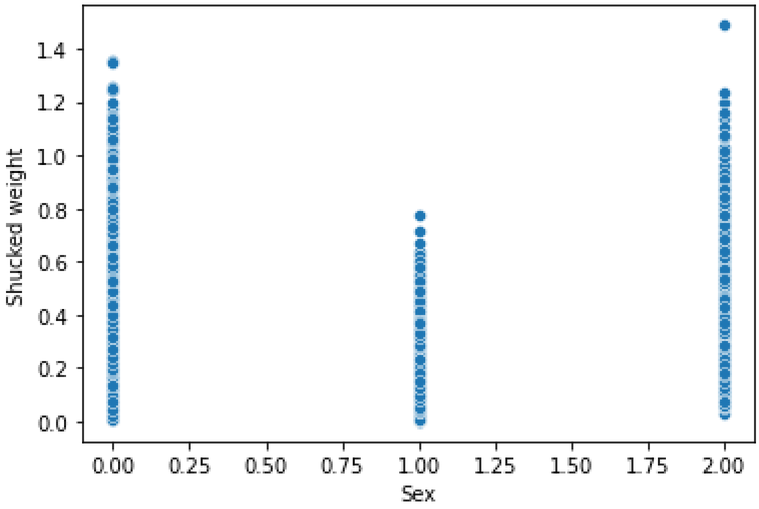
```
sns.boxplot(data=df, x='Sex', y='Whole weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2241d35490>
```



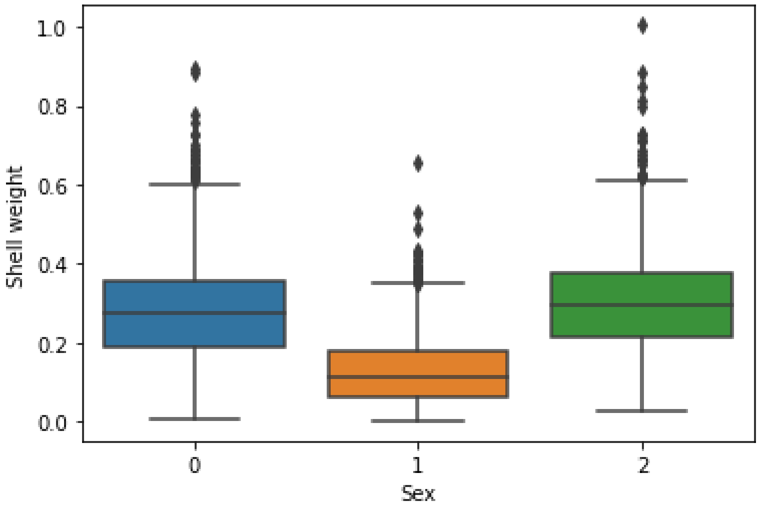
```
sns.scatterplot(data=df, x='Sex', y='Shucked weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2241c9a550>
```



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

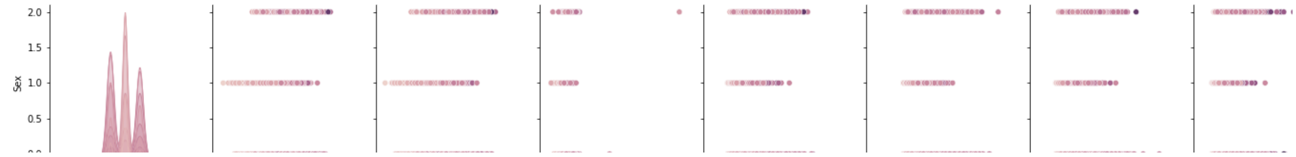
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2241c69750>
```



Univariate analysis

```
sns.pairplot(data=df, hue='Rings')
```

```
<seaborn.axisgrid.PairGrid at 0x7f2241462790>
```



```
df.describe()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4
mean	0.947091	0.523992	0.407881	0.139516	0.828742	0.359367	
std	0.822240	0.120093	0.099240	0.041827	0.490389	0.221963	
min	0.000000	0.075000	0.055000	0.000000	0.002000	0.001000	
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	
max	2.000000	0.815000	0.650000	1.130000	2.825500	1.488000	

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

```
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
```

```
df.shape
```

```
(4177, 10)
```

Checking outliers for the data

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	w
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	

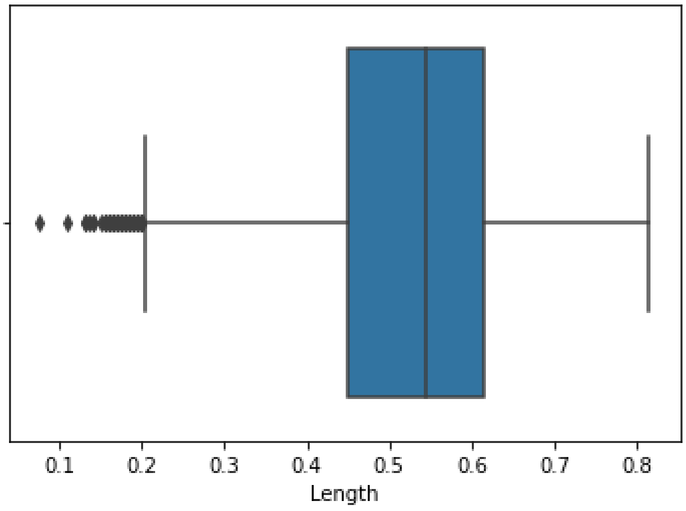
```
df.drop('Age',axis=1,inplace=True)
```

```
df.head()
```

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

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223f524090>
```

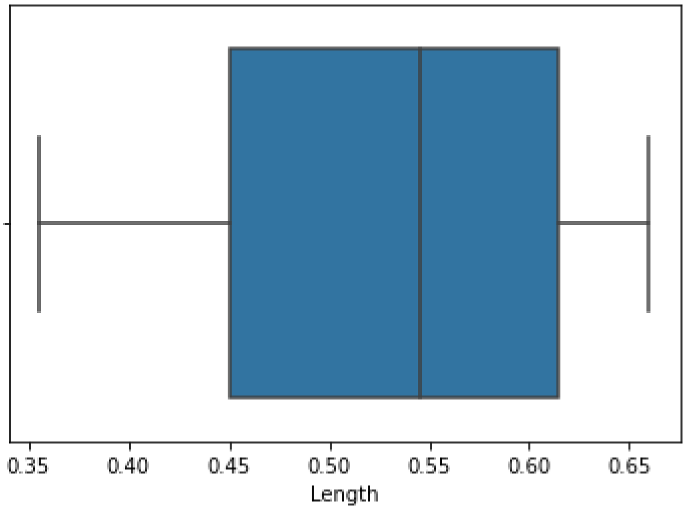


```
tenth_per = np.percentile(df['Length'], 10)
nine_per  = np.percentile(df['Length'], 90)

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

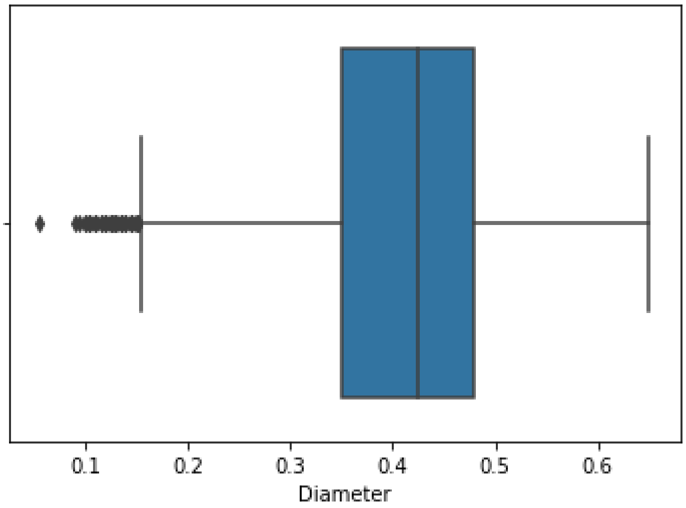
```
sns.boxplot(x=df['Length'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223f5fab50>
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223dcd1750>
```

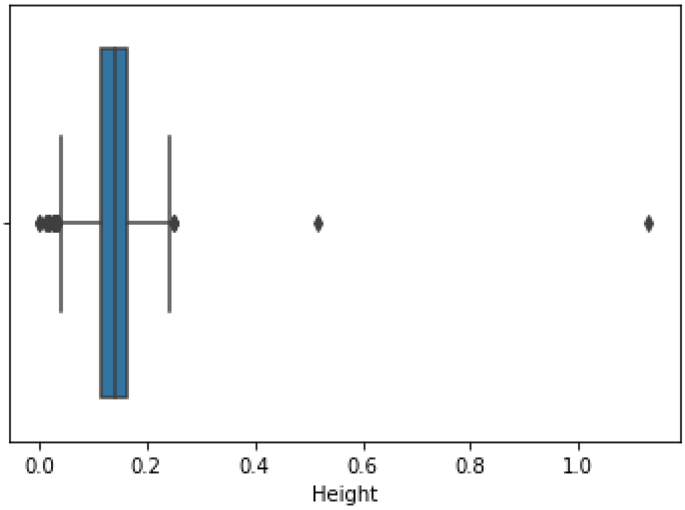


```
tenth_per = np.percentile(df['Diameter'], 10)
nine_per  = np.percentile(df['Diameter'], 90)

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


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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223dcad3d0>
```

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

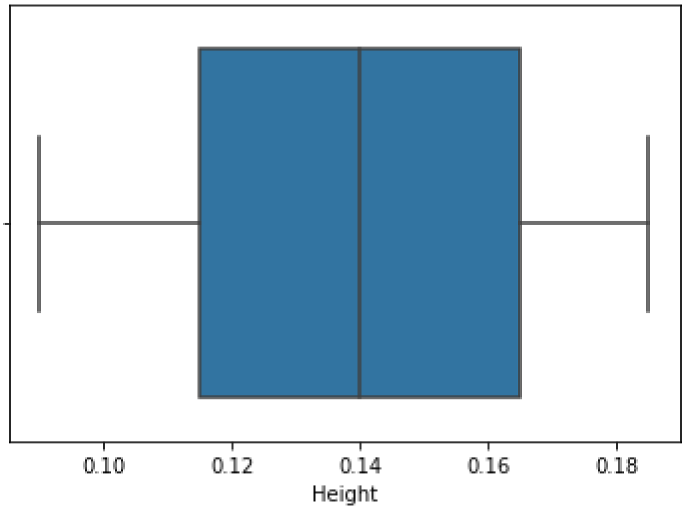
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223dc82150>
```



```
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'])
```

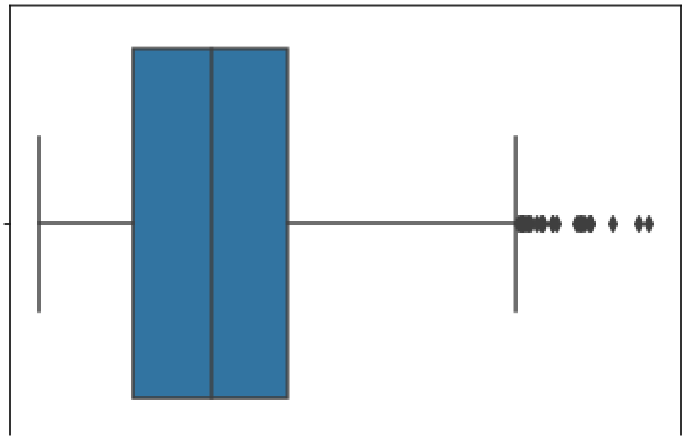
```
sns.boxplot(x=df['Height'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223dbe7e50>
```



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

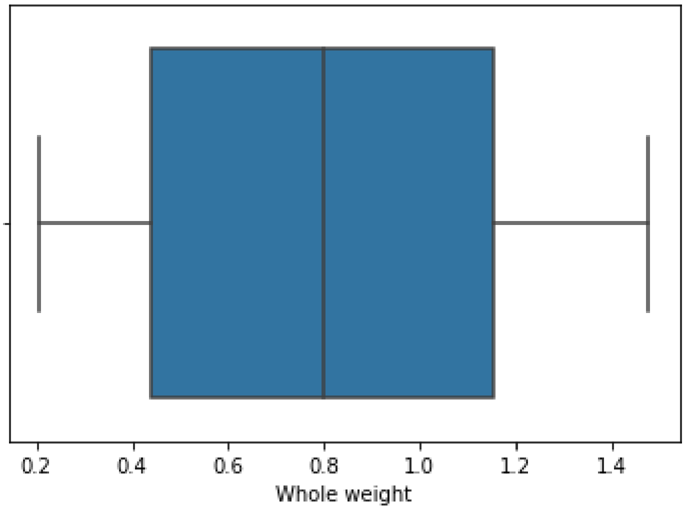
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223db5b1d0>
```



```
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['Whole weight'
df['Whole weight'] = np.where(df['Whole weight'] > nine_per, nine_per, df['Whole weight'])
```

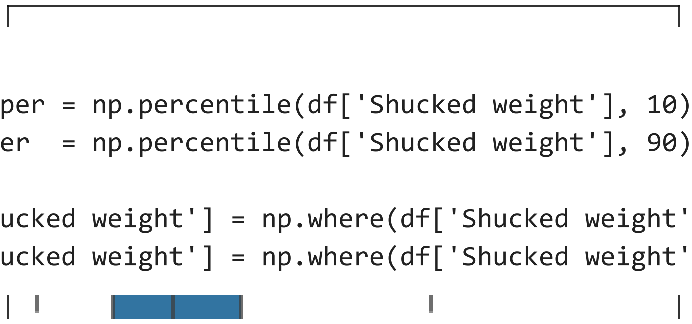
```
sns.boxplot(x=df['Whole weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223db40510>
```



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

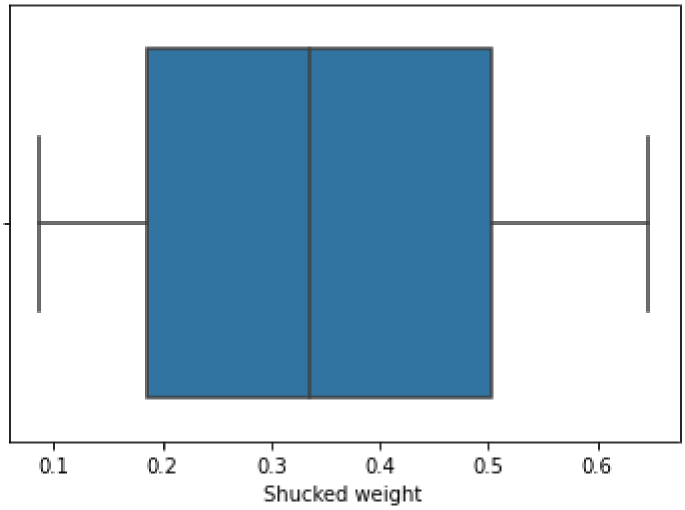
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223daba850>
```



```
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 w
df['Shucked weight'] = np.where(df['Shucked weight'] > nine_per, nine_per, df['Shucked wei
```

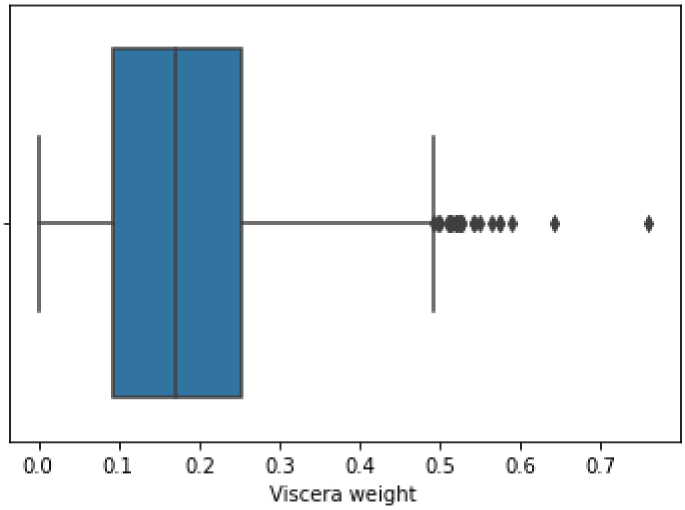
```
sns.boxplot(x=df['Shucked weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223da27d90>
```



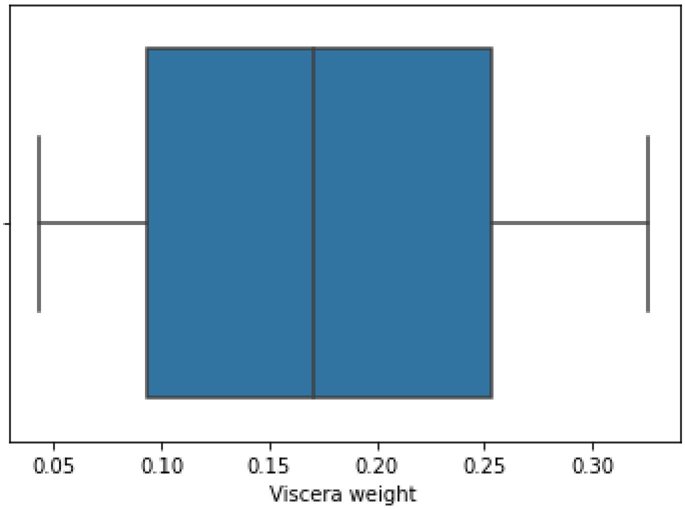
```
sns.boxplot(x=df['Viscera weight'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223d999c90>
```



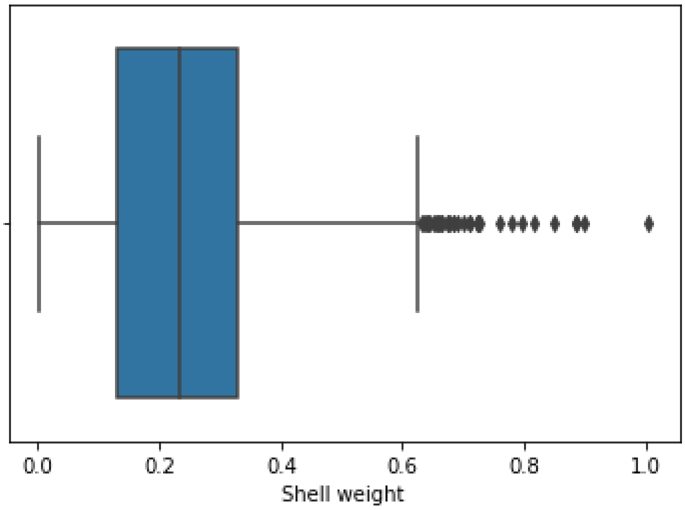
```
tenth_per = np.percentile(df['Viscera weight'], 10)
nine_per  = np.percentile(df['Viscera weight'], 90)

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f223d972590>
```



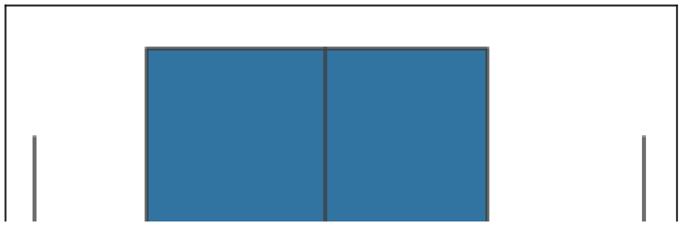
```
sns.boxplot(df['Shell weight'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f223d8fa850>
```



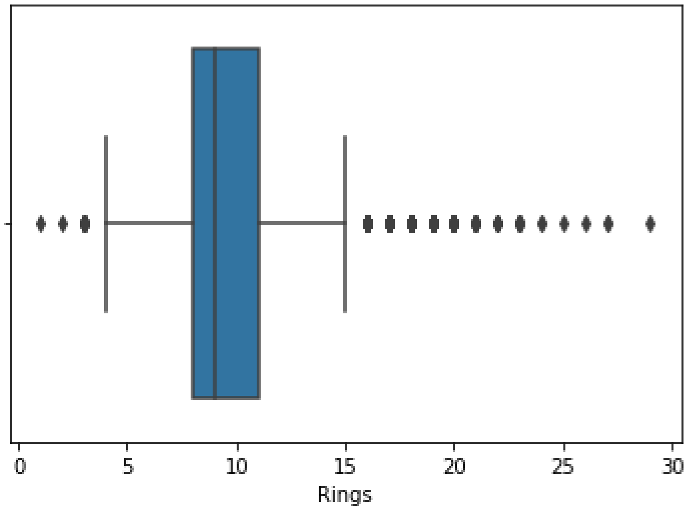
```
tenth_per = np.percentile(df['Shell weight'], 10)
nine_per  = np.percentile(df['Shell weight'], 90)

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f223d870390>
```



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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f223d83db10>
```



```
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: FutureWarning: Pas
  FutureWarning
```

```
df.describe()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4
mean	0.947091	0.527491	0.410466	0.139701	0.807958	0.348481	
std	0.822240	0.099873	0.083713	0.031559	0.418877	0.185356	
min	0.000000	0.355000	0.265000	0.090000	0.205000	0.086500	
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	
max	2.000000	0.660000	0.522000	0.185000	1.478200	0.647000	

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shel
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	
1	0	0.355	0.265	0.090	0.2255	0.0995	0.0485	
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	
4	1	0.355	0.265	0.090	0.2050	0.0895	0.0433	

Outlier treatment

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

```
df.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	w
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	
1	0	0.355	0.265	0.090	0.2255	0.0995	0.0485	
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	
4	1	0.355	0.265	0.090	0.2050	0.0895	0.0433	

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

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, random_state=101)
```

```
X_train.shape
```

```
     (2923, 9)
```

```
X_test.shape
```

```
     (1254, 9)
```

```
y_train.shape
```

```
     (2923,)
```

```
y_test.shape
```

```
     (1254,)
```

```
from sklearn.linear_model import LinearRegression
model1 = LinearRegression()
model1.fit(X_train,y_train)
```

```
     LinearRegression()
```

```
y_pred1 = model1.predict(X_test)
```

```
y_pred1
```

```
     array([12.5,  9.5, 12.5, ...,  8.5, 16.5, 12.5])
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
print(mean_absolute_error( y_test, y_pred1))
print(mean_squared_error(y_test, y_pred1))
```

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

```
print(r2_score( y_test,y_pred1))
```

```
     1.0
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
model2 = RandomForestRegressor(n_estimators=500)
model2.fit(X_train, y_train)

     RandomForestRegressor(n_estimators=500)
```

```
y_pred2 = model2.predict(X_test)
```

```
y_pred2

     array([12.5,  9.5, 12.5, ...,  8.5, 16.5, 12.5])
```

```
print(r2_score( y_test,y_pred2))

     1.0
```