```
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	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

```
df.tail()
```

	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

```
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	4177.
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.

```
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 0x7f1043c14cd0>
```

```
df.corr()
```

	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

```
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

```
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
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

```
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

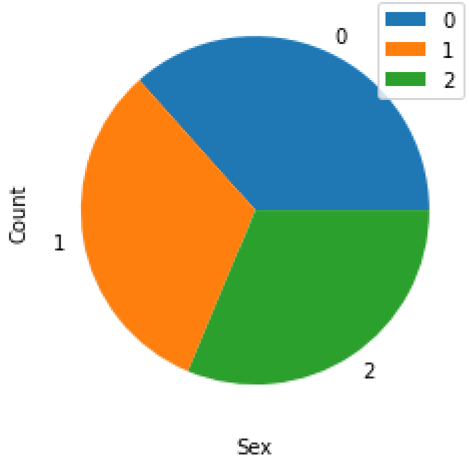
```
df.columns
```

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

Data visualization

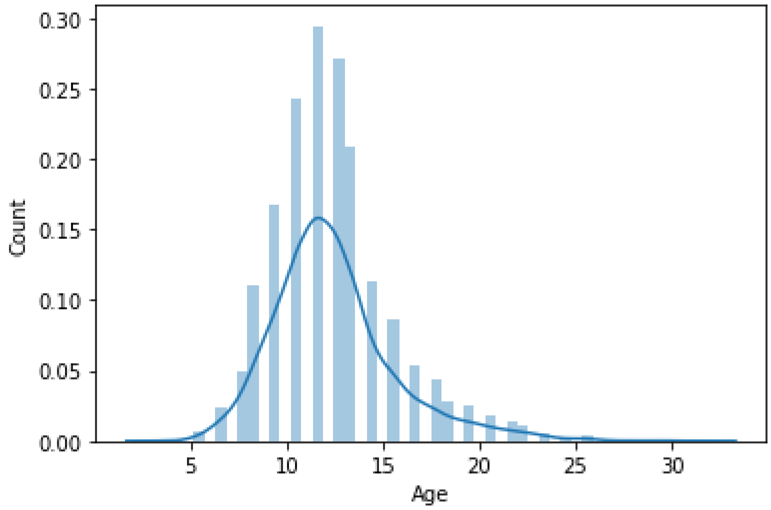
```
df['Sex'].value_counts().plot(kind='pie')
plt.legend()
plt.xlabel('Sex')
plt.ylabel('Count')
```

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



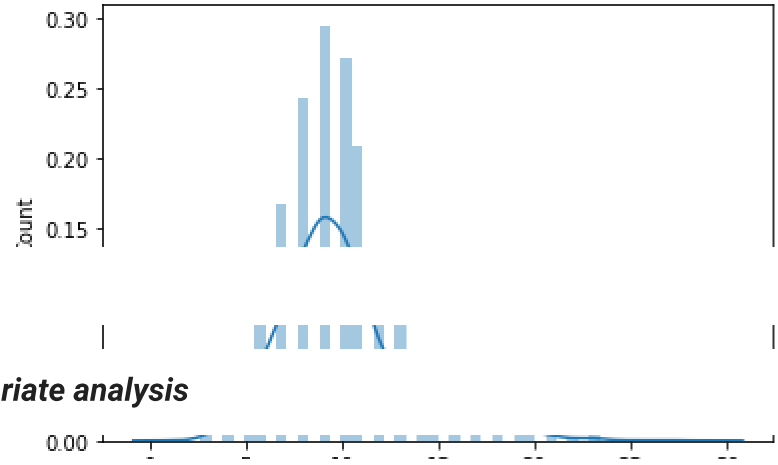
```
sns.distplot(df['Age'])
plt.xlabel('Age')
plt.ylabel('Count')
```

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



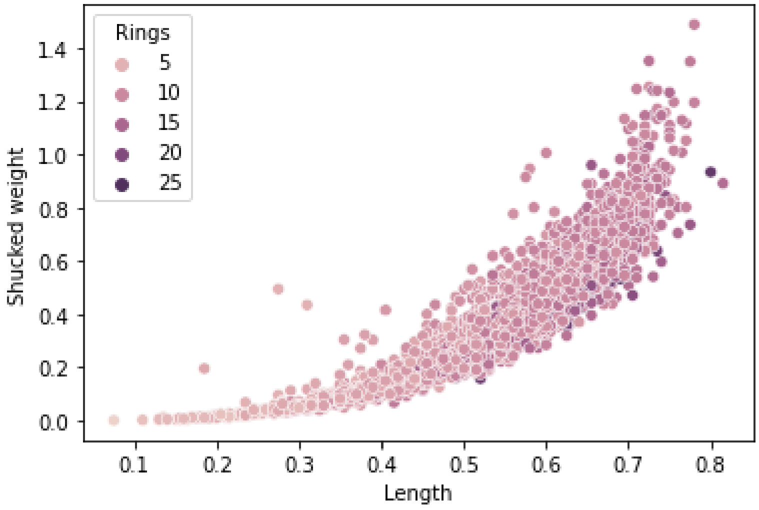
Bi-variate analysis

```
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

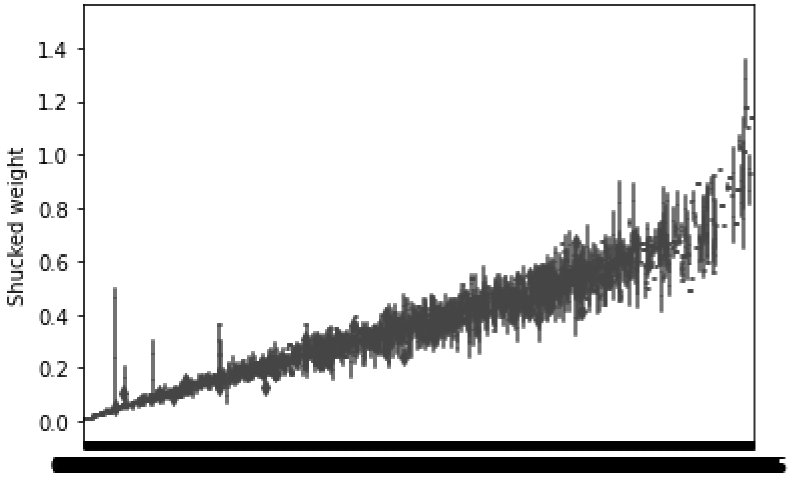
```
sns.scatterplot(data=df, x='Length', y='Shucked weight', hue='Rings',)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f10435c8190>
```



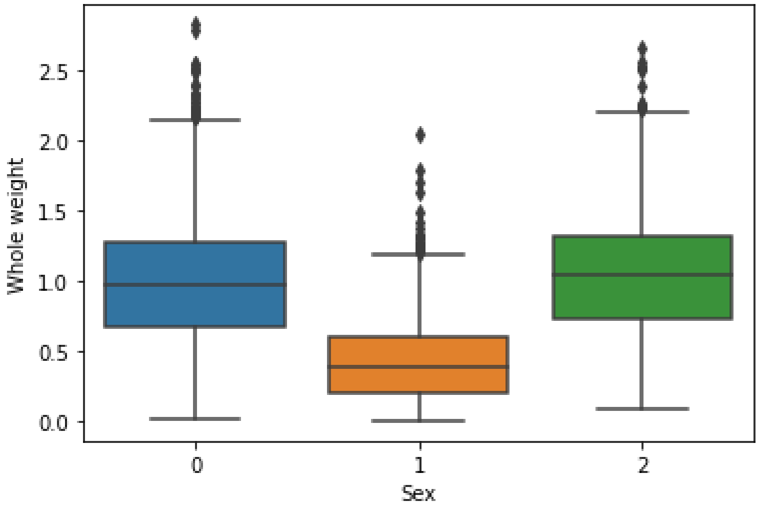
```
sns.boxplot(data=df, x='Whole weight', y='Shucked weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1042effcd0>
```



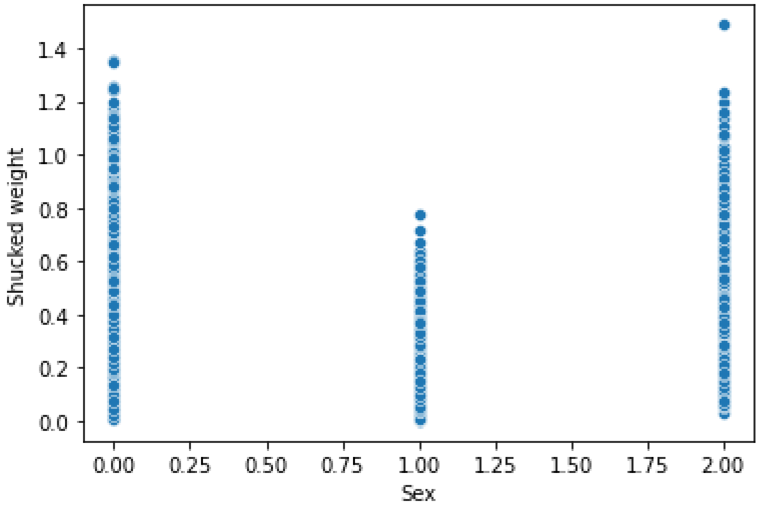
```
sns.boxplot(data=df, x='Sex', y='Whole weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f10363711d0>
```



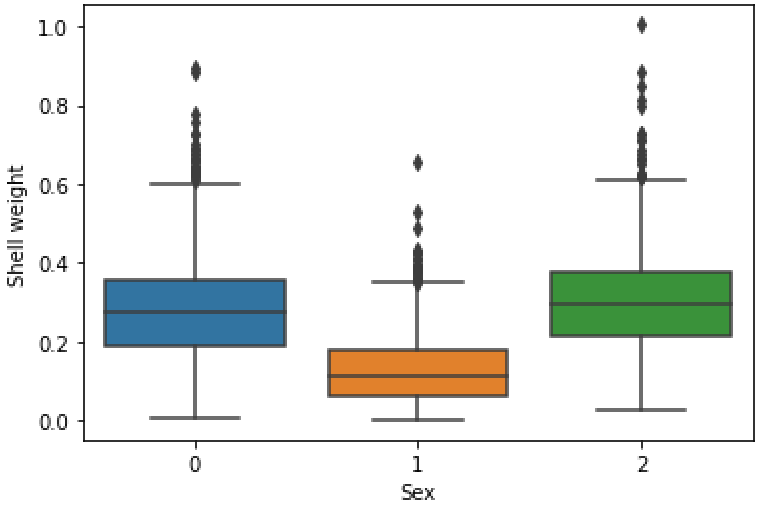
```
sns.scatterplot(data=df, x='Sex', y='Shucked weight')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1034358b90>
```



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

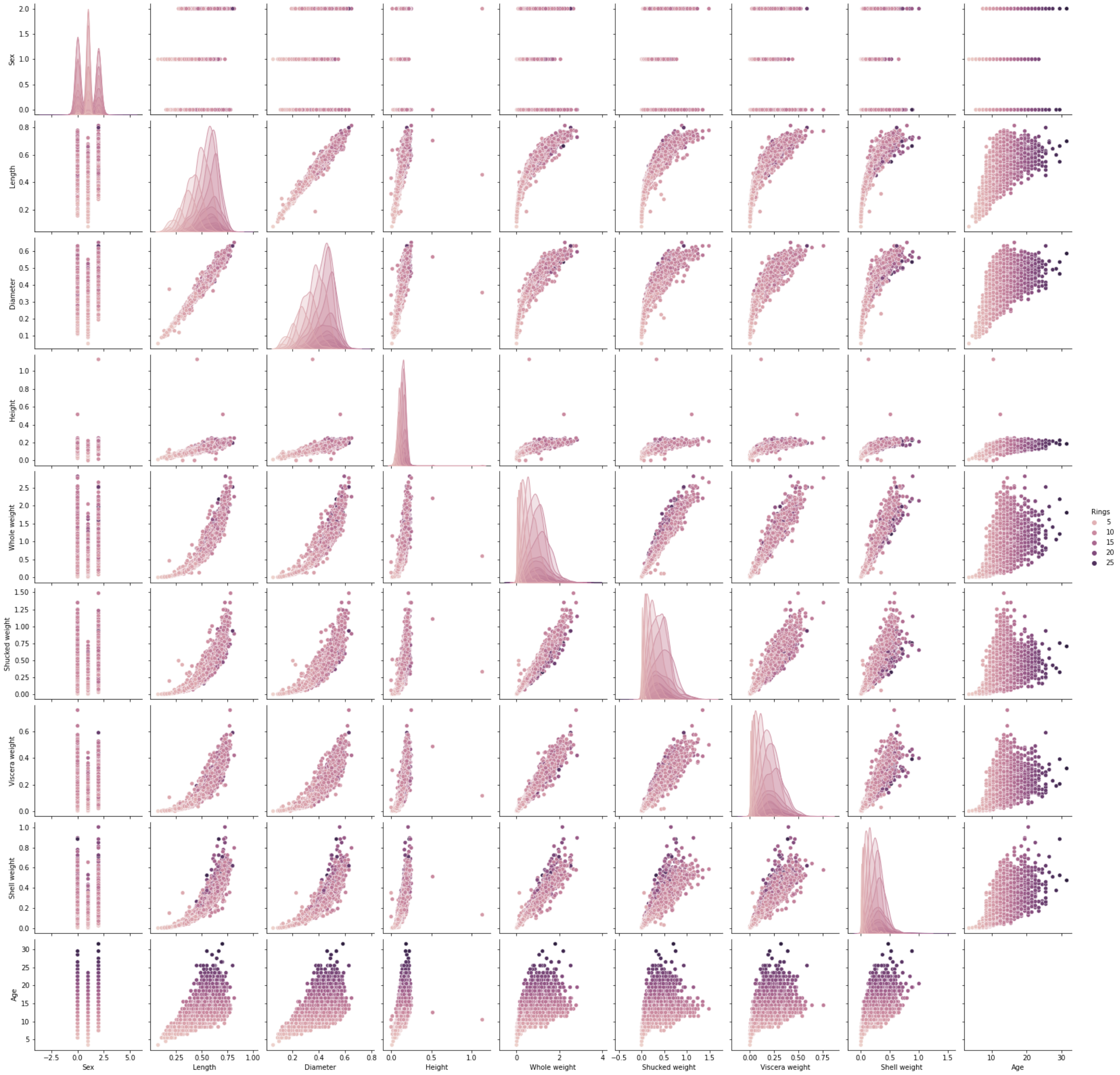
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1034b7eed0>
```

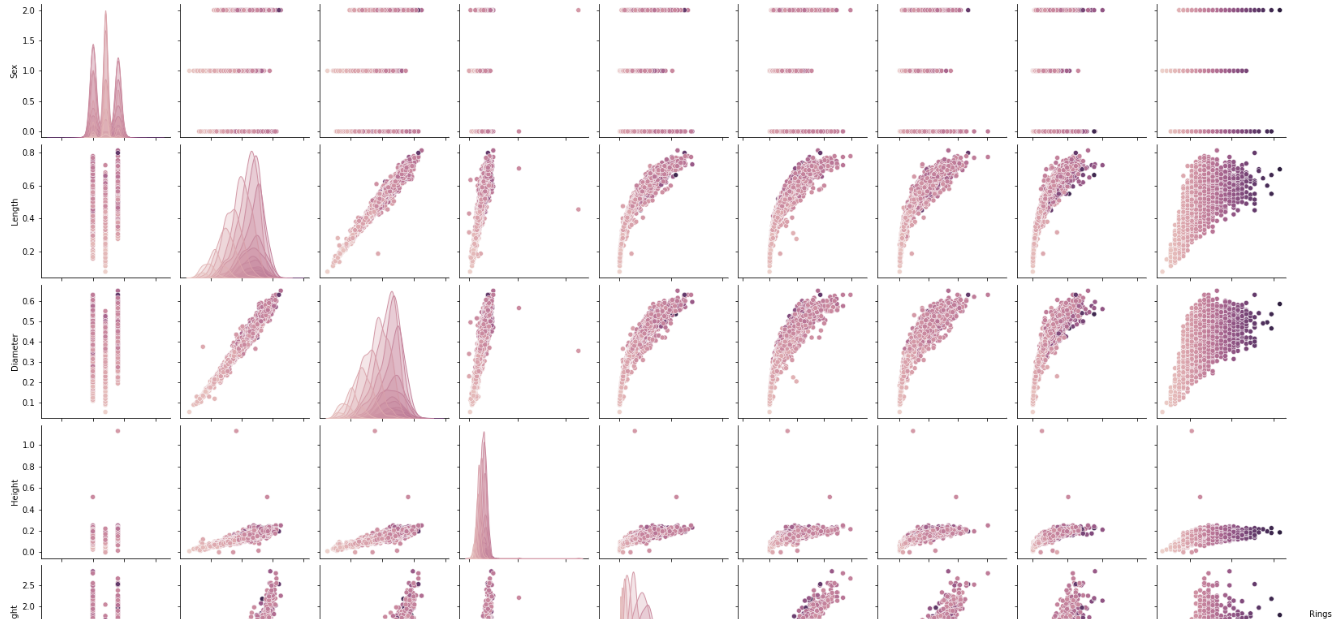


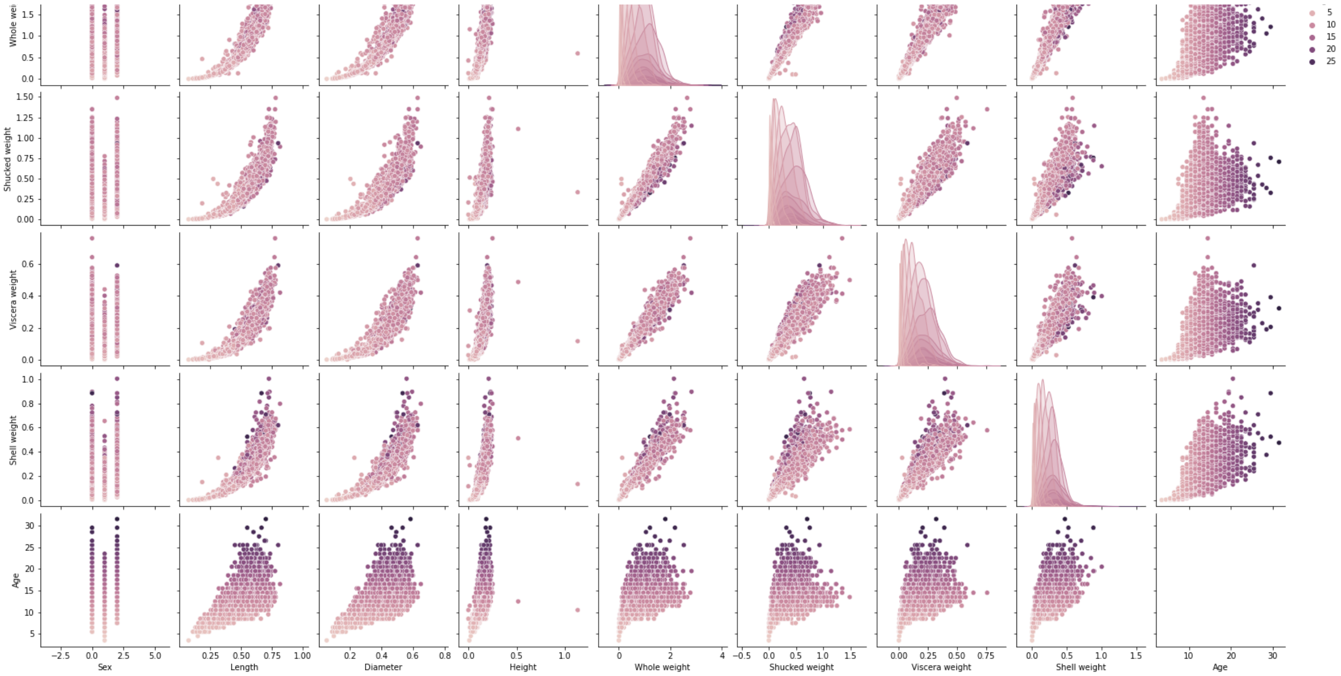
Univariate analysis

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

```
<seaborn.axisgrid.PairGrid at 0x7f1046734c50>
```


Sex
Length
Diameter
Height
Whole weight
Shucked weight
Viscera weight
Shell weight
Age
Rings
5
10
15
20
25


Sex
Length
Diameter
Height
ght
Rings


Whole wei
Shucked weight
Viscera weight
Shell weight
Age
Sex
Length
Diameter
Height
Whole weight
Shucked weight
Viscera weight
Shell weight
Age
5
10
15
20
25

```
df.describe()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	V
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.
mean	0.947091	0.523992	0.407881	0.139516	0.828742	0.359367	0.
std	0.822240	0.120093	0.099240	0.041827	0.490389	0.221963	0.
min	0.000000	0.075000	0.055000	0.000000	0.002000	0.001000	0.
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	0.

```
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()
```

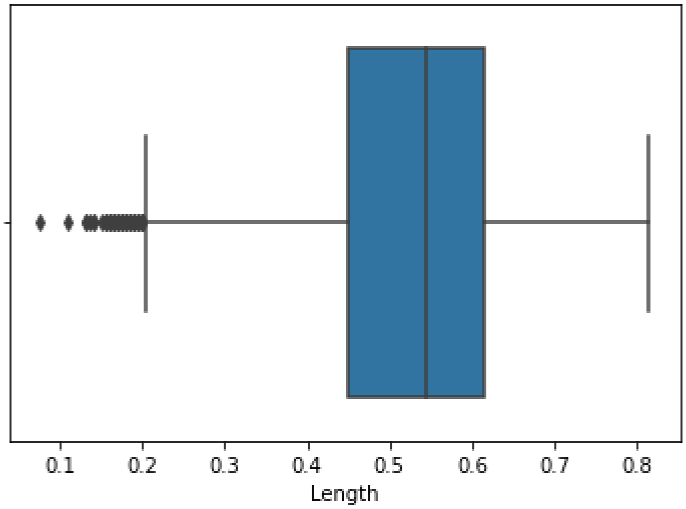
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	17.5
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	9.5
2	2	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	11.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	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.drop('Age',axis=1,inplace=True)
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

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1043f46890>
```



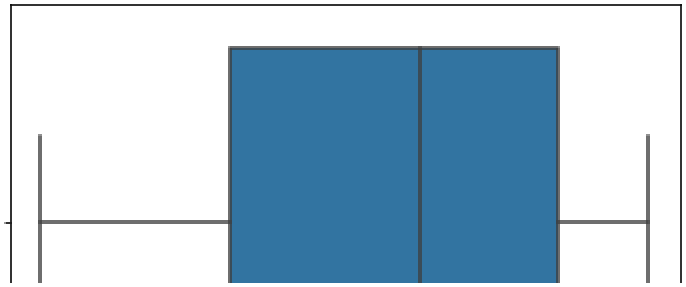
```
tenth_per = np.percentile(df['Length'], 10)
nine_per  = np.percentile(df['Length'], 90)

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 0x7f1043f9f350>
```



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

```
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'])
```

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

```
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'])
```

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

```
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'])
```

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

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

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

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

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

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

```
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'])
```

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

```
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'])
```

```
df.describe()
```

```
df.head()
```

Outlier treatment

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

```
df.head()
```

```
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
```

```
X_test.shape
```

```
y_train.shape
```

```
y_test.shape
```

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

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

```
y_pred1
```

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

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