Importing the libraries

In [ ]:

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

Loading of dataset

In [ ]:

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

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	0.1500	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7
...	...	...	...	...	...	...	...	...	...
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

4177 rows × 9 columns

In [ ]:

```
#modifying

data['Age'] = data.Rings + 1.5
data=data.rename(columns = {'Whole weight':'Whole_weight','Shucked weight': 'Shucked_we
ight','Viscera weight': 'Viscera_weight',
                            'Shell weight': 'Shell_weight'})
data
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_
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	
...	...	...	...	...	...	...	...	
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	

4177 rows × 10 columns

In [ ]:

```
data.head()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_wei
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0

Visualization

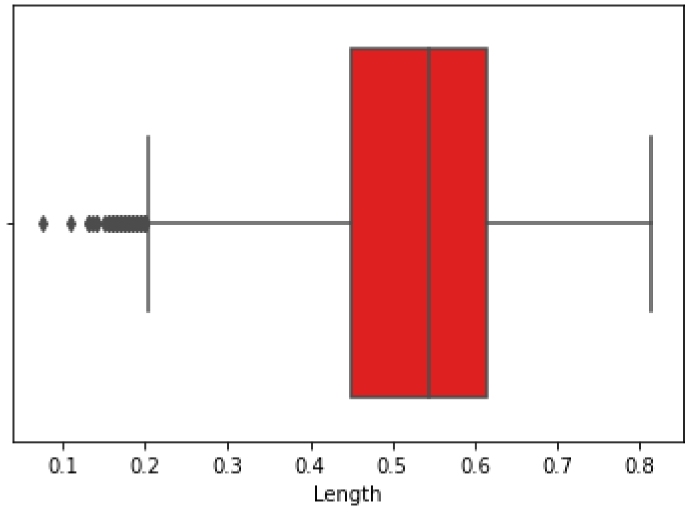
univarient analysis

In [ ]:

```
sns.boxplot(x=data.Length,color='red')
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88e2313c50>
```



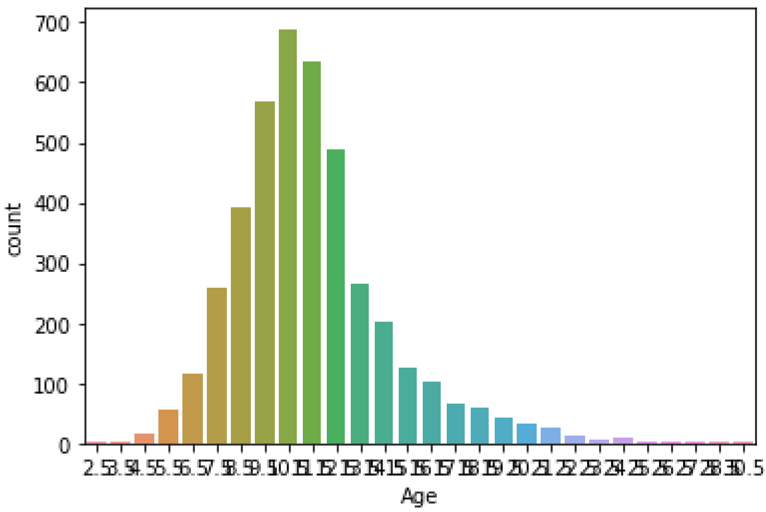
In [ ]:

```
sns.countplot(data['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argum
ents without an explicit keyword will result in an error or misinterpretat
ion.
  FutureWarning
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88e57d6cd0>
```

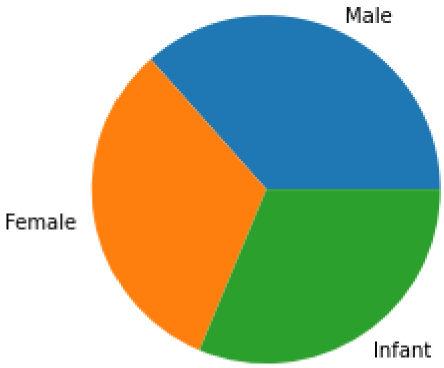


In [ ]:

```
plt.pie(data['Sex'].value_counts(),labels=['Male','Female','Infant'])
```

Out[ ]:

```
([<matplotlib.patches.Wedge at 0x7f88e217ce50>,
  <matplotlib.patches.Wedge at 0x7f88e218a350>,
  <matplotlib.patches.Wedge at 0x7f88e218a310>],
 [Text(0.45010440780275796, 1.0036961801643607, 'Male'),
  Text(-1.0848393519507589, -0.18199884741134378, 'Female'),
  Text(0.6099659291018239, -0.9153914820091724, 'Infant')])
```



In [ ]:

```
data.sum()
```

Out[ ]:

```
Sex               MMFMIIFFMFFMMFFMIFMMMIFFFFFMMMMFMFFMFFFMFFIIII...
Length                                                     2188.715
Diameter                                                    1703.72
Height                                                       582.76
Whole_weight                                               3461.656
Shucked_weight                                             1501.078
Viscera_weight                                             754.3395
Shell_weight                                               997.5965
Rings                                                         41493
Age                                                         47758.5
dtype: object
```

In [ ]:

```
data.mean()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_o
nly=None') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
  """Entry point for launching an IPython kernel.
```

Out[ ]:

```
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
Age               11.433684
dtype: float64
```

In [ ]:

```
data.median()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_o
nly=None') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
  """Entry point for launching an IPython kernel.
```

Out[ ]:

```
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
Age               10.5000
dtype: float64
```

bi-varient analysis

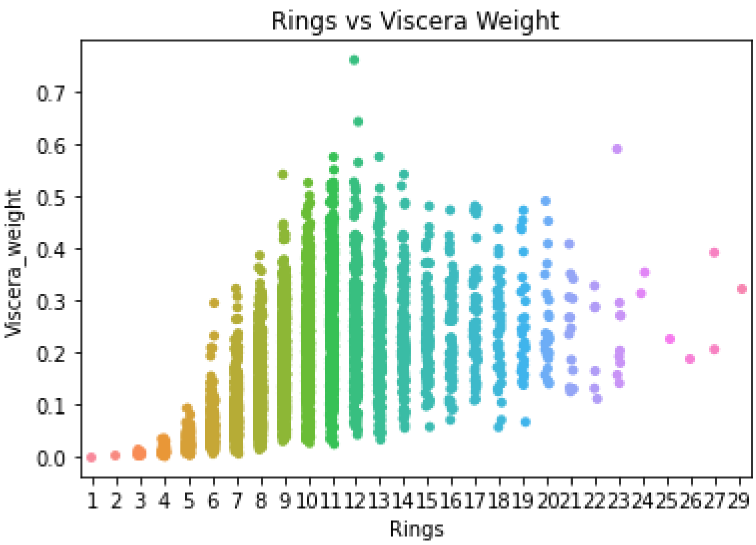
In [ ]:

```
sns.stripplot(data['Rings'], data['Viscera_weight'])
plt.title('Rings vs Viscera Weight')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWa
rning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpr
etation.
  FutureWarning
```

Out[ ]:

```
Text(0.5, 1.0, 'Rings vs Viscera Weight')
```



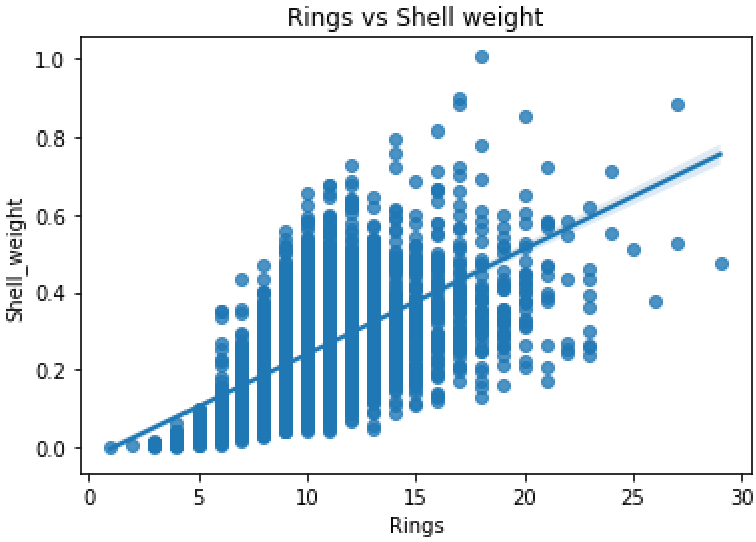
In [ ]:

```
sns.regplot(data['Rings'], data['Shell_weight'])
plt.title('Rings vs Shell weight')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWa
rning: Pass the following variables as keyword args: x, y. From version 0.
12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpr
etation.
  FutureWarning
```

Out[ ]:

```
Text(0.5, 1.0, 'Rings vs Shell weight')
```



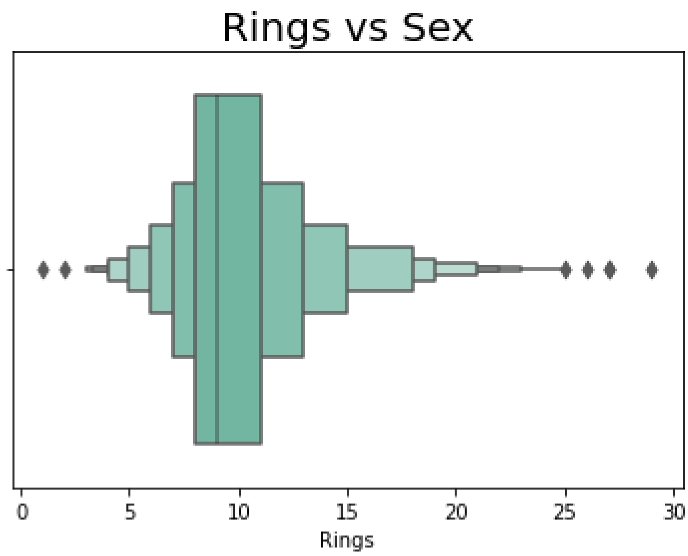
In [ ]:

```
sns.boxenplot(data['Rings'],hue = data['Sex'], palette = 'Set2')
plt.title('Rings vs Sex', fontsize = 20)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argum
ents without an explicit keyword will result in an error or misinterpretat
ion.
  FutureWarning
```

Out[ ]:

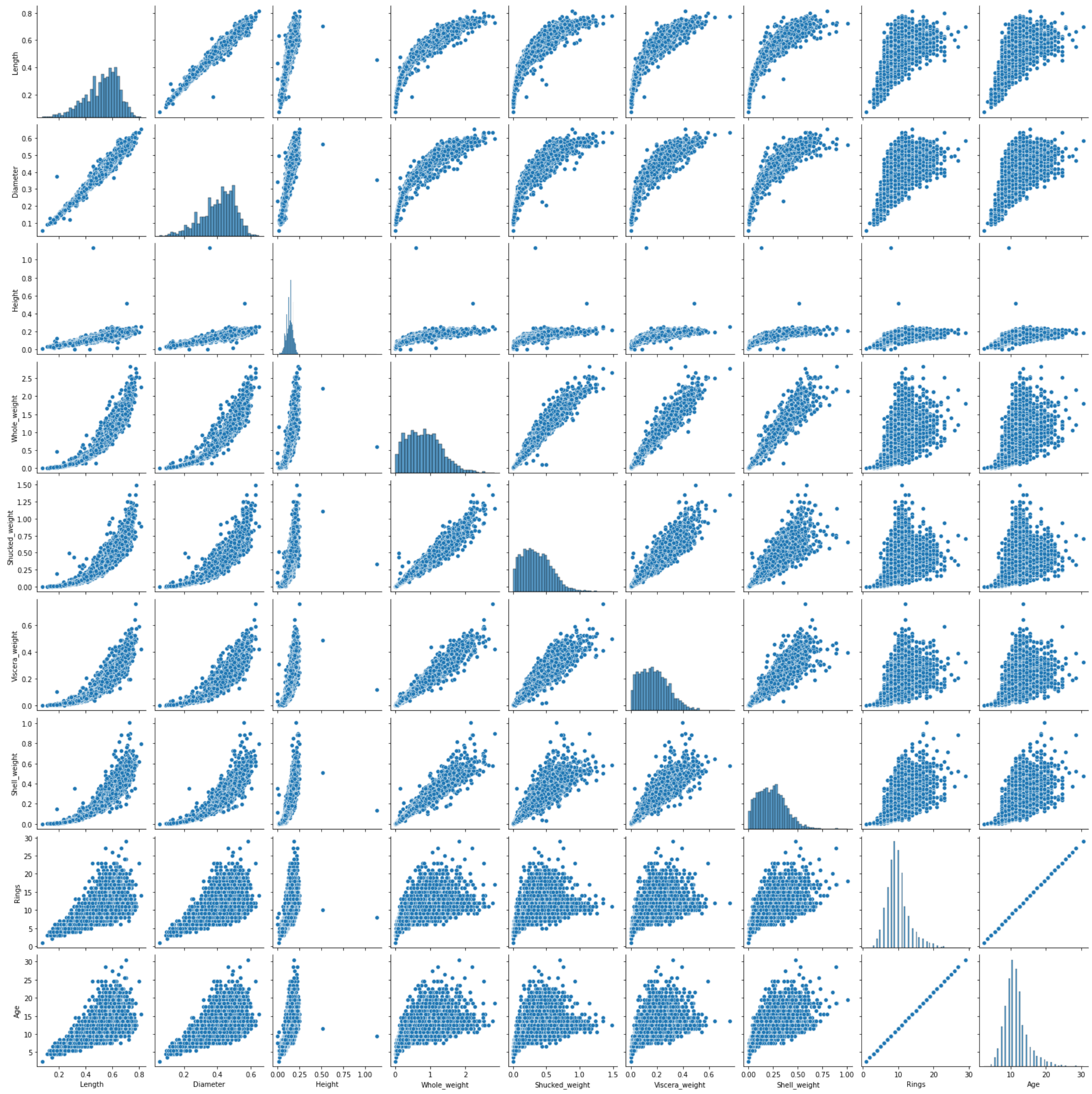
```
Text(0.5, 1.0, 'Rings vs Sex')
```



Multi-varient analysis

In [ ]:

```
sns.pairplot (data)
plt.show()
```



Descriptive statistics

In [ ]:

```
data.describe()
```

Out[ ]:

	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weigh
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.00000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.18059
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.10961
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.00050
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.09350
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.17100
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.25300
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.76000

Checking for null value

In [ ]:

```
data.isnull()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
4172	False	False	False	False	False	False	False	
4173	False	False	False	False	False	False	False	
4174	False	False	False	False	False	False	False	
4175	False	False	False	False	False	False	False	
4176	False	False	False	False	False	False	False	

4177 rows × 10 columns

In [ ]:

```
data.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
Age               0
dtype: int64
```

Encoding

In [ ]:

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
data.Sex = encoder.fit_transform(data.Sex)
data
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	
3	2	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	
...	...	...	...	...	...	...	...	
4172	0	0.565	0.450	0.165	0.8870	0.3700	0.2390	
4173	2	0.590	0.440	0.135	0.9660	0.4390	0.2145	
4174	2	0.600	0.475	0.205	1.1760	0.5255	0.2875	
4175	0	0.625	0.485	0.150	1.0945	0.5310	0.2610	
4176	2	0.710	0.555	0.195	1.9485	0.9455	0.3765	

4177 rows × 10 columns

Handling of Outliers

In [ ]:

```
outliers=data.quantile(q=(0.25,0.75))
outliers
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_v
0.25	0.0	0.450	0.35	0.115	0.4415	0.186	0.0935	
0.75	2.0	0.615	0.48	0.165	1.1530	0.502	0.2530	

In [ ]:

```
a = data.Age.quantile(0.25)
b = data.Age.quantile(0.75)
c = b - a
lower_limit = a - 1.5 * c
data.median(numeric_only=True)
```

Out[ ]:

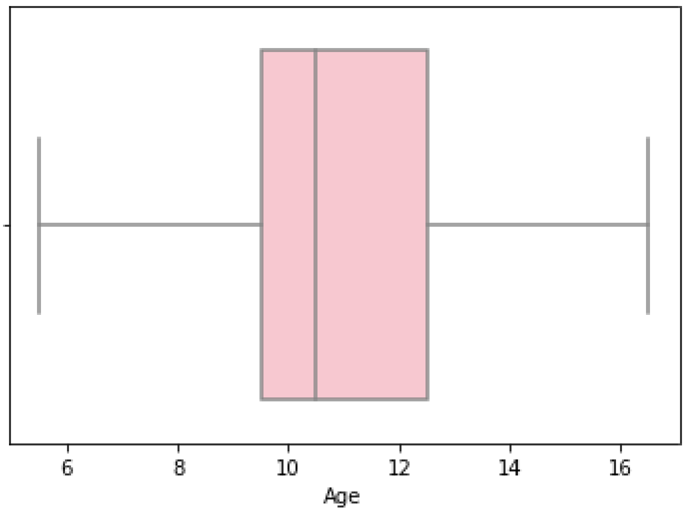
```
Sex                1.0000
Length             0.5450
Diameter           0.4250
Height             0.1400
Whole_weight       0.7995
Shucked_weight     0.3360
Viscera_weight     0.1710
Shell_weight       0.2340
Rings              9.0000
Age               10.5000
dtype: float64
```

In [ ]:

```
data['Age'] = np.where(data['Age'] < lower_limit, 7, data['Age'])
sns.boxplot(x=data.Age,showfliers = False,color = 'pink')
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88dd9f6bd0>
```



Independent & dependent variables

In [ ]:

```
x = data.iloc[:,:8]

y = data.iloc[:,8:]
```

In [ ]:

```
print("Independent variables\n",x.head(10))
```

```
Independent variables
    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
5     1   0.425     0.300   0.095        0.3515          0.1410
6     0   0.530     0.415   0.150        0.7775          0.2370
7     0   0.545     0.425   0.125        0.7680          0.2940
8     2   0.475     0.370   0.125        0.5095          0.2165
9     0   0.550     0.440   0.150        0.8945          0.3145

   Viscera_weight  Shell_weight
0          0.1010         0.150
1          0.0485         0.070
2          0.1415         0.210
3          0.1140         0.155
4          0.0395         0.055
5          0.0775         0.120
6          0.1415         0.330
7          0.1495         0.260
8          0.1125         0.165
9          0.1510         0.320
```

In [ ]:

```
print("Dependent variables\n",y.head(10))
```

```
Dependent variables
    Rings   Age
0     15  16.5
1      7   8.5
2      9  10.5
3     10  11.5
4      7   8.5
5      8   9.5
6     20  21.5
7     16  17.5
8      9  10.5
9     19  20.5
```

Feature scaling

In [ ]:

```
#standarization
from sklearn import preprocessing
standardScaler = preprocessing.StandardScaler()
new_x = standardScaler.fit_transform(x)
print("Standard scaling\n",new_x)
```

```
Standard scaling
 [[ 1.15198011 -0.57455813 -0.43214879 ... -0.60768536 -0.72621157
  -0.63821689]
 [ 1.15198011 -1.44898585 -1.439929   ... -1.17090984 -1.20522124
  -1.21298732]
 [-1.28068972  0.05003309  0.12213032 ... -0.4634999  -0.35668983
  -0.20713907]
 ...
 [ 1.15198011  0.6329849   0.67640943 ...  0.74855917  0.97541324
   0.49695471]
 [-1.28068972  0.84118198  0.77718745 ...  0.77334105  0.73362741
   0.41073914]
 [ 1.15198011  1.54905203  1.48263359 ...  2.64099341  1.78744868
   1.84048058]]
```

Spliting of data

In [ ]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y)
```

In [ ]:

```
x_train.shape
```

Out[ ]:

```
(3132, 8)
```

In [ ]:

```
y_train.shape
```

Out[ ]:

```
(3132, 2)
```

In [ ]:

```
x_test.shape
```

Out[ ]:

```
(1045, 8)
```

In [ ]:

```
y_test.shape
```

Out[ ]:

```
(1045, 2)
```

In [ ]:

```
x_test.head()
```

Out[ ]:

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_
3236	1	0.345	0.285	0.100	0.2225	0.0865	0.0580	
2774	0	0.580	0.425	0.150	0.8440	0.3645	0.1850	
2283	1	0.435	0.345	0.120	0.4475	0.2210	0.1120	
2547	1	0.230	0.180	0.050	0.0640	0.0215	0.0135	
795	2	0.515	0.405	0.145	0.6950	0.2150	0.1635	

In [ ]:

```
y_test.head()
```

Out[ ]:

	Rings	Age
3236	8	9.5
2774	9	10.5
2283	7	8.5
2547	5	6.5
795	15	16.5

Model building

In [ ]:

```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)
```

Out[ ]:

```
LinearRegression()
```

In [ ]:

```
#testing and training

from sklearn.metrics import r2_score
r2_score(mlr.predict(x_test),y_test)
```

Out[ ]:

```
0.04828394826189408
```