In [5]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler

import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew
```

In [6]:
```
df=pd.read_excel("assignment3.xlsx")
```

In [7]:
```
df.head()
```

Out[7]:

	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 [8]:
```
df.describe()
```

Out[8]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Ring
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.00000
mean	0.523492	0.407463	0.139384	0.826514	0.358380	0.180098	0.238203	9.92650
std	0.119970	0.099153	0.041801	0.490133	0.221747	0.109528	0.139205	3.22750
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.00000
25%	0.450000	0.350000	0.115000	0.441000	0.186000	0.093000	0.130000	8.00000
50%	0.545000	0.425000	0.140000	0.795500	0.334500	0.170000	0.230500	9.00000
75%	0.615000	0.480000	0.165000	1.149500	0.500000	0.252000	0.325500	11.00000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.00000

In [9]:
```
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 [10]:
```
df.shape
```

Out[10]: (4177, 9)

In [18]:
```
df['Age']= df['Rings']+1.5
dataset=df.drop('Rings', axis=1)
```

In [19]:
```
df.head()
```

Out[19]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	16.5
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	8.5
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	10.5
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	11.5
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	8.5

In [20]:
```
df.describe()
```

Out[20]:

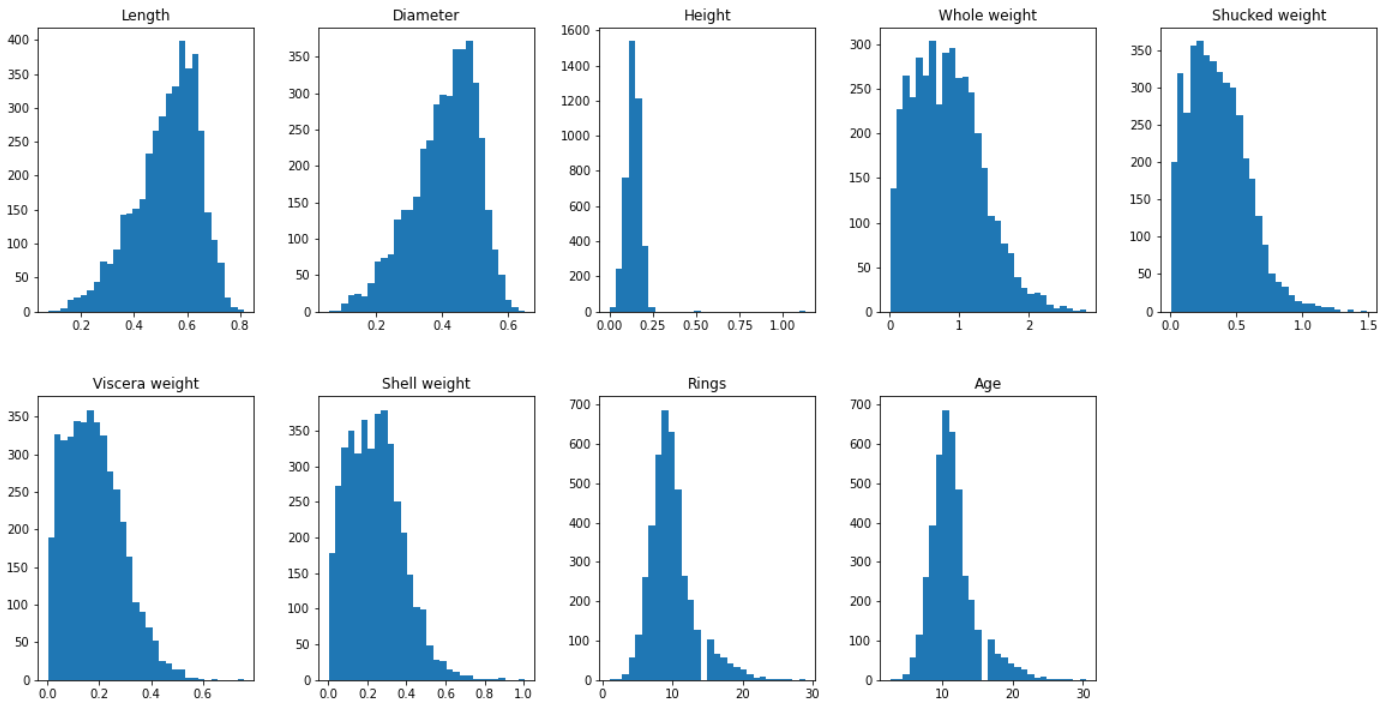
	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Ring
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.00000
mean	0.523492	0.407463	0.139384	0.826514	0.358380	0.180098	0.238203	9.92650
std	0.119970	0.099153	0.041801	0.490133	0.221747	0.109528	0.139205	3.22750
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.00000
25%	0.450000	0.350000	0.115000	0.441000	0.186000	0.093000	0.130000	8.00000
50%	0.545000	0.425000	0.140000	0.795500	0.334500	0.170000	0.230500	9.00000
75%	0.615000	0.480000	0.165000	1.149500	0.500000	0.252000	0.325500	11.00000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.00000

In [27]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix,accuracy_score
```

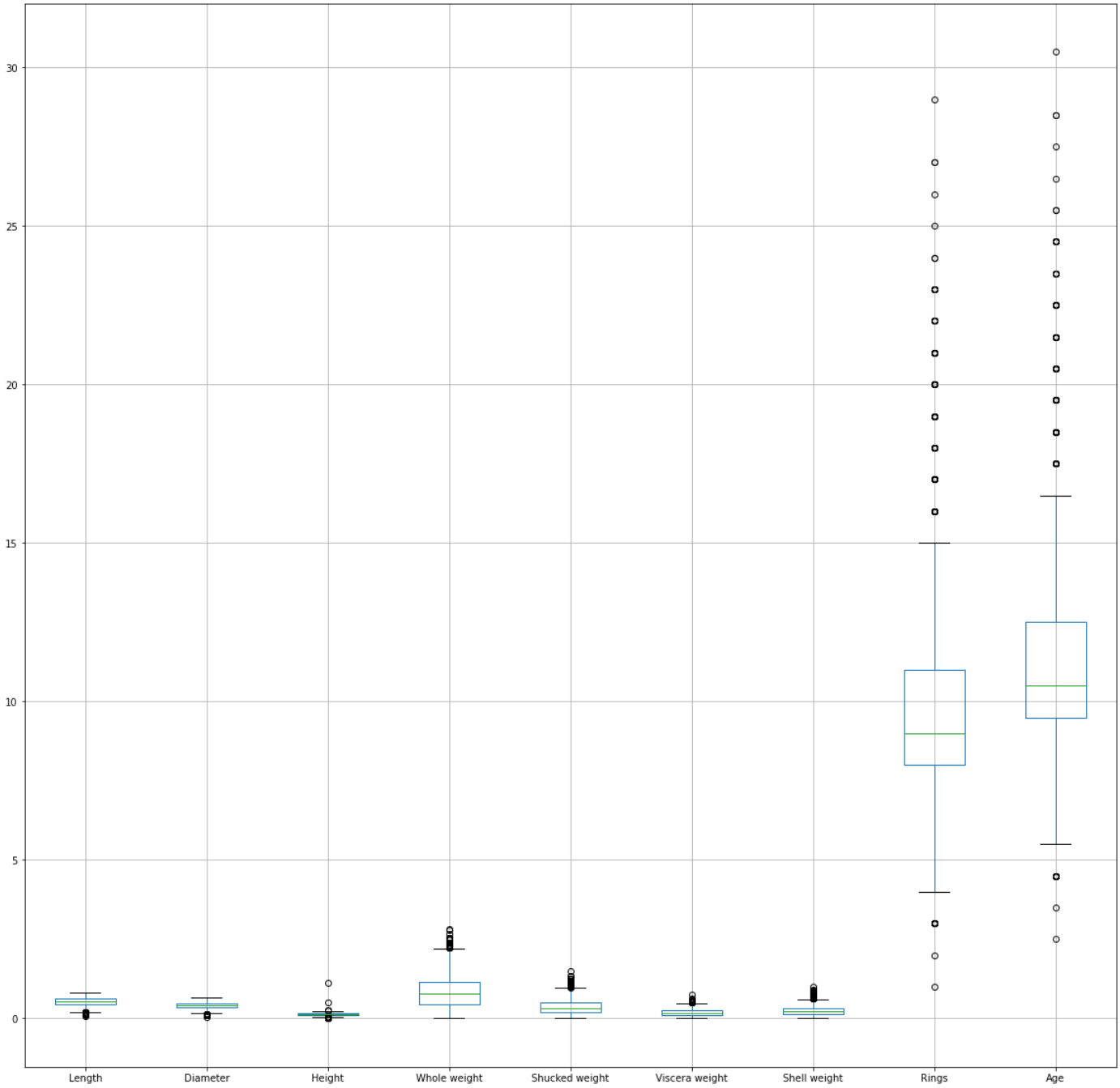
In [29]:
```
import seaborn as sns
```

In [32]:
```
import matplotlib.pyplot as plt
```

In [48]:
```
df.hist(figsize=(20,10), grid=False, layout=(2,5), bins=30)
plt.show()
```

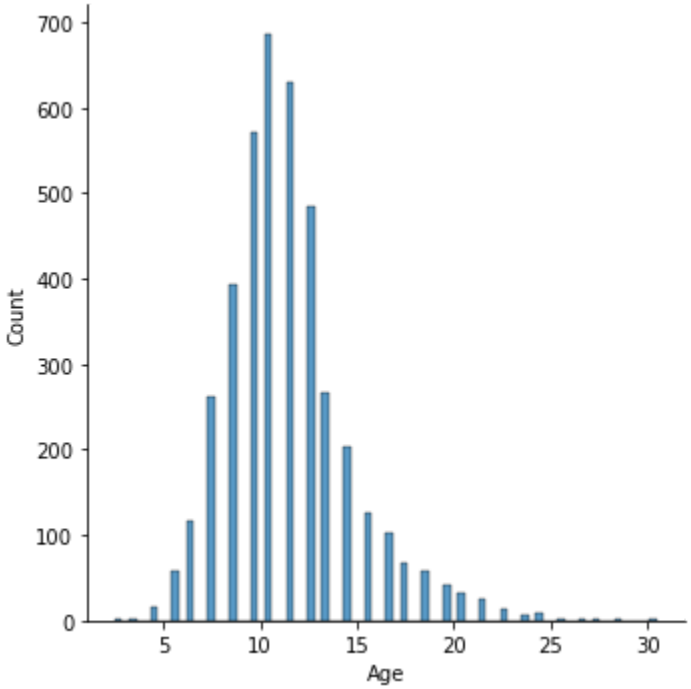


In [51]:
```
df.boxplot(figsize=(20,20))
plt.show()
```



In [52]:
```
plt.figure(figsize=(20,10))
sns.displot(dataset['Age'])
plt.show()
```

```
<Figure size 1440x720 with 0 Axes>
```

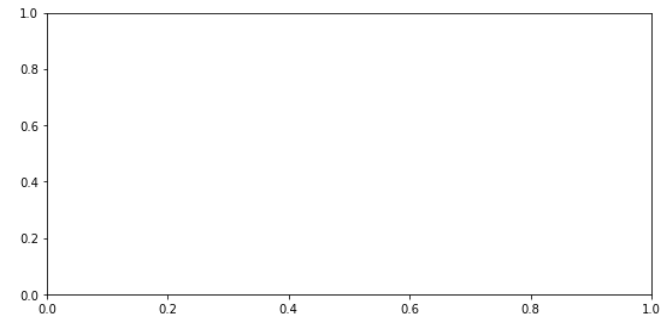


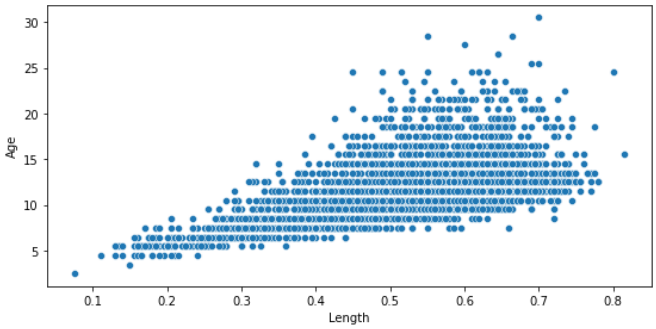
In [53]:
```
df.head()
```

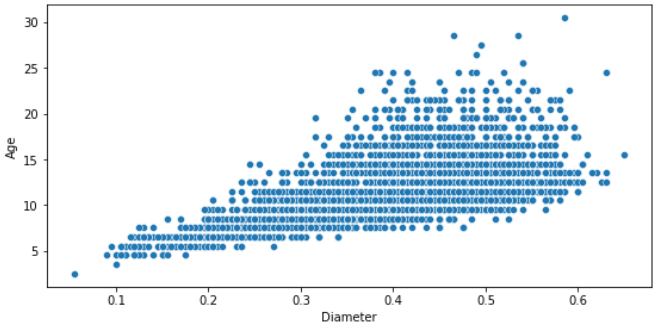
Out[53]:

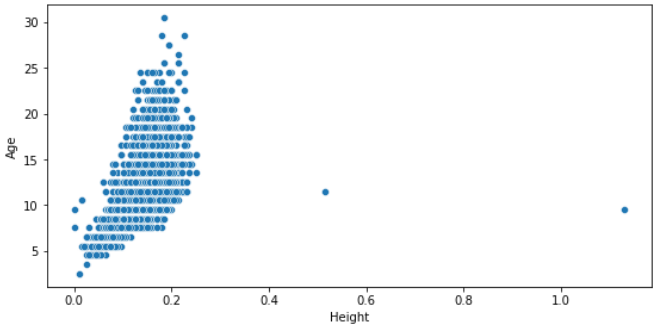
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	16.5
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	8.5
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	10.5
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	11.5
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	8.5

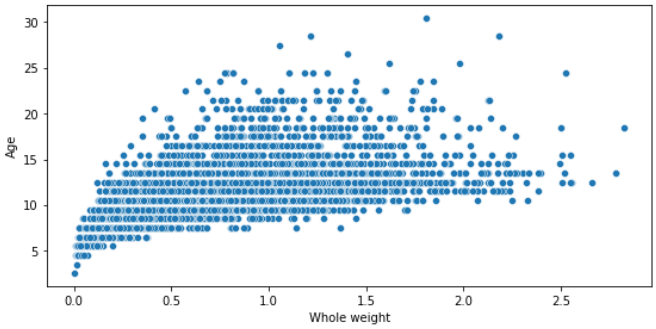
In [57]:
```
fig,axes=plt.subplots(4,2, figsize=(20,20))
axes=axes.flatten()
for i in range(1,len(dataset.columns)-1):
    sns.scatterplot(x=dataset.iloc[:,i],y=dataset['Age'],ax=axes[i])
plt.show()
```

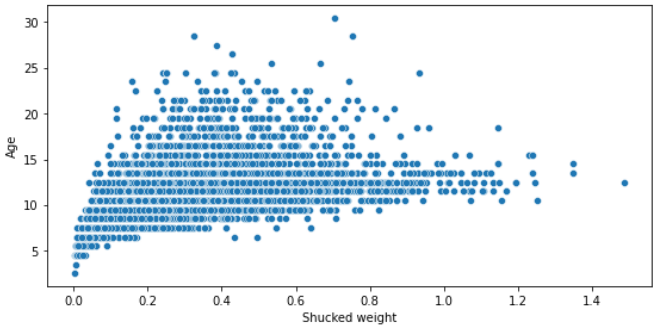


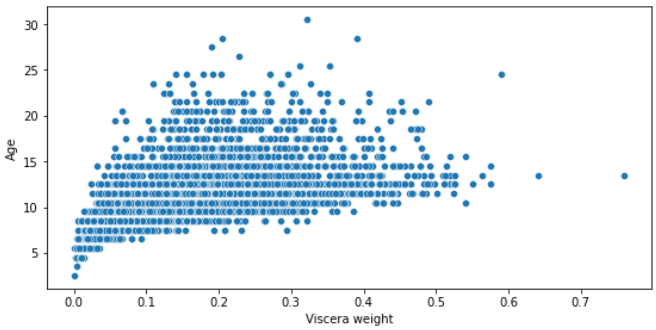


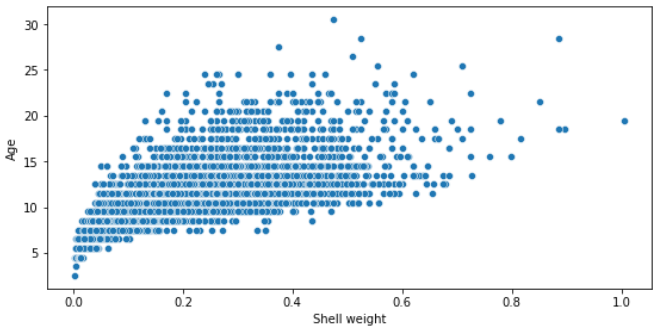




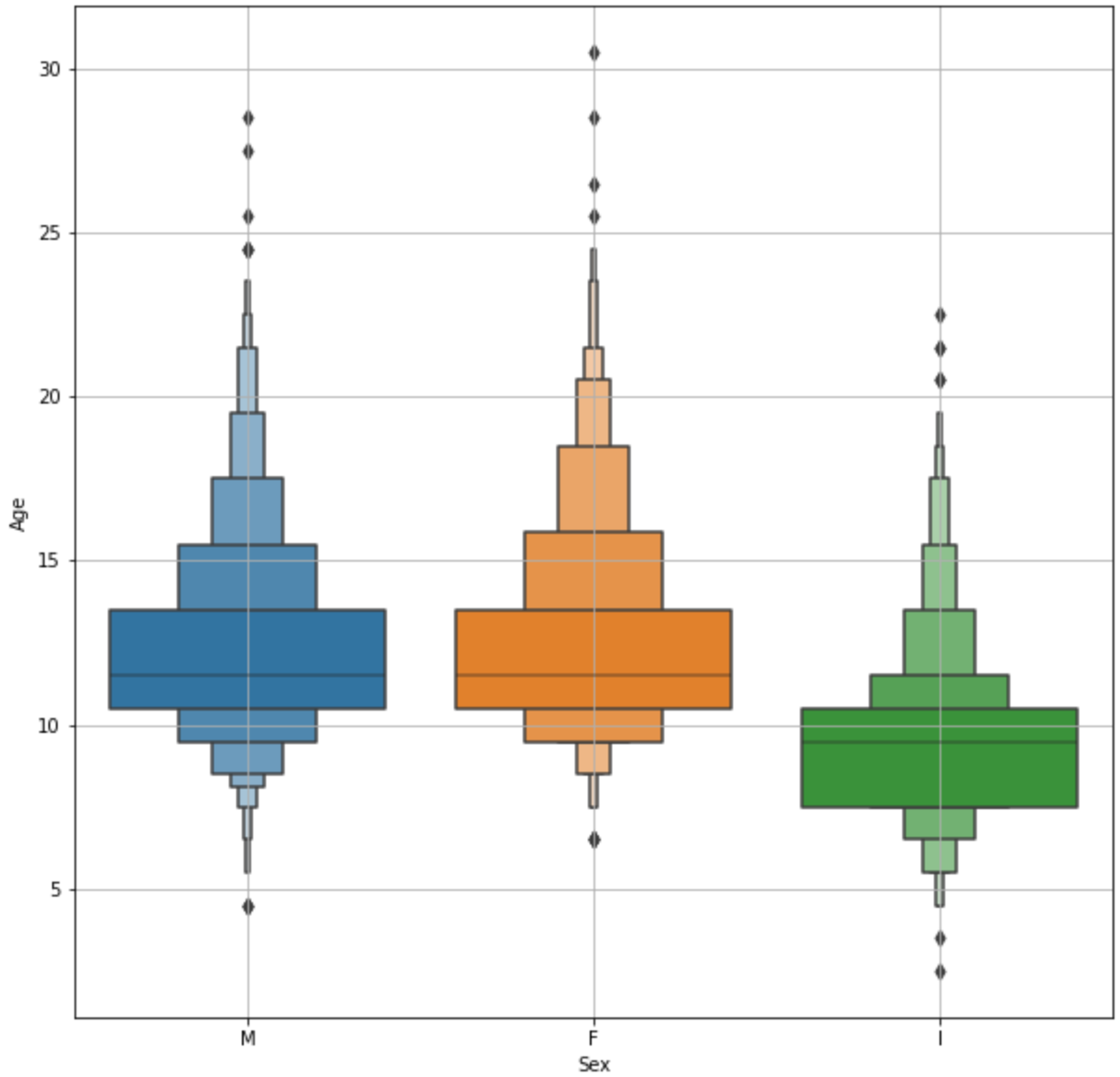








In [58]:
```
plt.figure(figsize=(10,10))
sns.boxenplot(y=dataset['Age'],x=dataset['Sex'])
plt.grid()
plt.show()
dataset.groupby('Sex')['Age'].describe()
```

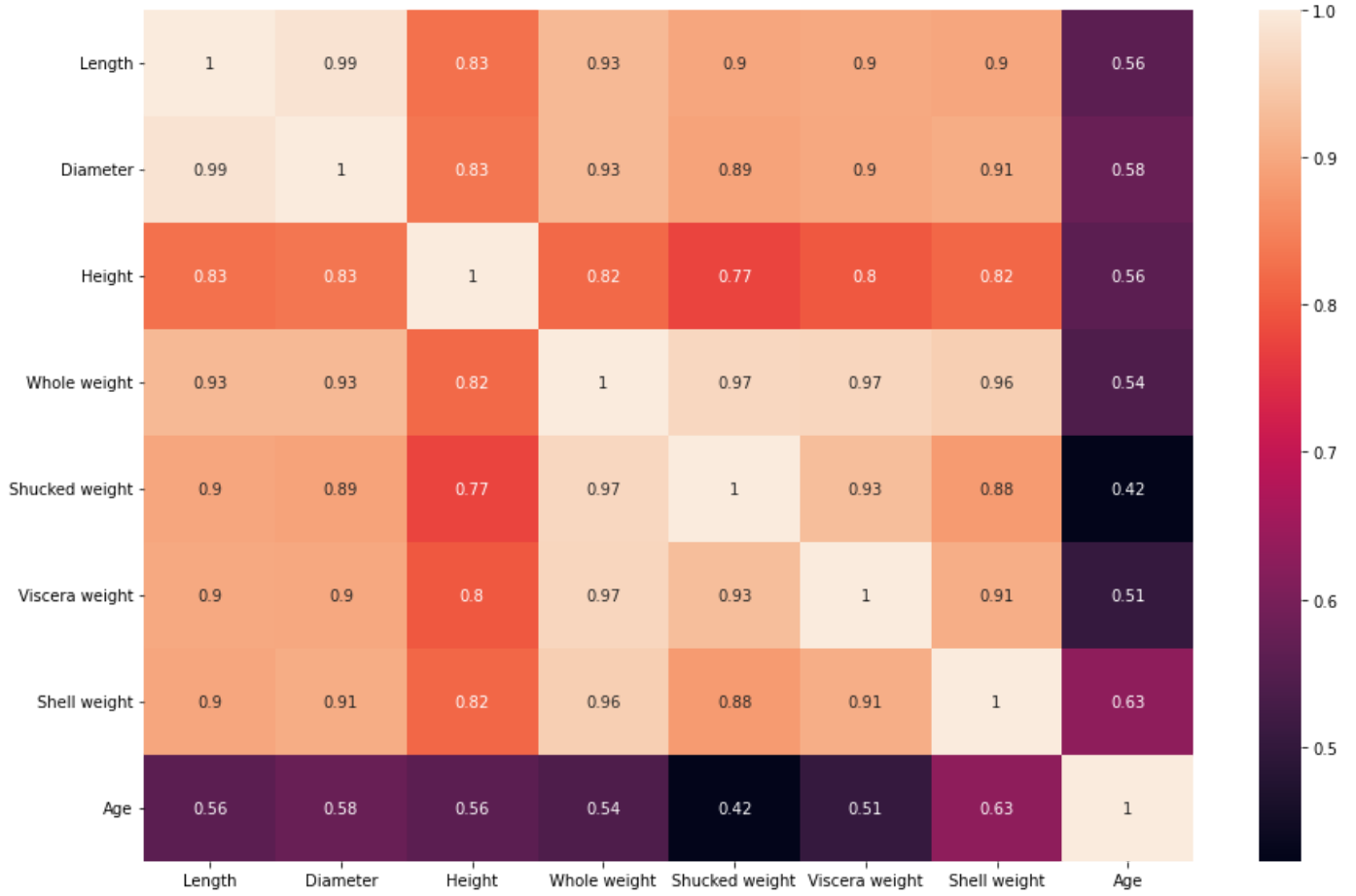


Out[58]:

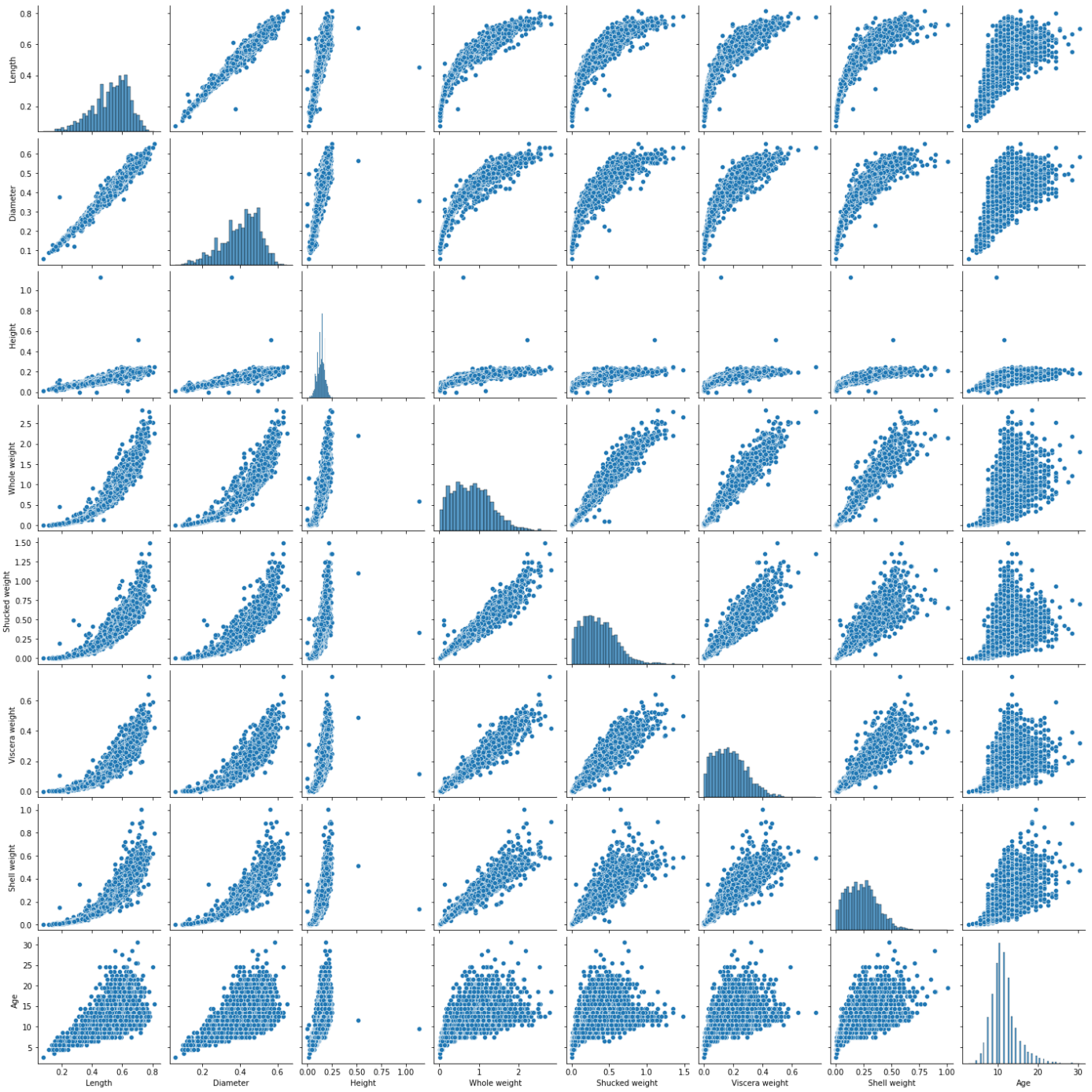
Sex	count	mean	std	min	25%	50%	75%	max
F	1302.0	12.634409	3.108279	6.5	10.5	11.5	13.5	30.5
I	1350.0	9.385926	2.506985	2.5	7.5	9.5	10.5	22.5
M	1525.0	12.201639	3.032549	4.5	10.5	11.5	13.5	28.5

In [61]:

```
plt.figure(figsize=(15,10))
sns.heatmap(dataset.corr(),annot=True)
plt.show()
```



In [62]:
```
sns.pairplot(dataset)
plt.show()
```



In [63]:
```
df.mean()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\3698961737.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.mean()
```

Out[63]:
```
Length             0.523492
Diameter           0.407463
Height             0.139384
Whole weight       0.826514
Shucked weight     0.358380
Viscera weight     0.180098
Shell weight       0.238203
Rings              9.926502
Age               11.426502
dtype: float64
```

In [64]:
```
df.mode()
```

Out[64]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	M	0.550	0.45	0.15	0.2225	0.175	0.1715	0.275	9.0	10.5
1	NaN	0.625	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [65]:
```
df.median()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\530051474.py:1: FutureWarni
ng: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is d
eprecated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  df.median()
```

Out[65]:
```
Length            0.5450
Diameter          0.4250
Height            0.1400
Whole weight      0.7955
Shucked weight    0.3345
Viscera weight    0.1700
Shell weight      0.2305
Rings             9.0000
Age              10.5000
dtype: float64
```

In [66]:
```
df.isna()
```

Out[66]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings	Age
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
4172	False	False	False	False	False	False	False	False	False	False
4173	False	False	False	False	False	False	False	False	False	False
4174	False	False	False	False	False	False	False	False	False	False
4175	False	False	False	False	False	False	False	False	False	False
4176	False	False	False	False	False	False	False	False	False	False

4177 rows × 10 columns

In [67]:
```
df.isna().any()
```

Out[67]:
```
Sex               False
Length            False
Diameter          False
Height            False
Whole weight      False
Shucked weight    False
Viscera weight    False
Shell weight      False
Rings             False
Age               False
dtype: bool
```

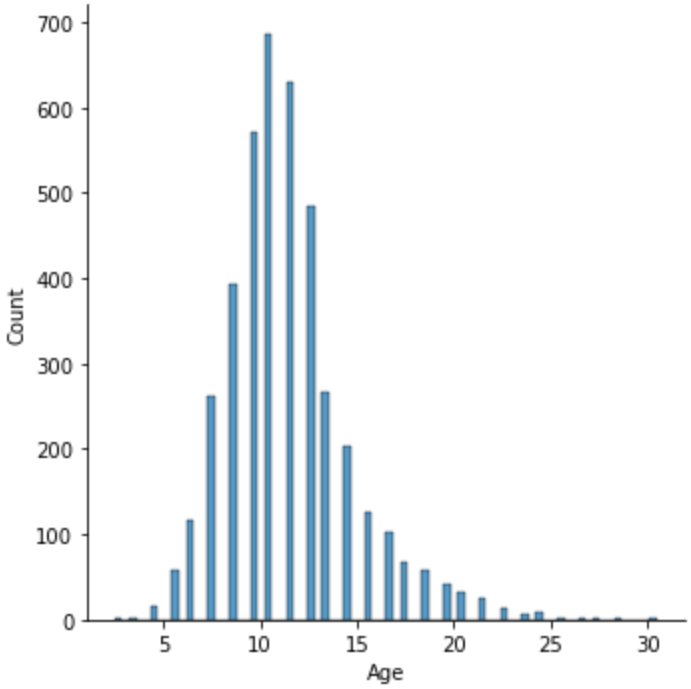
In [68]:
```
df.skew()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\1665899112.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  df.skew()
```

Out[68]:
```
Length           -0.632768
Diameter         -0.601756
Height            3.141725
Whole weight      0.541428
Shucked weight    0.729970
Viscera weight    0.602058
Shell weight      0.630726
Rings             1.114599
Age               1.114599
dtype: float64
```

In [70]:
```
print(sns.displot(dataset['Age']))
```

```
<seaborn.axisgrid.FacetGrid object at 0x000001774E38EA30>
```



In [71]:
```
df.kurt()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\1257127604.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  df.kurt()
```

Out[71]:
```
Length            0.063726
Diameter         -0.048134
Height           76.259325
Whole weight     -0.009623
Shucked weight    0.618376
Viscera weight    0.100616
Shell weight      0.542706
Rings             2.321858
Age               2.321858
dtype: float64
```

In [72]:
```
df.var()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\1568254755.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.var()
```

Out[72]:
```
Length             0.014393
Diameter           0.009831
Height             0.001747
Whole weight       0.240230
Shucked weight     0.049172
Viscera weight     0.011996
Shell weight       0.019378
Rings             10.416771
Age               10.416771
dtype: float64
```

In [73]:
```
df.std()
```

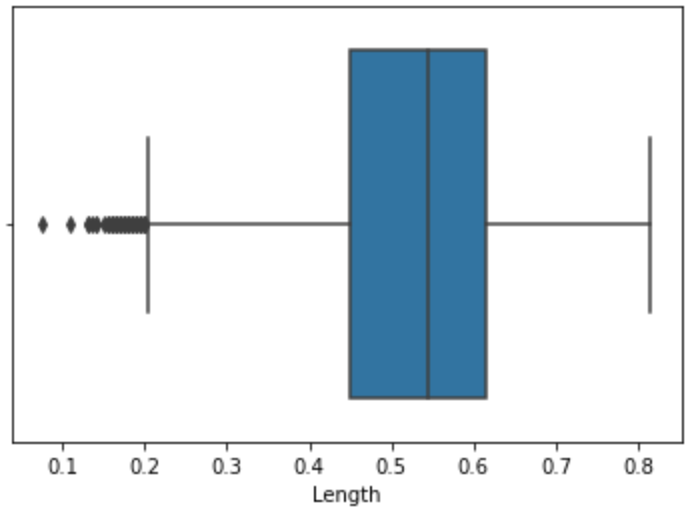
```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\3390915376.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.std()
```

Out[73]:
```
Length             0.119970
Diameter           0.099153
Height             0.041801
Whole weight       0.490133
Shucked weight     0.221747
Viscera weight     0.109528
Shell weight       0.139205
Rings              3.227502
Age                3.227502
dtype: float64
```

In [74]:
```
sns.boxplot(dataset['Length'])
```

```
C:\Users\kokila periyasamy\anaconda3\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[74]:
```
<AxesSubplot:xlabel='Length'>
```



In [75]:
```
qnt=dataset.quantile(q=(0.30,0.45))
qnt
```

Out[75]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Age
0.30	0.470	0.365	0.120	0.501	0.213	0.1075	0.15	9.5
0.45	0.525	0.410	0.135	0.720	0.302	0.1551	0.21	10.5

In [76]:
```
iqr=qnt.loc[0.45]-qnt.loc[0.30]
iqr
```

Out[76]:
```
Length            0.0550
Diameter          0.0450
Height            0.0150
Whole weight      0.2190
Shucked weight    0.0890
Viscera weight    0.0476
Shell weight      0.0600
Age               1.0000
dtype: float64
```

In [77]:
```
lower=qnt.loc[0.30]-1.5*iqr
lower
```

Out[77]:
```
Length            0.3875
Diameter          0.2975
Height            0.0975
Whole weight      0.1725
Shucked weight    0.0795
Viscera weight    0.0361
Shell weight      0.0600
Age               8.0000
dtype: float64
```

In [78]:
```
upper=qnt.loc[0.30]+1.5*iqr
upper
```

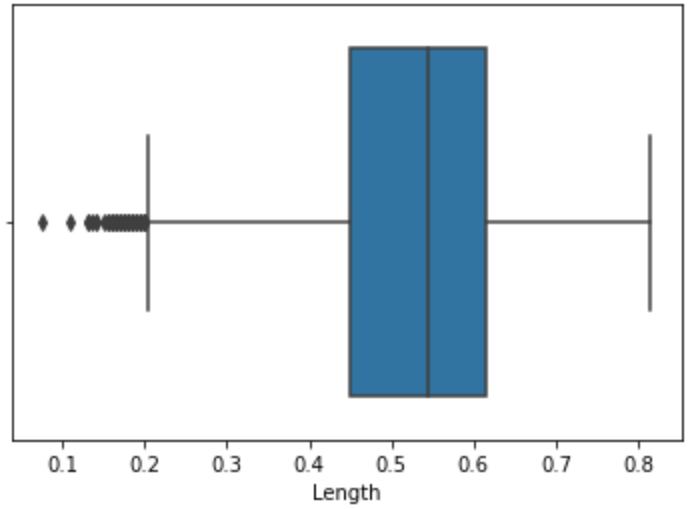
Out[78]:
```
Length             0.5525
Diameter           0.4325
Height             0.1425
Whole weight       0.8295
Shucked weight     0.3465
Viscera weight     0.1789
Shell weight       0.2400
Age               11.0000
dtype: float64
```

In [85]:
```
dataset['Length']=np.where(dataset['Length']>45,31,dataset['Length'])
```

In [82]:
```
sns.boxplot(dataset['Length'])
```

```
C:\Users\kokila periyasamy\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12, the only va
lid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[82]:
```
<AxesSubplot:xlabel='Length'>
```



In [86]:
```
df=pd.read_excel("assignment3.xlsx")
```

In [87]:
```
df.head()
```

Out[87]:

	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 [88]:
```
dataset['Sex'].replace( {'F':1,'M':0},inplace=True)

dataset.head()
```

Out[88]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Age
0	0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	16.5
1	0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	8.5
2	1	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	10.5
3	0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	11.5
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	8.5

In [89]:
```
y=dataset['Height']
y.head()
```

Out[89]:
```
0    0.095
1    0.090
2    0.135
3    0.125
4    0.080
Name: Height, dtype: float64
```

In [90]:
```
x=dataset.drop(columns=['Height'],axis=1)

x.head()
```

Out[90]:

	Sex	Length	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Age
0	0	0.455	0.365	0.5140	0.2245	0.1010	0.150	16.5
1	0	0.350	0.265	0.2255	0.0995	0.0485	0.070	8.5
2	1	0.530	0.420	0.6770	0.2565	0.1415	0.210	10.5
3	0	0.440	0.365	0.5160	0.2155	0.1140	0.155	11.5
4	I	0.330	0.255	0.2050	0.0895	0.0395	0.055	8.5

In [91]:
```
dataset=pd.get_dummies(dataset,columns=['Height'])

dataset.head()
```

Out[91]:

	Sex	Length	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Age	Height_0.0	Height_0.01	...	Height_0.21	H
0	0	0.455	0.365	0.5140	0.2245	0.1010	0.150	16.5	0	0	...	0	
1	0	0.350	0.265	0.2255	0.0995	0.0485	0.070	8.5	0	0	...	0	
2	1	0.530	0.420	0.6770	0.2565	0.1415	0.210	10.5	0	0	...	0	
3	0	0.440	0.365	0.5160	0.2155	0.1140	0.155	11.5	0	0	...	0	
4	I	0.330	0.255	0.2050	0.0895	0.0395	0.055	8.5	0	0	...	0	

5 rows × 59 columns

In [92]:
```
dataset = pd.get_dummies(dataset, drop_first=True)
dataset.head()
```

Out[92]:

	Length	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Age	Height_0.0	Height_0.01	Height_0.015	...	Height
0	0.455	0.365	0.5140	0.2245	0.1010	0.150	16.5	0	0	0	...	
1	0.350	0.265	0.2255	0.0995	0.0485	0.070	8.5	0	0	0	...	
2	0.530	0.420	0.6770	0.2565	0.1415	0.210	10.5	0	0	0	...	
3	0.440	0.365	0.5160	0.2155	0.1140	0.155	11.5	0	0	0	...	
4	0.330	0.255	0.2050	0.0895	0.0395	0.055	8.5	0	0	0	...	

5 rows × 60 columns

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

In [ ]:
```
x=scale(x)
```

In [93]:
```
x
```

Out[93]:

	Sex	Length	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Age
0	0	0.455	0.365	0.5140	0.2245	0.1010	0.1500	16.5
1	0	0.350	0.265	0.2255	0.0995	0.0485	0.0700	8.5
2	1	0.530	0.420	0.6770	0.2565	0.1415	0.2100	10.5
3	0	0.440	0.365	0.5160	0.2155	0.1140	0.1550	11.5
4	I	0.330	0.255	0.2050	0.0895	0.0395	0.0550	8.5
...	...	...	...	...	...	...	...	...
4172	1	0.565	0.450	0.8870	0.3700	0.2390	0.2490	12.5
4173	0	0.590	0.440	0.9660	0.4390	0.2145	0.2605	11.5
4174	0	0.600	0.475	1.1760	0.5255	0.2875	0.3080	10.5
4175	1	0.625	0.485	1.0945	0.5310	0.2610	0.2960	11.5
4176	0	0.710	0.555	1.9485	0.9455	0.3765	0.4950	13.5

4177 rows × 8 columns

In [94]:
```
x.mean()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\3791599204.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  x.mean()
```

Out[94]:
```
Length             0.523492
Diameter           0.407463
Whole weight       0.826514
Shucked weight     0.358380
Viscera weight     0.180098
Shell weight       0.238203
Age               11.426502
dtype: float64
```

In [95]:
```
x.std()
```

```
C:\Users\kokila periyasamy\AppData\Local\Temp\ipykernel_4964\2178744708.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select only valid columns be
fore calling the reduction.
  x.std()
```

Out[95]:
```
Length            0.119970
Diameter          0.099153
Whole weight      0.490133
Shucked weight    0.221747
Viscera weight    0.109528
Shell weight      0.139205
Age               3.227502
dtype: float64
```

In [103…
```
from sklearn.model_selection import train_test_split
```

In [104…
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [105…
```
x_train.shape
```

Out[105]:
```
(3341, 8)
```

In [106…
```
x_test.shape
```

Out[106]:
```
(836, 8)
```

In [107…
```
y_train.shape
```

Out[107]:
```
(3341,)
```

In [108…
```
y_test.shape
```

Out[108]:
```
(836,)
```

bulid the model

In [109…
```
from sklearn.tree import DecisionTreeClassifier
```

model=DecisionTreeClassifier()

In [110…
```
dataset=pd.get_dummies(dataset,drop_first=True)
```

In [111…
```
dataset.head()
```

Out[111]:

	Length	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Age	Height_0.0	Height_0.01	Height_0.015	...	Heigl
0	0.455	0.365	0.5140	0.2245	0.1010	0.150	16.5	0	0	0	...	
1	0.350	0.265	0.2255	0.0995	0.0485	0.070	8.5	0	0	0	...	
2	0.530	0.420	0.6770	0.2565	0.1415	0.210	10.5	0	0	0	...	
3	0.440	0.365	0.5160	0.2155	0.1140	0.155	11.5	0	0	0	...	
4	0.330	0.255	0.2050	0.0895	0.0395	0.055	8.5	0	0	0	...	

5 rows × 60 columns

In [112…
```
X  = dataset.drop('Height_1.13',axis=1)
y = dataset['Height_1.13']

from sklearn.model_selection import train_test_split
X_train, X_test, y_test, y_test = train_test_split(X, y, test_size=0.33)

from sklearn.preprocessing import StandardScaler
ss = StandardScaler()

X_trains = ss.fit_transform(X_train)
X_tests = ss.transform(X_test)
```

In [114…
```
from sklearn.feature_selection import RFE
lr = LinearRegression()
n = [{'n_features_to_select':list(range(1,10))}]
rfe = RFE(lr)

from sklearn.model_selection import GridSearchCV
gsearch = GridSearchCV(rfe, param_grid=n, cv=3)
gsearch.fit(X, y)

gsearch.best_params_
```

Out[114]: {'n_features_to_select': 1}

In [115…
```
lr=LinearRegression()
rf = RFE(lr,n_features_to_select=8)
rf.fit(X,y)
pd.DataFrame(rf.ranking_,  index=X.columns, columns=['Class'])
```

Out[115]:

	Class
Length	45
Diameter	47
Whole weight	49
Shucked weight	46
Viscera weight	48
Shell weight	44
Age	52
Height_0.0	36
Height_0.01	40
Height_0.015	42
Height_0.02	38
Height_0.025	43
Height_0.03	41
Height_0.035	39
Height_0.04	34
Height_0.045	14
Height_0.05	30
Height_0.055	35
Height_0.06	32
Height_0.065	11
Height_0.07	29
Height_0.075	1
Height_0.08	28
Height_0.085	1
Height_0.09	23
Height_0.095	37
Height_0.1	1
Height_0.105	13
Height_0.11	3
Height_0.115	19
Height_0.12	1
Height_0.125	26
Height_0.13	1
Height_0.135	18
Height_0.14	10
Height_0.145	24
Height_0.15	22
Height_0.155	2
Height_0.16	1

	Class
Height_0.165	7
Height_0.17	9
Height_0.175	8
Height_0.18	6
Height_0.185	1
Height_0.19	5
Height_0.195	1
Height_0.2	4
Height_0.205	12
Height_0.21	16
Height_0.215	20
Height_0.22	27
Height_0.225	15
Height_0.23	25
Height_0.235	17
Height_0.24	21
Height_0.25	33
Height_0.515	31
Sex_1	50
Sex_I	51

In [ ]: