

Assignment -3

Team ID	PNT2022TMID15174
Student Name	Charan C.D
Student Roll Number	7179KCTKCTKCTKCTKCT19BEC130
Maximum Marks	2 Marks

Input

```
import numpy as np
import pandas as pd

df=pd.read_csv('abalone.csv')
df.head()
```

Output

	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

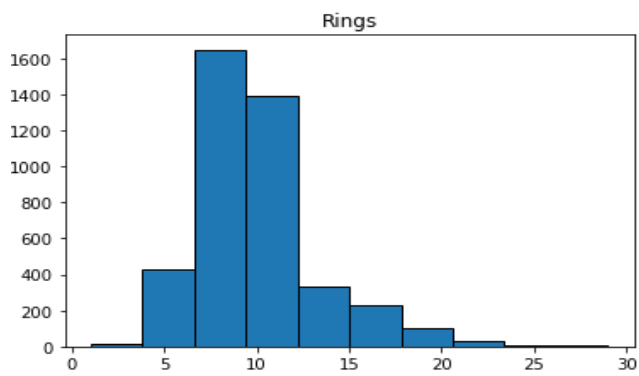
Input

```
import matplotlib.pyplot as plt
import seaborn as sns

df.hist(column='Rings',grid=False,edgecolor='black')
```

Output

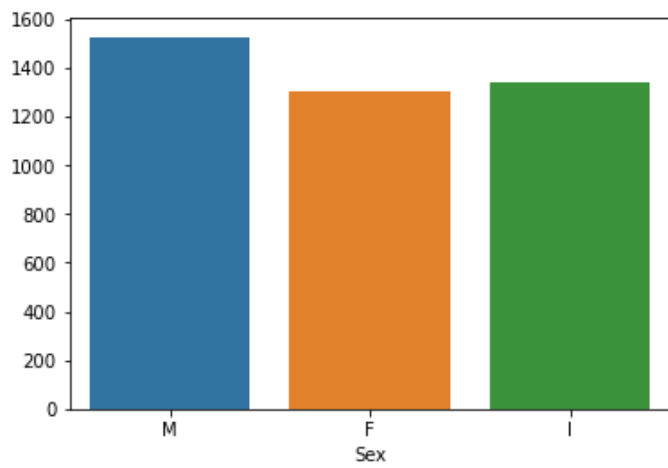
```
array([],
      dtype=object)
```



Input

```
sns.countplot(x='Sex',data=df)
```

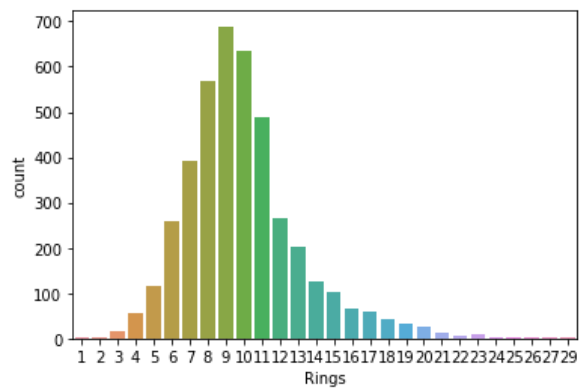
Output



Input

```
sns.countplot(x="Rings",data=df)
```

Output



Input

```
df.describe()
```

Output

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	8.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	9.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	11.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000

Input

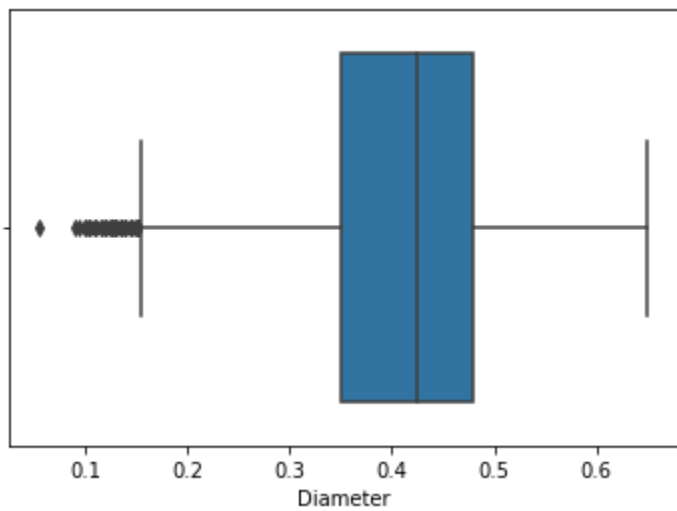
```
df.info()
```

```
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
```

Input

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

Output



Input

```
outliers=np.where(df['Rings']>3)
```

```
outliers
```

Output

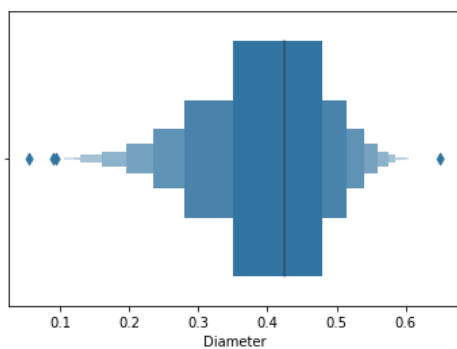
```
(array([ 0, 1, 2, ..., 4174, 4175, 4176], dtype=int64),)
```

Input

```
for i in outliers:  
    df['Rings'][i]=3
```

```
sns.boxenplot(df['Diameter'])
```

Output



Input

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df["Length"]=le.fit_transform(df["Length"])
df['Diameter']=le.fit_transform(df['Diameter'])
df['Height']=le.fit_transform(df['Height'])
```

```
df.head()
```

Output

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	66	56	18	0.5140	0.2245	0.1010	0.150	3
1	M	45	36	17	0.2255	0.0995	0.0485	0.070	3
2	F	81	67	26	0.6770	0.2565	0.1415	0.210	3
3	M	63	56	24	0.5160	0.2155	0.1140	0.155	3
4	I	41	34	15	0.2050	0.0895	0.0395	0.055	3

Input

```
x=df.iloc[:, 1:9]
y=df.iloc[:, 8]
```

```
x=x.drop(['Length','Height'],axis=1)
x
```

Output

	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	56	0.5140	0.2245	0.1010	0.1500	3
1	36	0.2255	0.0995	0.0485	0.0700	3
2	67	0.6770	0.2565	0.1415	0.2100	3
3	56	0.5160	0.2155	0.1140	0.1550	3
4	34	0.2050	0.0895	0.0395	0.0550	3
...
4172	73	0.8870	0.3700	0.2390	0.2490	3
4173	71	0.9660	0.4390	0.2145	0.2605	3
4174	78	1.1760	0.5255	0.2875	0.3080	3
4175	80	1.0945	0.5310	0.2610	0.2960	3
4176	94	1.9485	0.9455	0.3765	0.4950	3

4177 rows × 6 columns

Input

```
y
```

Output

```
0      3
```

```
1      3
2      3
3      3
4      3
..
4172   3
4173   3
4174   3
4175   3
4176   3
Name: Rings, Length: 4177, dtype: int64
```

Input

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=0)

xtrain.shape
xtest.shape
```

Output

```
(1254, 6)
```

Input

```
from sklearn.linear_model import LinearRegression

l=LinearRegression()
l.fit(xtrain,ytrain)
```

Output

```
LinearRegression()
```

Input

```
ypred=l.predict(xtest)

ypred
```

Output

```
array([3.00036065, 3.00036065, 3.00036065, ..., 3.00036065, 3.00036065,
       3.00036065])
```

Input

```
from sklearn import metrics  
  
metrics.mean_absolute_error(ytest,ypred)
```

Output

```
0.0006075048987421218
```

Input

```
metrics.mean_squared_error(ytest,ypred)
```

Output

```
7.67236335876925e-05
```

Input

```
np.sqrt(metrics.mean_squared_error(ytest,ypred))
```

Output

```
0.008759202794072786
```