In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('abalone.csv')
```

In [3]:
```
data
```

Out[3]:

	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 [4]:
```
data.isnull().sum()
```

Out[4]:
```
Sex               0
Length            0
Diameter          0
Height            0
Whole weight      0
Shucked weight    0
Viscera weight    0
Shell weight      0
Rings             0
dtype: int64
```

In [5]:
```
data.columns
```

Out[5]:
```
Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'Rings'],
      dtype='object')
```

In [6]:
```
data.describe()
```

Out[6]:

	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

In [7]:
```
data.Rings = data['Rings'].astype('float')
```

In [8]:
```
data.dtypes
```

Out[8]:
```
Sex                object
Length            float64
Diameter          float64
Height            float64
Whole weight      float64
Shucked weight    float64
Viscera weight    float64
Shell weight      float64
Rings             float64
dtype: object
```

In [9]:
```
data.hist(figsize=(15,15), grid = True, layout = (3,3), bins = 30)
plt.show()
```

In [10]:
```
data.Rings.describe()
```

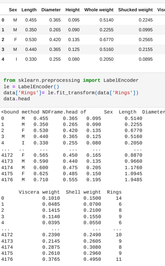
Out[10]:
```
count    4177.000000
mean        9.933684
std         3.224169
min         1.000000
25%         8.000000
50%         9.000000
75%        11.000000
max        29.000000
Name: Rings, dtype: float64
```

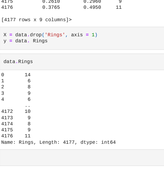
In [11]:
```
plt.figure(figsize = (20,10))
sns.heatmap(data.corr(), annot = True)
```

Out[11]:
```
<AxesSubplot:>
```

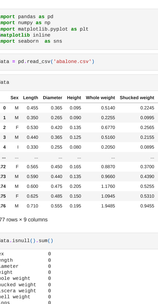
In [12]:
```
plt.figure(figsize=(10,10))
sns.jointplot(data=data, x='Rings', y='Height', kind='reg');
sns.jointplot(data=data, x='Rings', y='Shell weight', kind='reg');
```

Out[12]:
```
<Figure size 720x720 with 0 Axes>
```



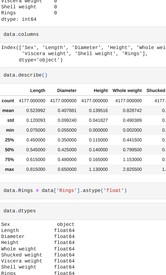


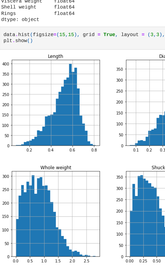
In [13]:
```
plt.figure(figsize=(7, 7))
sns.boxplot(data=data, x='Sex', y='Rings');
```



In [40]:
```
plt.figure(figsize=(20, 5))
sns.lmplot(data=data, x='Rings', y='Height', hue='Sex', fit_reg=True);
sns.lmplot(data=data, x='Rings', y='Shell weight', hue='Sex', fit_reg=True);
```

Out[40]:
```
<Figure size 1440x360 with 0 Axes>
```





In [14]:
```
data.head()
```

Out[14]:

	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	15.0
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7.0
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9.0
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10.0
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7.0

In [15]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['Rings']= le.fit_transform(data['Rings'])
data.head
```

Out[15]:
```
<bound method NDFrame.head of       Sex  Length  Diameter  Height  Whole weight  Shucked weight  \
0       M   0.455     0.365   0.095        0.5140          0.2245
1       M   0.350     0.265   0.090        0.2255          0.0995
2       F   0.530     0.420   0.135        0.6770          0.2565
3       M   0.440     0.365   0.125        0.5160          0.2155
4       I   0.330     0.255   0.080        0.2050          0.0895
...   ...     ...       ...     ...           ...             ...
4172    F   0.565     0.450   0.165        0.8870          0.3700
4173    M   0.590     0.440   0.135        0.9660          0.4390
4174    M   0.600     0.475   0.205        1.1760          0.5255
4175    F   0.625     0.485   0.150        1.0945          0.5310
4176    M   0.710     0.555   0.195        1.9485          0.9455

      Viscera weight  Shell weight  Rings
0             0.1010        0.1500     14
1             0.0485        0.0700      6
2             0.1415        0.2100      8
3             0.1140        0.1550      9
4             0.0395        0.0550      6
...              ...           ...    ...
4172          0.2390        0.2490     10
4173          0.2145        0.2605      9
4174          0.2875        0.3080      8
4175          0.2610        0.2960      9
4176          0.3765        0.4950     11

[4177 rows x 9 columns]>
```

In [16]:
```
X = data.drop('Rings', axis = 1)
y = data. Rings
```

In [17]:
```
data.Rings
```

Out[17]:
```
0       14
1        6
2        8
3        9
4        6
       ...
4172    10
4173     9
4174     8
4175     9
4176    11
Name: Rings, Length: 4177, dtype: int64
```

In [ ]:
```

```