In [227]:
```
import pandas as pd

dataset = pd.read_csv("/content/abalone.csv")
df=pd.DataFrame(dataset)
df
```

Out[227]:

	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 [228]:
```
from warnings import filterwarnings
filterwarnings("ignore")
```

In [229]:
```
df.columns
```

Out[229]:
```
Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'Rings'],
      dtype='object')
```

In [230]:
```
df.dtypes
```

Out[230]:
```
Sex               object
Length           float64
Diameter         float64
Height           float64
Whole weight     float64
Shucked weight   float64
Viscera weight   float64
Shell weight     float64
Rings              int64
dtype: object
```

In [231]:
```
df.describe()
```

Out[231]:

	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 [232]:
```
df.shape
```

Out[232]:
```
(4177, 9)
```

In [233]:
```
df.isnull().sum()
```

Out[233]:
```
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 [234]:
```
df.duplicated()
```

Out[234]:
```
0       False
1       False
2       False
3       False
4       False
        ...
4172    False
4173    False
4174    False
4175    False
4176    False
Length: 4177, dtype: bool
```

In [235]:
```
df.duplicated().sum()
```

Out[235]:
```
0
```

In [236]:
```
df.head()
```

Out[236]:

	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

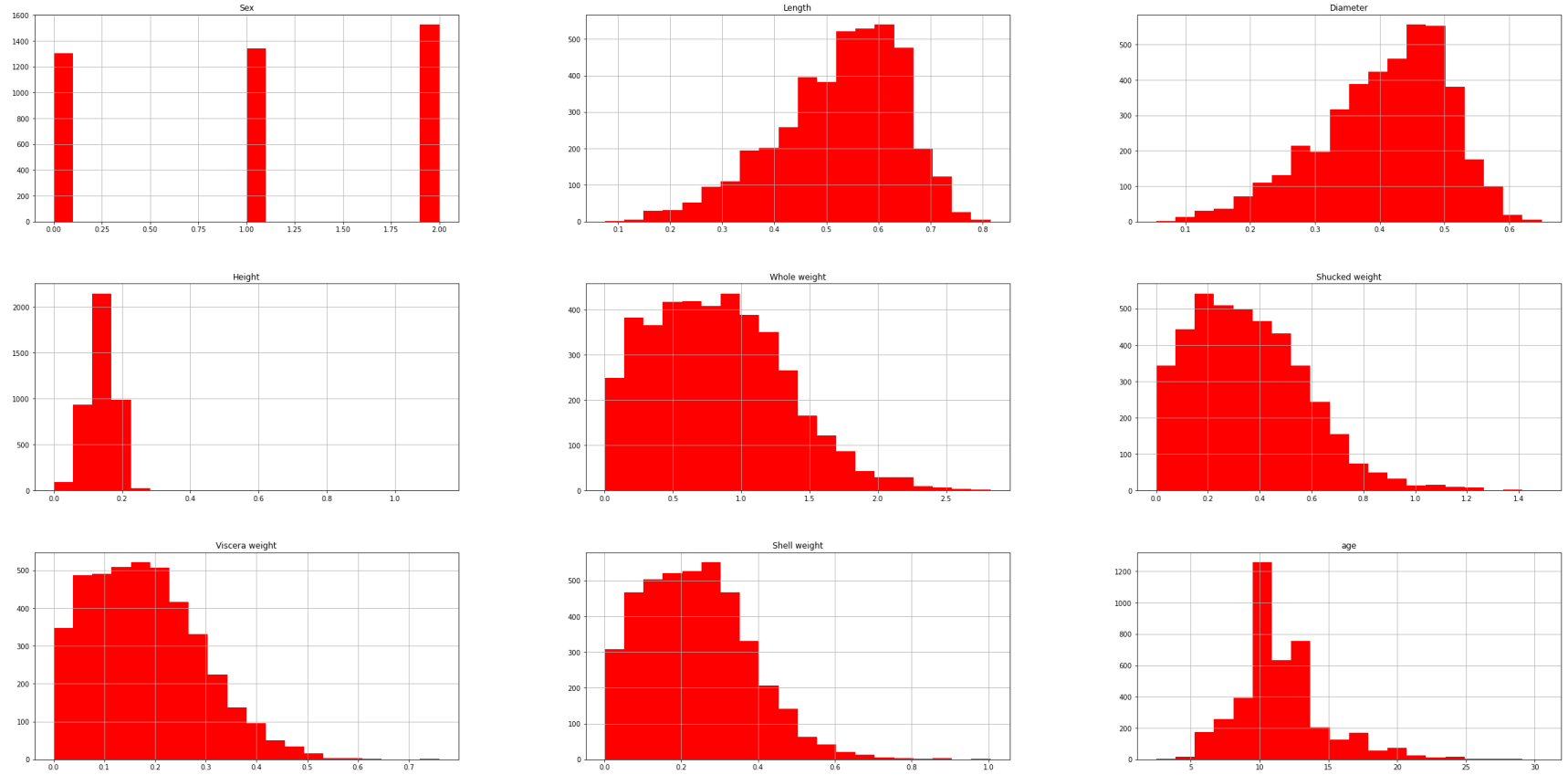
In [237]:
```
df.tail()
```

Out[237]:

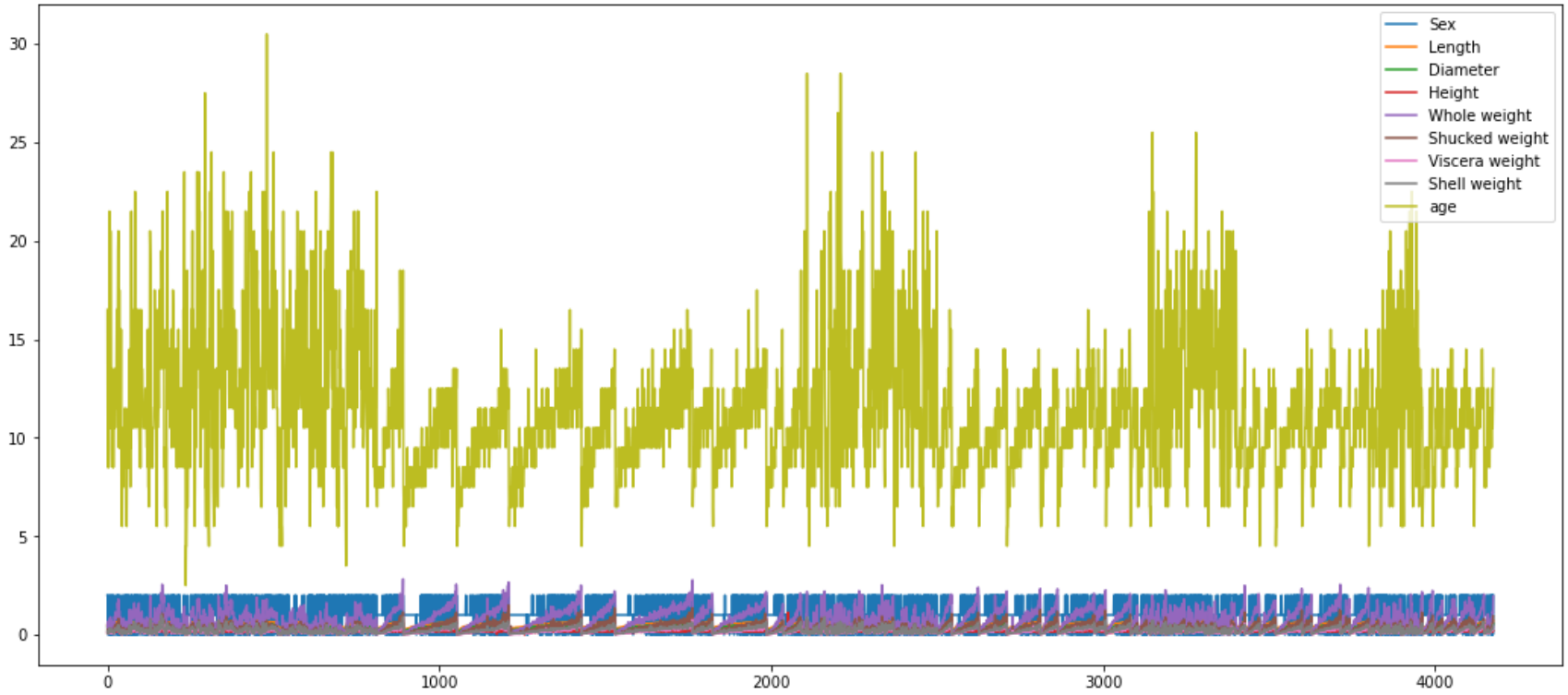
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
4172	F	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	11
4173	M	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	10
4174	M	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	9
4175	F	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	10
4176	M	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	12

In [16]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [17]:
```
df.hist(bins=20,figsize=(40,20),color='r')
plt.show()
```

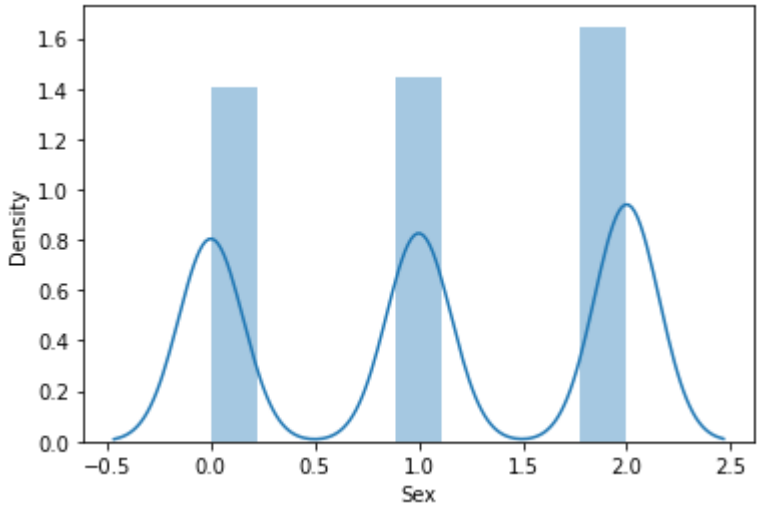


In [221]:
```
df.plot(figsize=(18, 8))
plt.show()
```



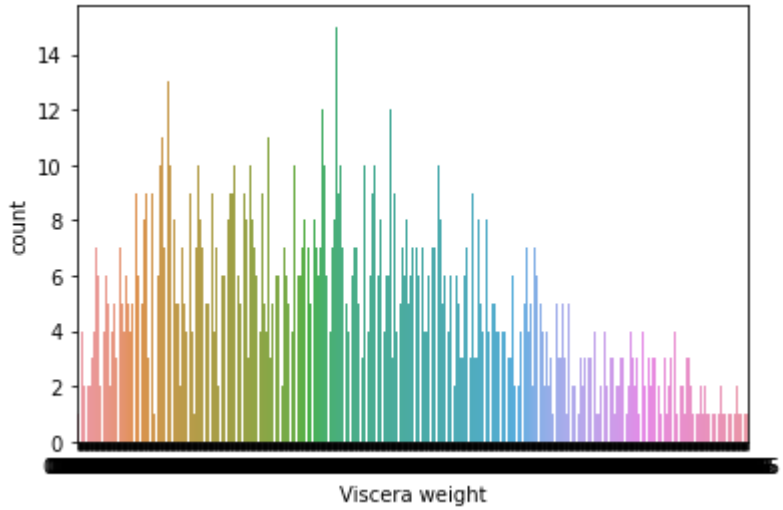
In [222]:
```
sns.distplot(df["Sex"])
```

Out[222]:
```
<AxesSubplot:xlabel='Sex', ylabel='Density'>
```

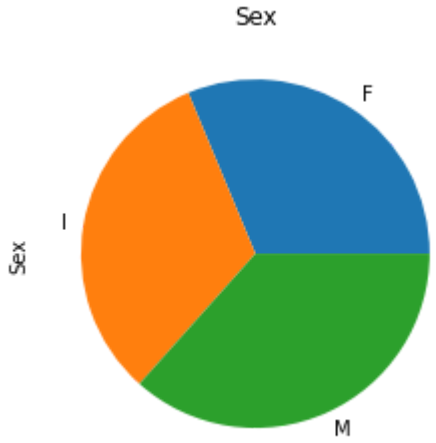


In [245]:
```
sns.countplot(x="Viscera weight",data=df)
```

Out[245]:
```
<AxesSubplot:xlabel='Viscera weight', ylabel='count'>
```

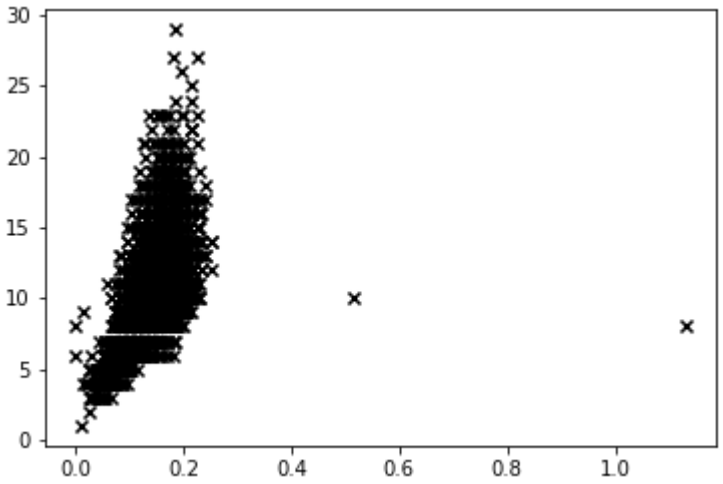


In [243]:
```
df.groupby('Sex').Sex.count().plot(kind='pie')
plt.title('Sex')
plt.show()
```



In [241]:
```
plt.scatter(df['Height'] , df['Rings'], c='k', marker='x')
```

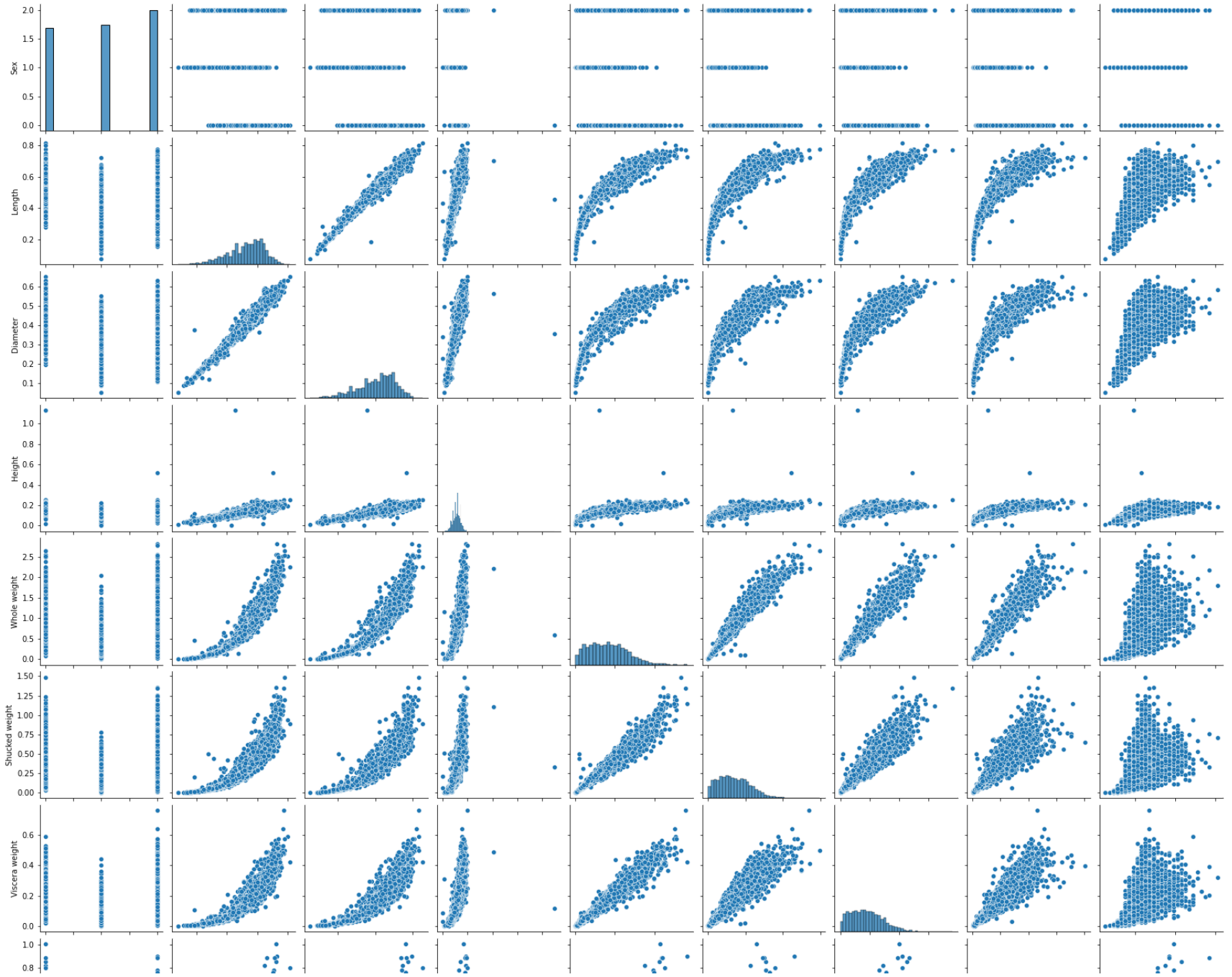
Out[241]:
```
<matplotlib.collections.PathCollection at 0x1d3d61a3eb0>
```

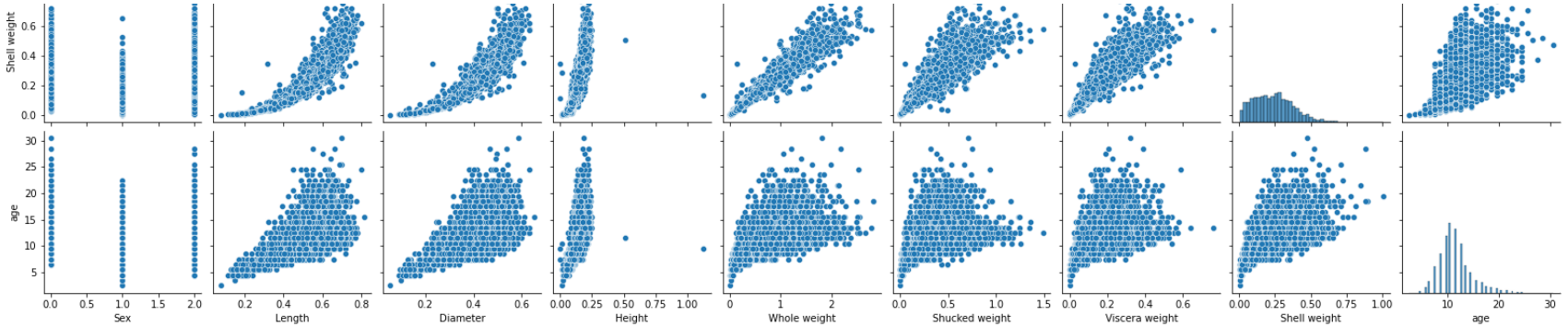


In [239]:
```
sns.countplot(x='Sex',hue='Whole weight',data=df)
```

Out[239]:
```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```

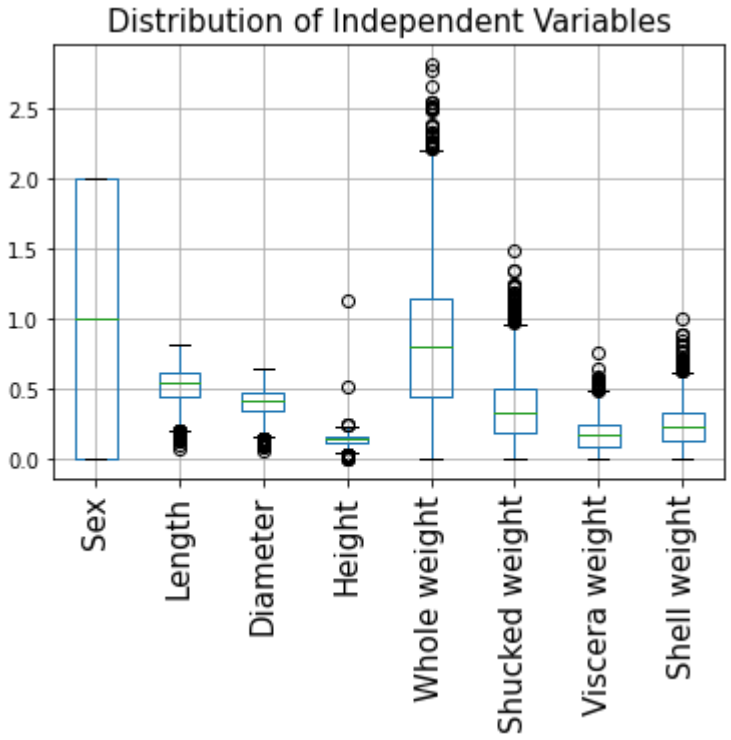
In [19]:
```
sns.pairplot(df)
plt.show()
```





In [21]:
```
cols = ['Sex','Length','Diameter','Height','Whole weight','Shucked weight','Viscera weight','Shell weight']
df[cols].boxplot()
plt.title('Distribution of Independent Variables', fontsize = 15)
plt.xticks(rotation = 'vertical', fontsize = 15)

plt.show()
```



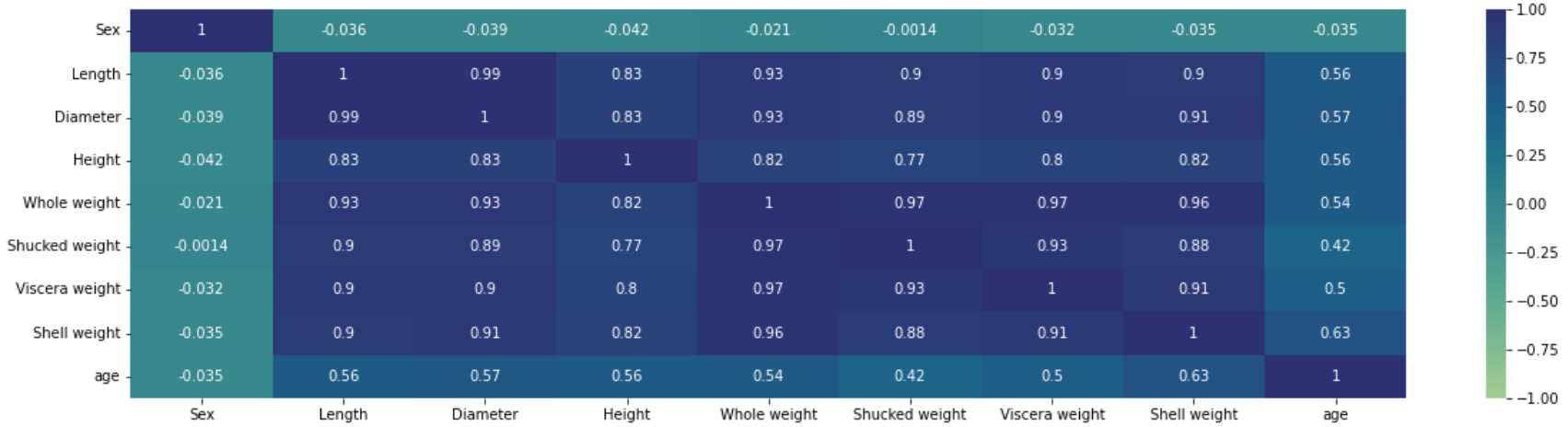
In [23]:
```
corr=df.corr()
corr
```

Out[23]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age
Sex	1.000000	-0.036066	-0.038874	-0.042077	-0.021391	-0.001373	-0.032067	-0.034854	-0.034627
Length	-0.036066	1.000000	0.986812	0.827554	0.925261	0.897914	0.903018	0.897706	0.556720
Diameter	-0.038874	0.986812	1.000000	0.833684	0.925452	0.893162	0.899724	0.905330	0.574660
Height	-0.042077	0.827554	0.833684	1.000000	0.819221	0.774972	0.798319	0.817338	0.557467
Whole weight	-0.021391	0.925261	0.925452	0.819221	1.000000	0.969405	0.966375	0.955355	0.540390
Shucked weight	-0.001373	0.897914	0.893162	0.774972	0.969405	1.000000	0.931961	0.882617	0.420884
Viscera weight	-0.032067	0.903018	0.899724	0.798319	0.966375	0.931961	1.000000	0.907656	0.503819
Shell weight	-0.034854	0.897706	0.905330	0.817338	0.955355	0.882617	0.907656	1.000000	0.627574
age	-0.034627	0.556720	0.574660	0.557467	0.540390	0.420884	0.503819	0.627574	1.000000

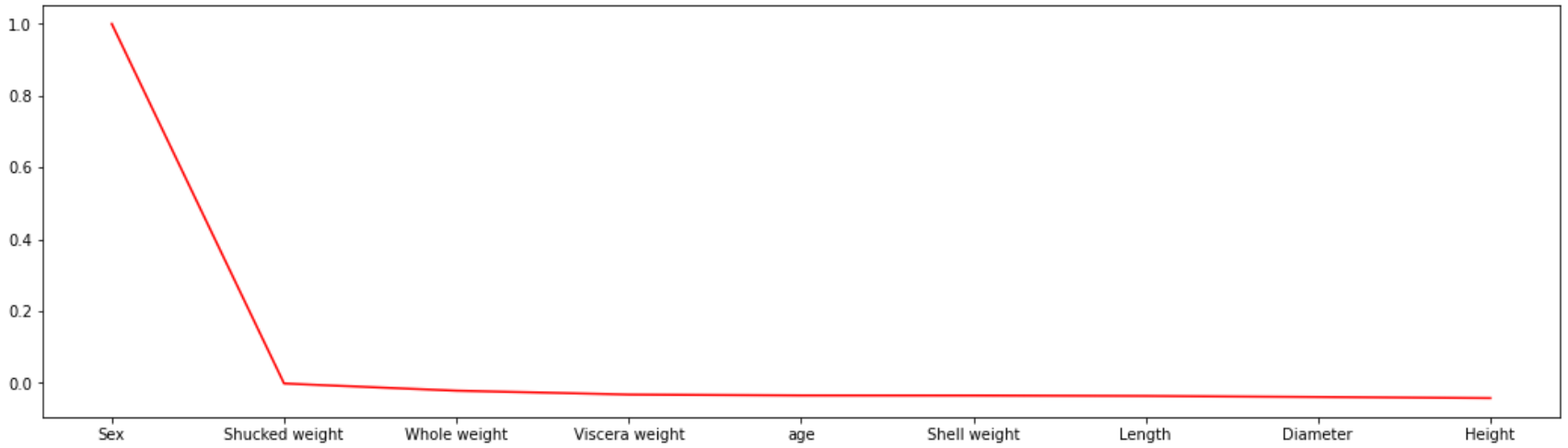
In [103]:
```
plt.figure(figsize=(20,5))
sns.heatmap(corr, annot=True, vmin=-1, cmap='crest')
```

Out[103]: <AxesSubplot:>



In [220]:
```
plt.figure(figsize=(18,5))
corr['Sex'].sort_values(ascending=False).plot(color='r')
```

Out[220]: <AxesSubplot:>



In [202]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

for i in df:
    if df[i].dtype=='object':
        df[i]=le.fit_transform(df[i])
```

In [206]:
```
x=df.drop(columns=['Sex'],axis = 1)
y = df.Sex
```

In [207]:
```
from sklearn.preprocessing import scale
X_Scaled = pd.DataFrame(scale(x), columns=x.columns)
X_Scaled.head()
```

Out[207]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age
0	-0.574558	-0.432149	-1.064424	-0.641898	-0.607685	-0.726212	-0.638217	1.571544
1	-1.448986	-1.439929	-1.183978	-1.230277	-1.170910	-1.205221	-1.212987	-0.910013
2	0.050033	0.122130	-0.107991	-0.309469	-0.463500	-0.356690	-0.207139	-0.289624
3	-0.699476	-0.432149	-0.347099	-0.637819	-0.648238	-0.607600	-0.602294	0.020571
4	-1.615544	-1.540707	-1.423087	-1.272086	-1.215968	-1.287337	-1.320757	-0.910013

In [209]:
```
from sklearn.model_selection import train_test_split
X_Train, X_Test, Y_Train, Y_Test = train_test_split(X_Scaled, y, test_size=0.2,random_state=0)
```

In [210]:
```
print(X.shape, X_train.shape, X_test.shape)
```

```
(4177, 8) (3759, 8) (418, 8)
```

In [211]:
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=10,criterion='entropy')
```

In [212]:
```
model.fit(X_Train,Y_Train)
```

Out[212]:
```
RandomForestClassifier(criterion='entropy', n_estimators=10)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [213]:
```
y_predict = model.predict(X_Test)
```

In [214]:
```
y_predict_train = model.predict(X_Train)
```

In [216]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

In [217]:
```
print('Training accuracy: ',accuracy_score(Y_Train,y_predict_train))
print('Testing accuracy: ',accuracy_score(Y_Test,y_predict))
```

```
Training accuracy:  0.9829392397485782
Testing accuracy:  0.5179425837320574
```

In [218]:
```
pd.crosstab(Y_Test,y_predict)
```

Out[218]:

col_0	0	1	2
Sex			
0	117	29	103
1	36	211	44
2	139	52	105

In [219]:
```
print(classification_report(Y_Test,y_predict))
```

```
              precision    recall  f1-score   support

           0       0.40      0.47      0.43       249
           1       0.72      0.73      0.72       291
           2       0.42      0.35      0.38       296

    accuracy                           0.52       836
   macro avg       0.51      0.52      0.51       836
weighted avg       0.52      0.52      0.52       836
```

In [ ]:
```

```