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

In [ ]:

```
df = pd.read_csv('/content/abalone.csv')
```

In [ ]:

```
df.head()
```

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	0.150	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 [ ]:

```
df.describe()
```

Out[ ]:

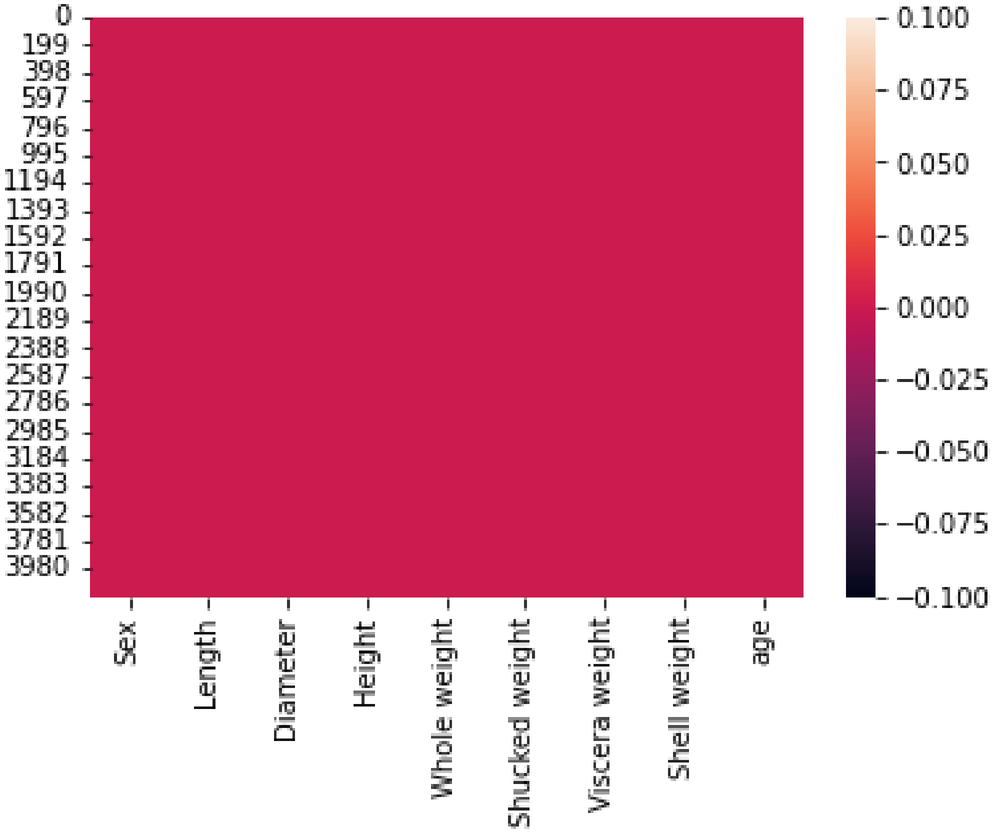
	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.120093	0.099240	0.041827	0.490389	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

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
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 [ ]:

```
sns.heatmap(df.isnull())
```

Out[ ]:



In [ ]:

```
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   age             4177 non-null   float64
dtypes: float64(8), object(1)
memory usage: 293.8+ KB
```

In [ ]:

```
numerical_features
```

Out[ ]:

```
Index(['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'age'],
      dtype='object')
```

In [ ]:

```
categorical_features
```

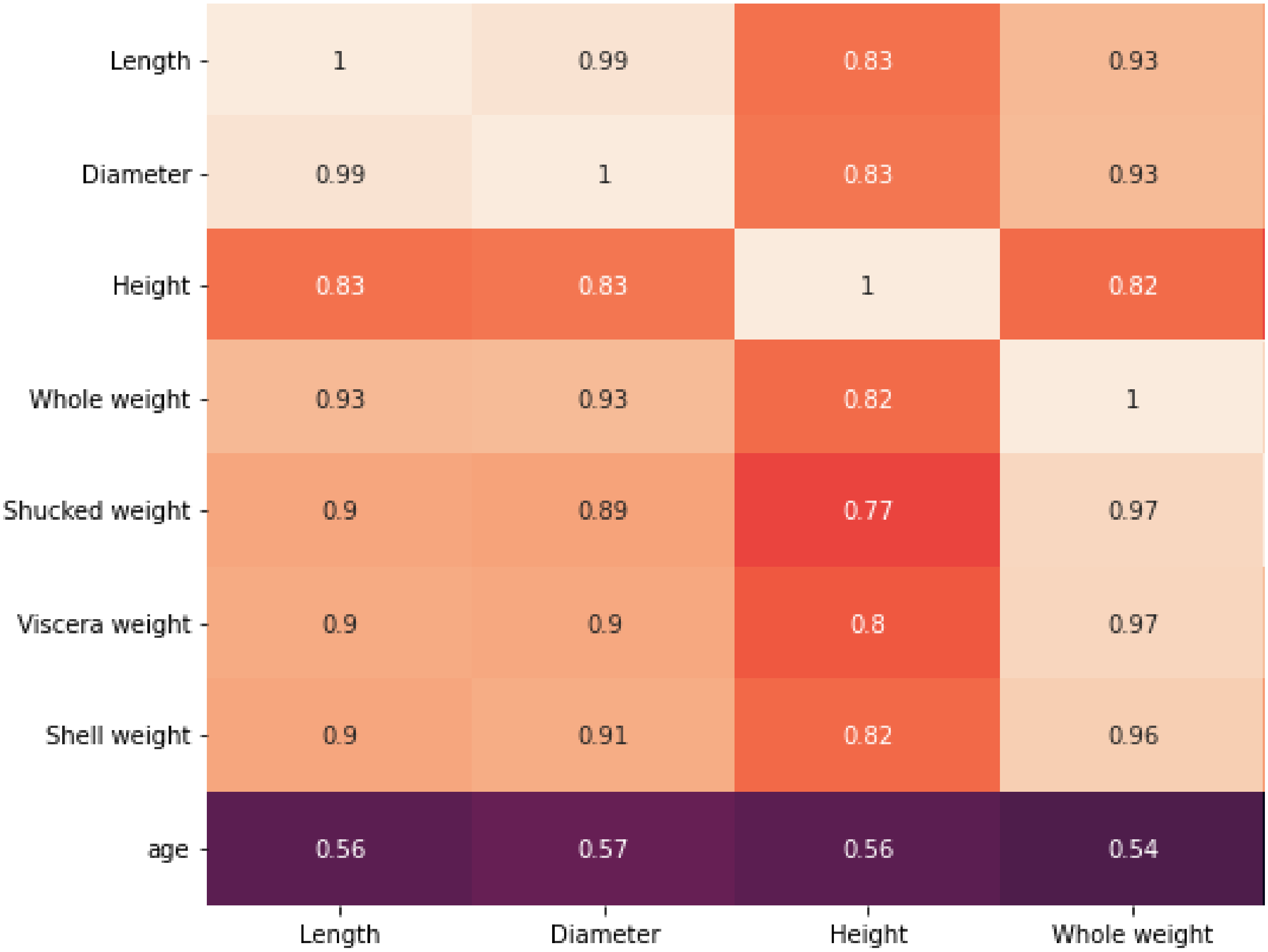
Out[ ]:

```
Index(['Sex'], dtype='object')
```

In [ ]:

```
plt.figure(figsize = (20,7))
sns.heatmap(df[numerical_features].corr(),annot = True)
```

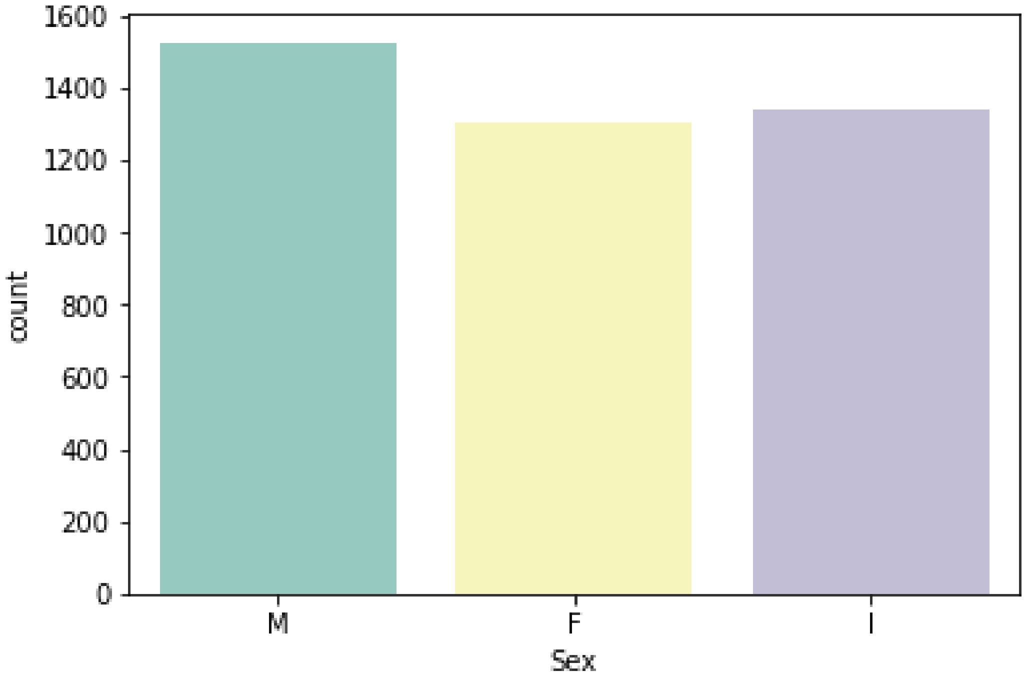
Out[ ]:



In [ ]:

```
sns.countplot(x = 'Sex', data = df, palette = 'Set3')
```

Out[ ]:

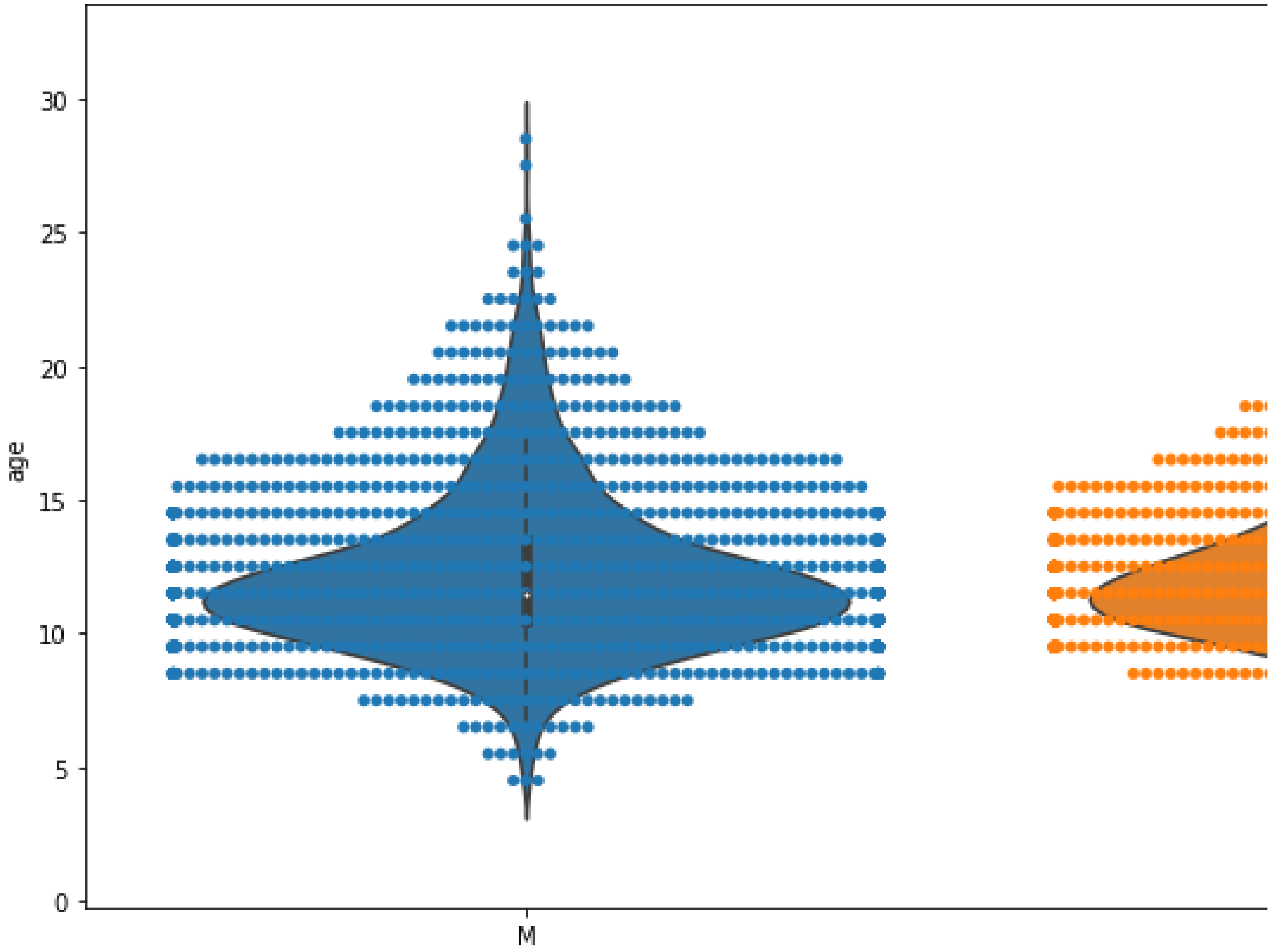


In [ ]:

```
plt.figure(figsize = (20,7))
sns.swarmplot(x = 'Sex', y = 'age', data = df, hue = 'Sex')
sns.violinplot(x = 'Sex', y = 'age',data = df)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarni
ng: 56.2% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarni
ng: 52.2% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarni
ng: 58.5% of the points cannot be placed; you may want to decrease the size o
f the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[ ]:

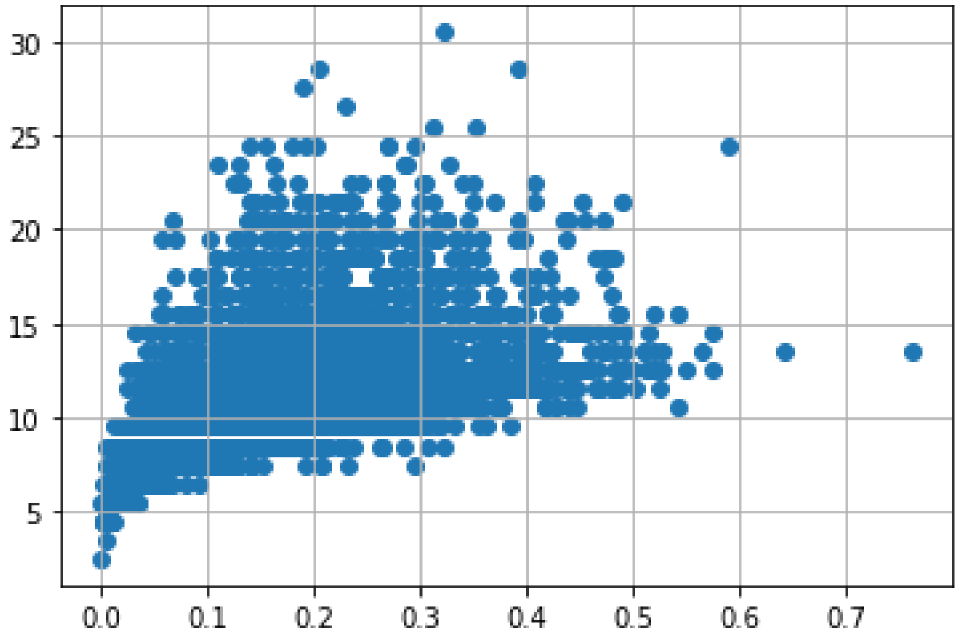


In [ ]:

```
df = pd.get_dummies(df)
dummy_df = df
```

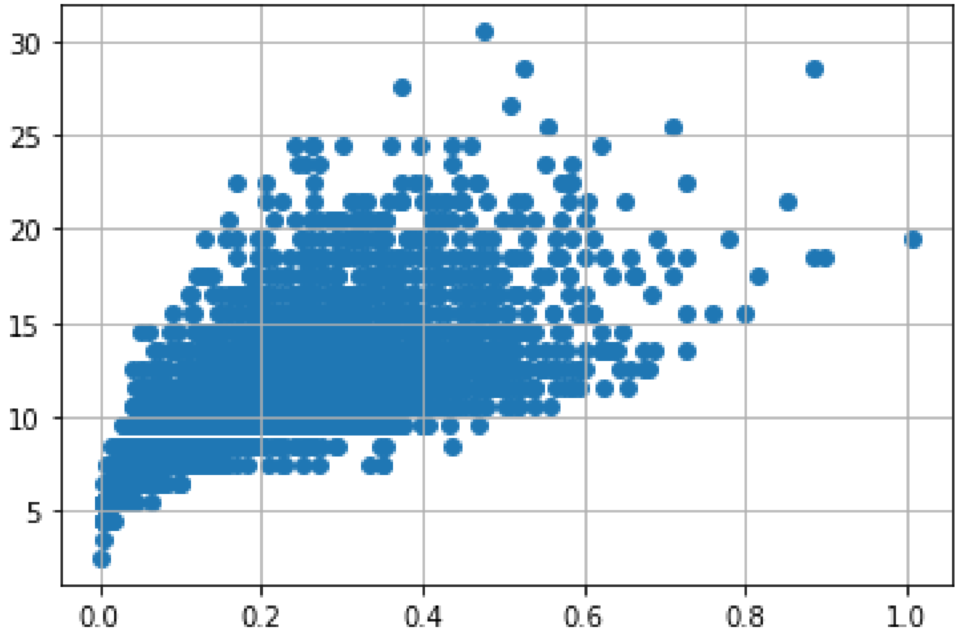
In [ ]:

```
#viscera weight
var = 'Viscera weight'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



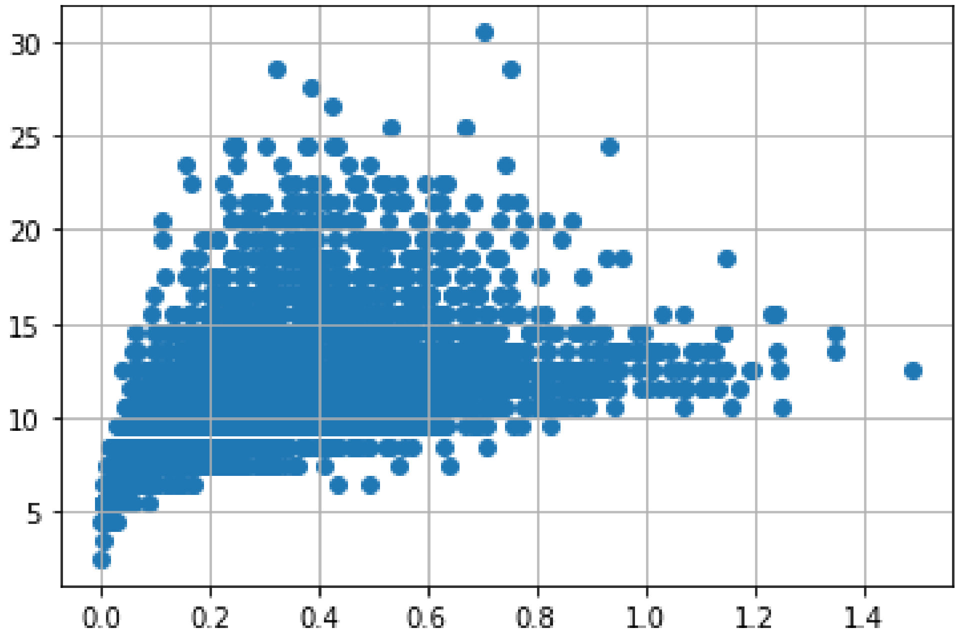
In [ ]:

```
#shell weight
var = 'Shell weight'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



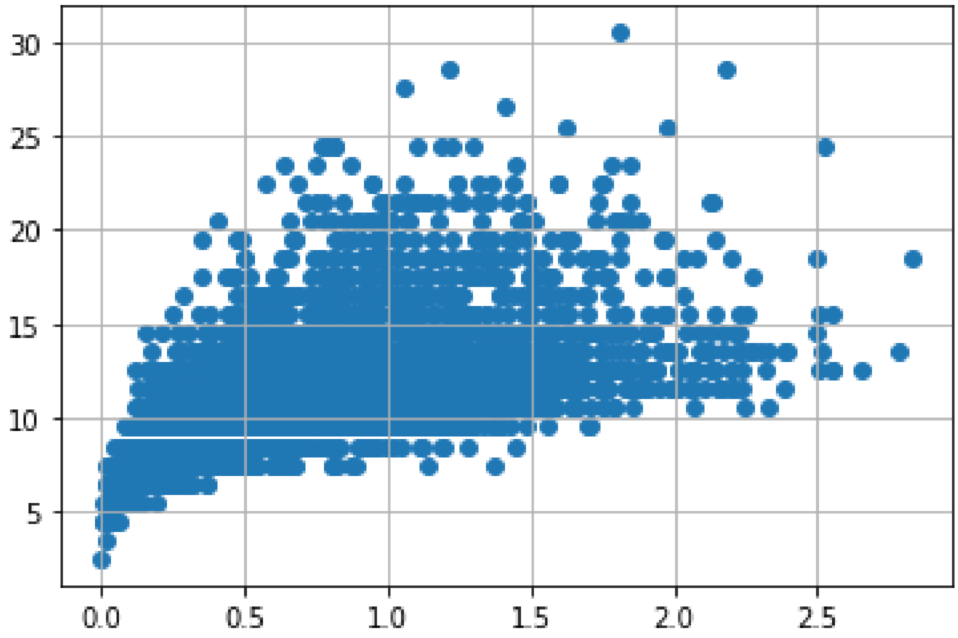
In [ ]:

```
#shucked weight
var = 'Shucked weight'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



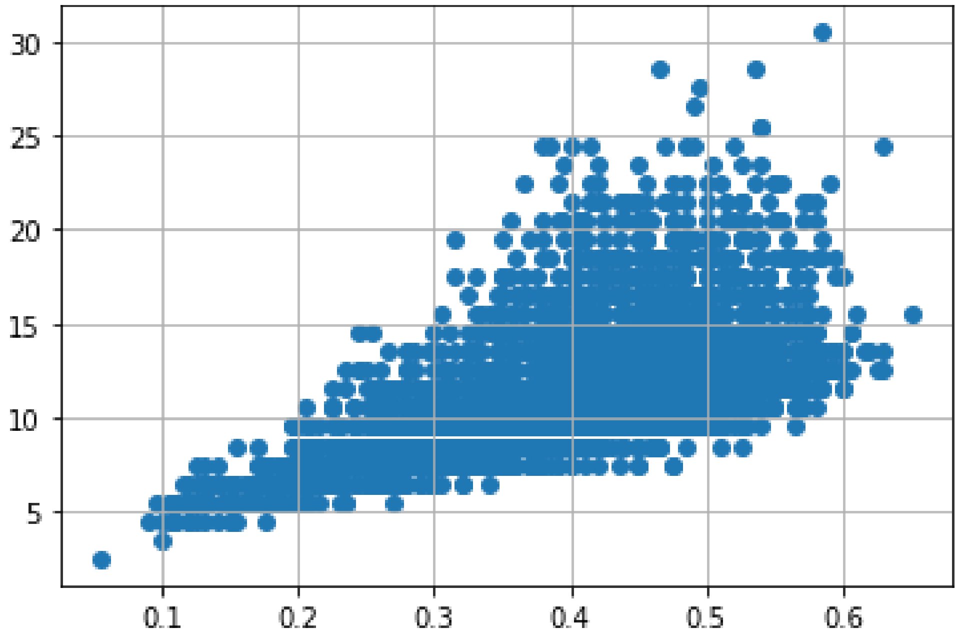
In [ ]:

```
#whole weight
var = 'Whole weight'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



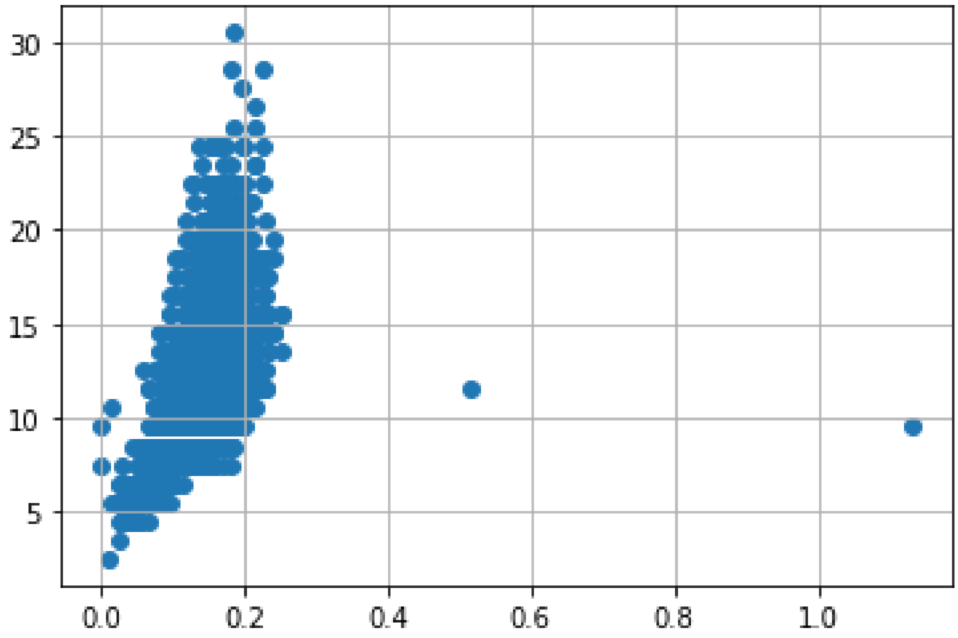
In [ ]:

```
#diameter
var = 'Diameter'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



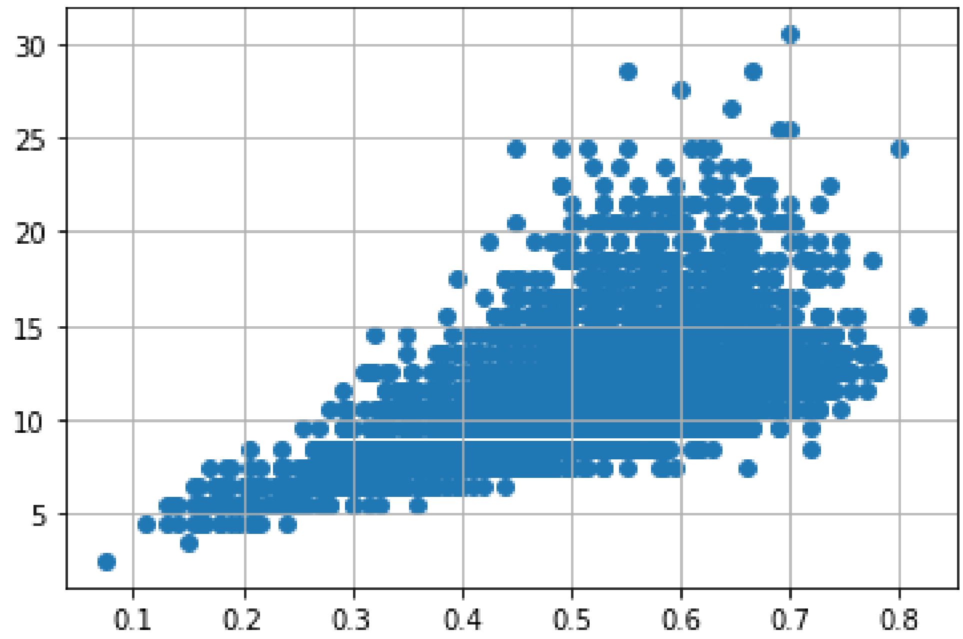
In [ ]:

```
#height
var = 'Height'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



In [ ]:

```
#length
var = 'Length'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
```



In [ ]:

```
#independent scale
```

In [ ]:

```
x=df.iloc[:, 0:1]
x
```

Out[ ]:

	Length
0	0.455
1	0.350
2	0.530
3	0.440
4	0.330
...	...
4172	0.565
4173	0.590

	Length
4174	0.600
4175	0.625
4176	0.710

4177 rows × 1 columns

In [ ]:

```
#dependent variable
```

In [ ]:

```
y=df.iloc[:,1:]
y
```

Out[ ]:

	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
0	0.365	0.095	0.5140	0.2245	0.1010	0.1500	16.5	0	0	1
1	0.265	0.090	0.2255	0.0995	0.0485	0.0700	8.5	0	0	1
2	0.420	0.135	0.6770	0.2565	0.1415	0.2100	10.5	1	0	0
3	0.365	0.125	0.5160	0.2155	0.1140	0.1550	11.5	0	0	1
4	0.255	0.080	0.2050	0.0895	0.0395	0.0550	8.5	0	1	0
...	...	...	...	...	...	...	...	...	...	...
4172	0.450	0.165	0.8870	0.3700	0.2390	0.2490	12.5	1	0	0
4173	0.440	0.135	0.9660	0.4390	0.2145	0.2605	11.5	0	0	1
4174	0.475	0.205	1.1760	0.5255	0.2875	0.3080	10.5	0	0	1
4175	0.485	0.150	1.0945	0.5310	0.2610	0.2960	11.5	1	0	0

	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
4176	0.555	0.195	1.9485	0.9455	0.3765	0.4950	13.5	0	0	1

4177 rows × 10 columns

In [ ]:

```
#Train the Model, Test the Model, split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state
=0)
```

In [ ]:

```
x_train.shape
```

Out[ ]:

```
(3341, 1)
```

In [ ]:

```
x_test.shape
```

Out[ ]:

```
(836, 1)
```

In [ ]:

```
x_test
```

Out[ ]:

	Length
668	0.550
1580	0.500
3784	0.620
463	0.220
2615	0.645
...	...
575	0.610
3231	0.410

	Length
1084	0.445
290	0.540
2713	0.250

836 rows × 1 columns

In [ ]:

```
y_train.shape
```

Out[ ]:

```
(3341, 10)
```

In [ ]:

```
y_test.shape
```

Out[ ]:

```
(836, 10)
```

model of building

In [ ]:

```
# Build the Model
from sklearn.linear_model import linearRegression
slr=LinearRegression()
slr.fit(x_train,y_train)
```

In [45]:

```
x_test
```

Out[45]:

	Length
668	0.550
1580	0.500
3784	0.620
463	0.220
2615	0.645
...	...

	Length
575	0.610
3231	0.410
1084	0.445
290	0.540
2713	0.250

836 rows × 1 columns

In [46]:

```
y_test
```

Out[46]:

	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
668	0.425	0.155	0.9175	0.2775	0.2430	0.3350	14.5	0	0	1
1580	0.400	0.120	0.6160	0.2610	0.1430	0.1935	9.5	0	1	0
3784	0.480	0.155	1.2555	0.5270	0.3740	0.3175	12.5	0	0	1
463	0.165	0.055	0.0545	0.0215	0.0120	0.0200	6.5	0	1	0
2615	0.500	0.175	1.5105	0.6735	0.3755	0.3775	13.5	0	0	1
...	...	...	...	...	...	...	...	...	...	...
575	0.475	0.140	1.1330	0.5275	0.2355	0.3500	12.5	1	0	0
3231	0.325	0.120	0.3745	0.1580	0.0810	0.1250	13.5	0	0	1
1084	0.345	0.105	0.4090	0.1675	0.1015	0.1170	8.5	0	1	0

	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
290	0.435	0.180	0.9960	0.3835	0.2260	0.3250	18.5	0	0	1
2713	0.175	0.060	0.0635	0.0275	0.0080	0.0200	5.5	0	1	0

836 rows × 10 columns

In [49]:

```
#descriptive statistics
df.describe()
```

Out[49]:

	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
count	4177.000000	4177.000000	4177.000000	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	11.433684	0.312904	0.321283	0.365813
std	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169	0.463731	0.467025	0.481715
min	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	2.500000	0.000000	0.000000	0.000000
25%	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	9.500000	0.000000	0.000000	0.000000
50%	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	10.500000	0.000000	0.000000	0.000000
75%	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	12.500000	1.000000	1.000000	1.000000
max	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	30.500000	1.000000	1.000000	1.000000

In [52]:

```
#multple
```

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

Out[52]:

	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 [53]:

```
df.tail()
```

Out[53]:

	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