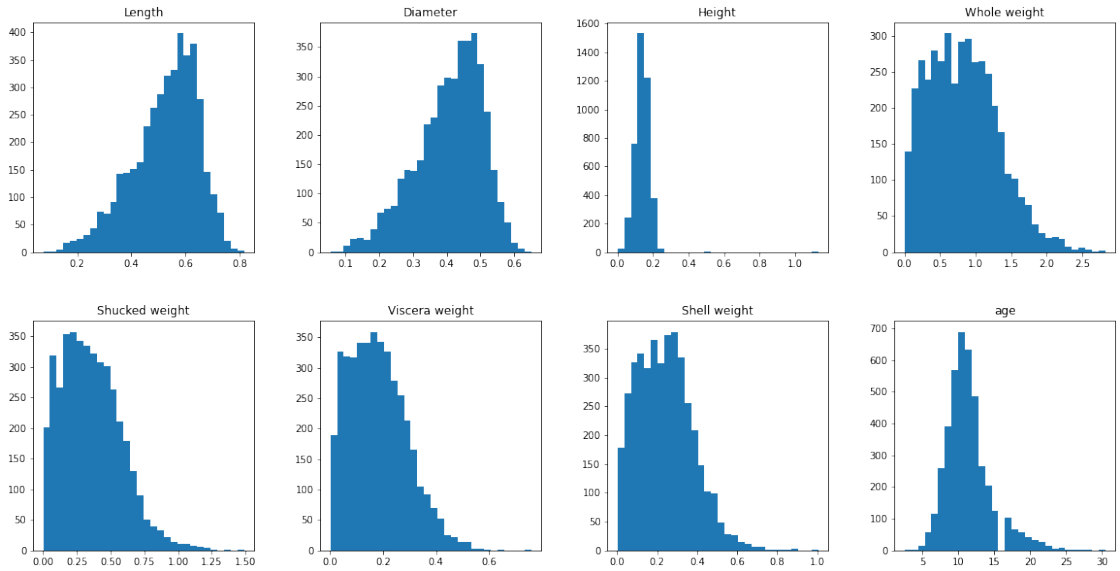
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

df=pd.read_csv("abalone.csv")

df
```

```
     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     15
1             0.0485        0.0700      7
2             0.1415        0.2100      9
3             0.1140        0.1550     10
4             0.0395        0.0550      7
...              ...           ...    ...
4172          0.2390        0.2490     11
4173          0.2145        0.2605     10
4174          0.2875        0.3080      9
4175          0.2610        0.2960     10
4176          0.3765        0.4950     12

[4177 rows x 9 columns]
```

```
df['age'] = df['Rings']+1.5
df = df.drop('Rings', axis = 1)

df.hist(figsize=(20,10), grid=False, layout=(2, 4), bins = 30)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9eb4af90>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9defc2d0>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9deb68d0>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9de6bed0>],
```

```
       [<matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9de30510>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9dde6b10>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9ddaa1d0>,
        <matplotlib.axes._subplots.AxesSubplot object at
0x7f3a9dd5d710>]],
      dtype=object)
```

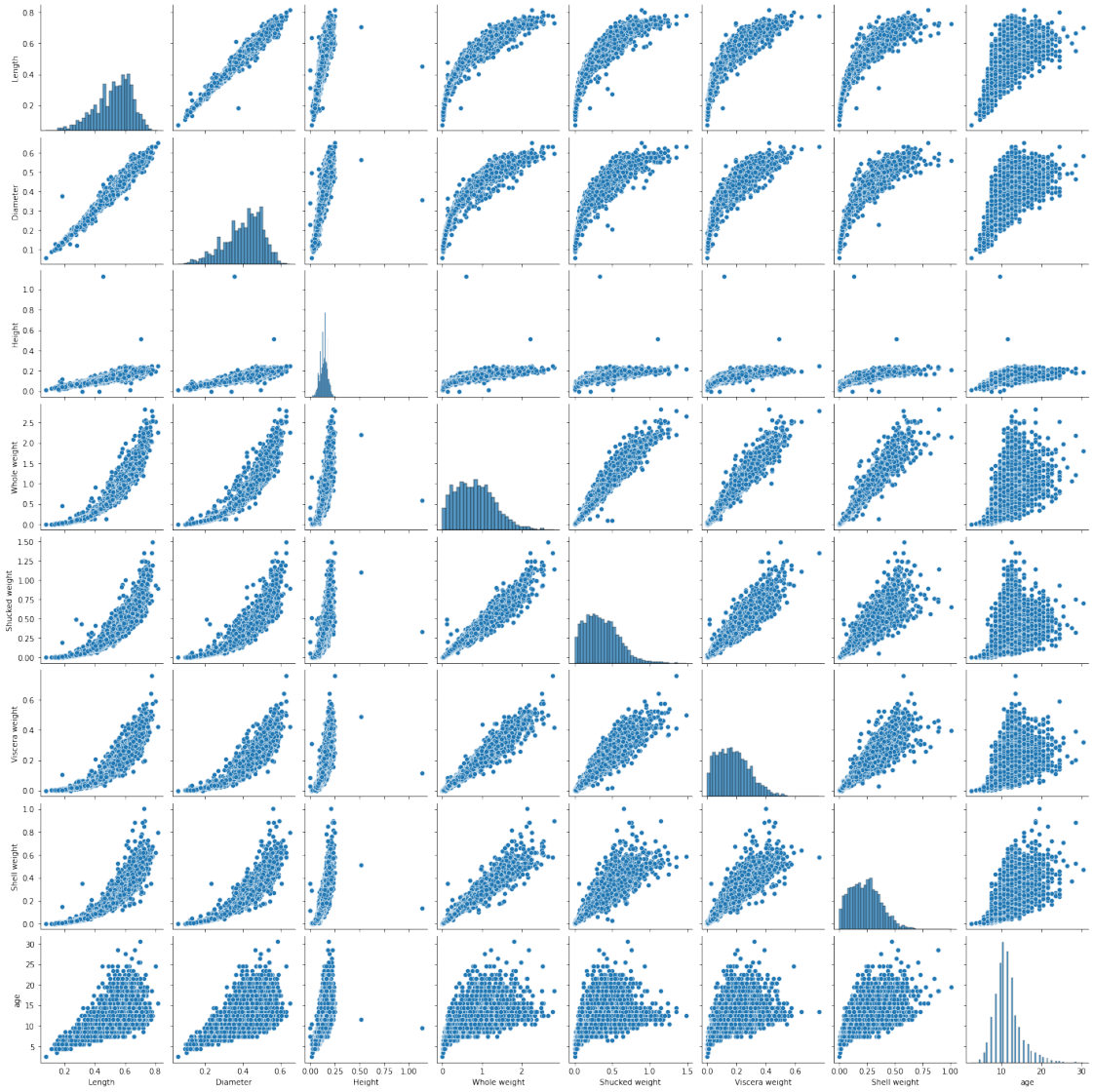



```
df.groupby('Sex')[['Length', 'Diameter', 'Height', 'Whole weight',
'Shucked weight',
       'Viscera weight', 'Shell weight',
'age']].mean().sort_values('age')
```

```
       Length  Diameter    Height  Whole weight  Shucked weight  \
Sex
I    0.427746  0.326494  0.107996      0.431363        0.191035
M    0.561391  0.439287  0.151381      0.991459        0.432946
F    0.579093  0.454732  0.158011      1.046532        0.446188

     Viscera weight  Shell weight        age
Sex
I          0.092010      0.128182   9.390462
M          0.215545      0.281969  12.205497
F          0.230689      0.302010  12.629304
```

```
numerical_features = df.select_dtypes(include = [np.number]).columns
sns.pairplot(df[numerical_features])
```

```
<seaborn.axisgrid.PairGrid at 0x7f3a9d89c490>
```




```
df.describe()
```

	Length	Diameter	Height	Whole weight	Shucked weight
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.523992	0.407881	0.139516	0.828742	0.359367
std	0.120093	0.099240	0.041827	0.490389	0.221963
min	0.075000	0.055000	0.000000	0.002000	0.001000
25%	0.450000	0.350000	0.115000	0.441500	0.186000
50%	0.545000	0.425000	0.140000	0.799500	0.336000
75%	0.615000	0.480000	0.165000	1.153000	

```
0.502000
max       0.815000     0.650000     1.130000      2.825500
1.488000
```

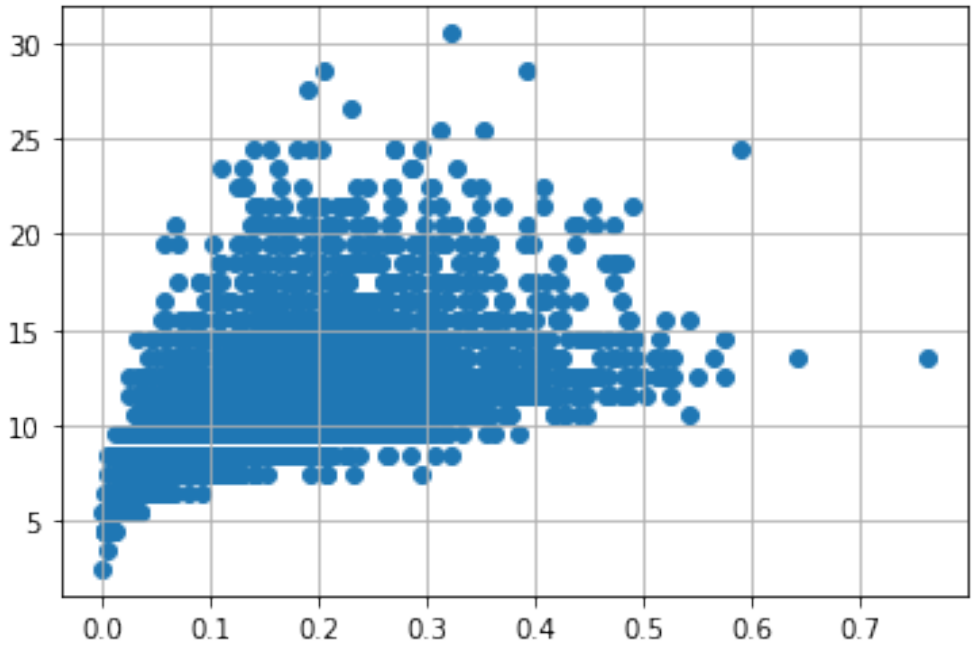
```
       Viscera weight  Shell weight          age
count     4177.000000   4177.000000  4177.000000
mean         0.180594      0.238831    11.433684
std          0.109614      0.139203     3.224169
min          0.000500      0.001500     2.500000
25%          0.093500      0.130000     9.500000
50%          0.171000      0.234000    10.500000
75%          0.253000      0.329000    12.500000
max          0.760000      1.005000    30.500000
```

```
df.isnull().sum()
```

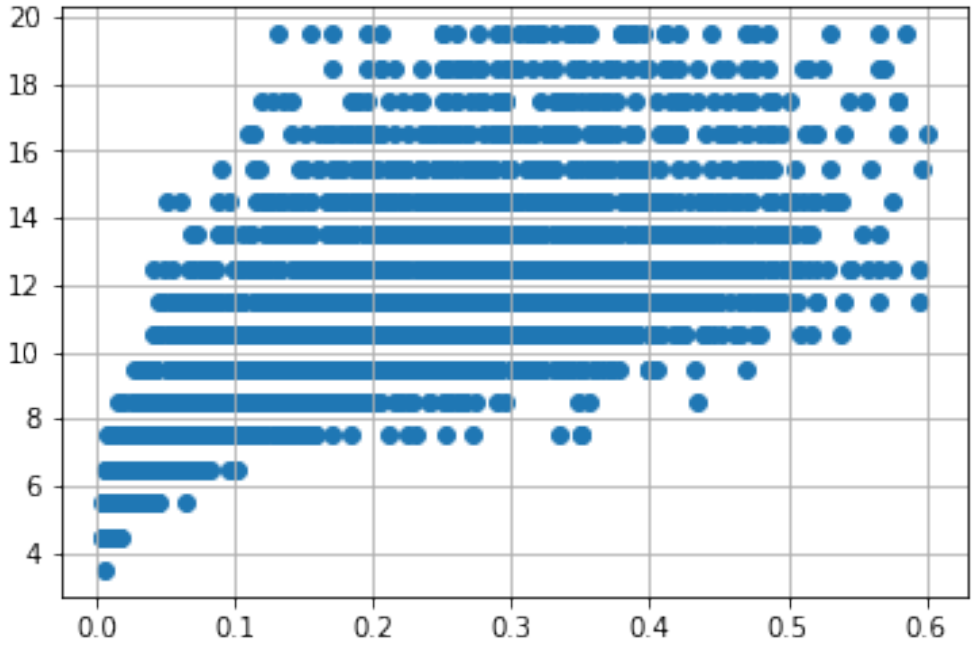
```
Sex               0
Length            0
Diameter          0
Height            0
Whole weight      0
Shucked weight    0
Viscera weight    0
Shell weight      0
age               0
dtype: int64
```

```
df = pd.get_dummies(df)
dummy_data = df.copy()
```

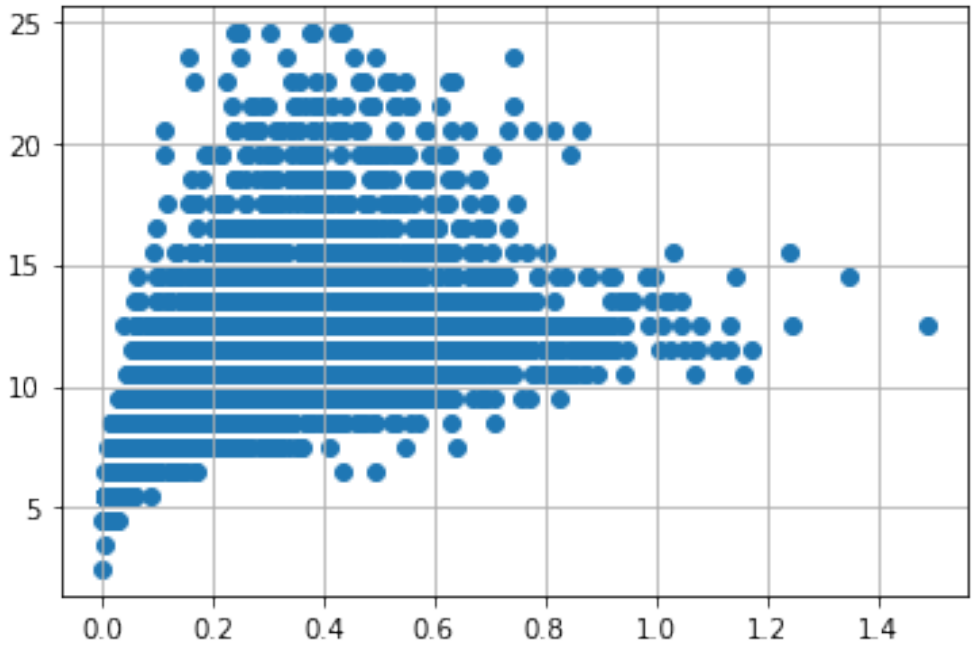
```
var = 'Viscera weight'
plt.scatter(x = df[var], y = df['age'],)
plt.grid(True)
```



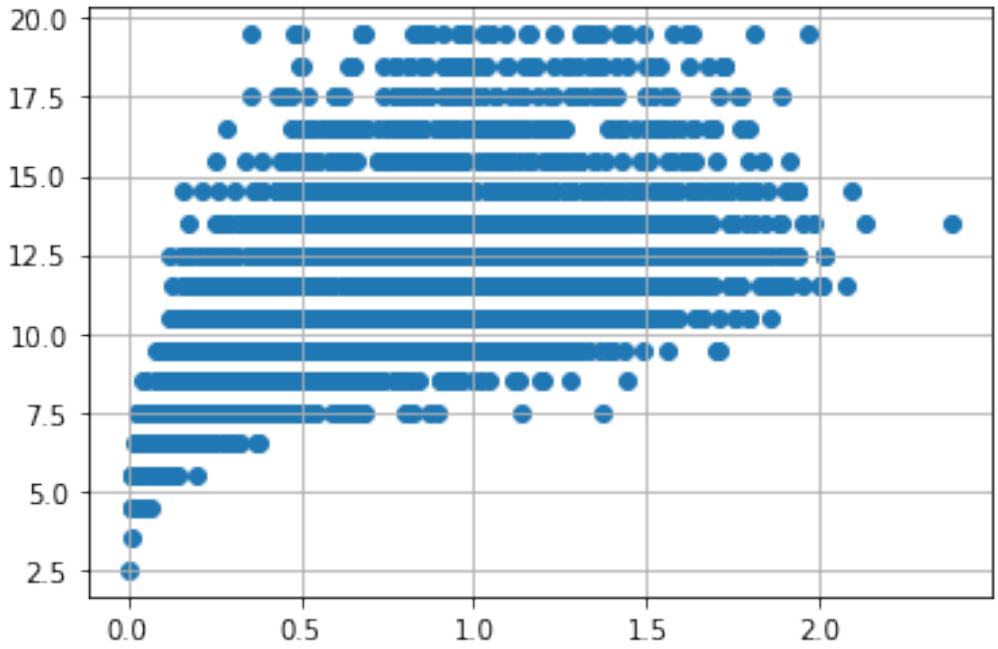

```
df.drop(df[(df['Viscera weight']> 0.5) & (df['age'] < 20)].index,
inplace=True)
df.drop(df[(df['Viscera weight']<0.5) & (df['age'] > 25)].index,
inplace=True)
var = 'Shell weight'
plt.scatter(x = df[var], y = df['age'],)
plt.grid(True)
#Outliers removal
df.drop(df[(df['Shell weight']> 0.6) & (df['age'] < 25)].index,
inplace=True)
df.drop(df[(df['Shell weight']<0.8) & (df['age'] > 25)].index,
inplace=True)
```



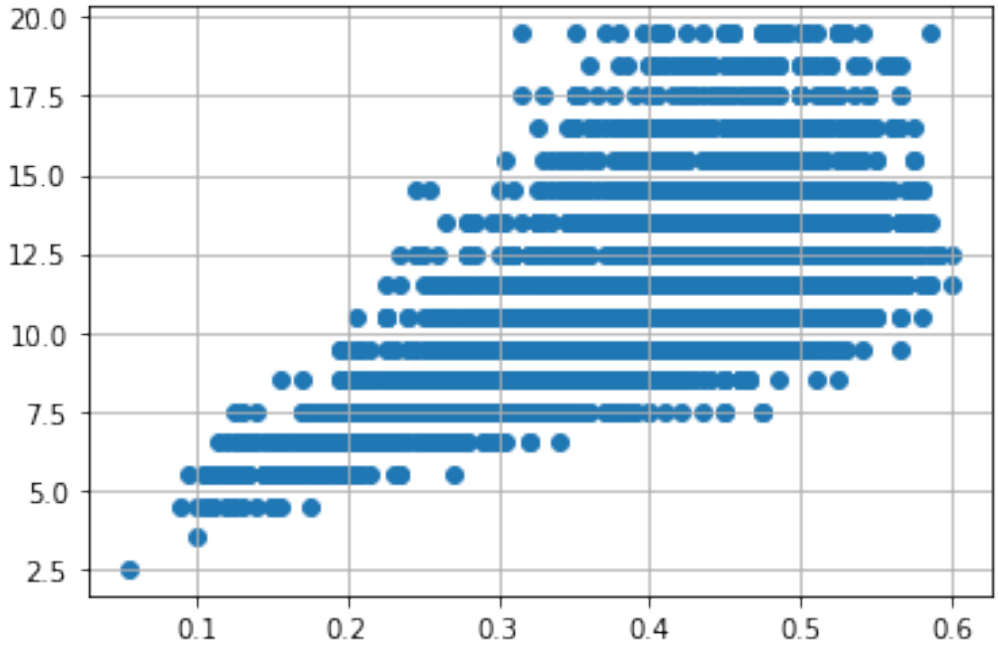

```
var = 'Shucked weight'
plt.scatter(x = df[var], y = df['age'],)
plt.grid(True)

#Outlier removal
df.drop(df[(df['Shucked weight']>= 1) & (df['age'] < 20)].index,
inplace=True)
df.drop(df[(df['Shucked weight']<1) & (df['age'] > 20)].index,
inplace=True)
```



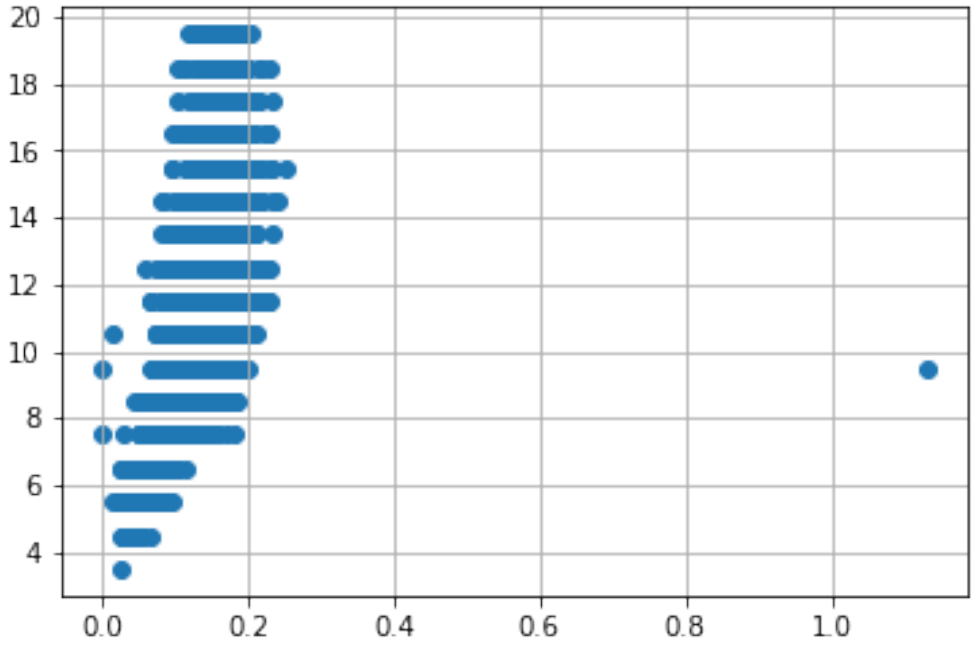
```
var = 'Whole weight'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)

df.drop(df[(df['Whole weight'] >= 2.5) &
           (df['age'] < 25)].index, inplace = True)
df.drop(df[(df['Whole weight']<2.5) & (
df['age'] > 25)].index, inplace = True)
```




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

df.drop(df[(df['Diameter'] <0.1) &
           (df['age'] < 5)].index, inplace = True)
df.drop(df[(df['Diameter']<0.6) & (
df['age'] > 25)].index, inplace = True)
df.drop(df[(df['Diameter']>=0.6) & (
df['age'] < 25)].index, inplace = True)
```

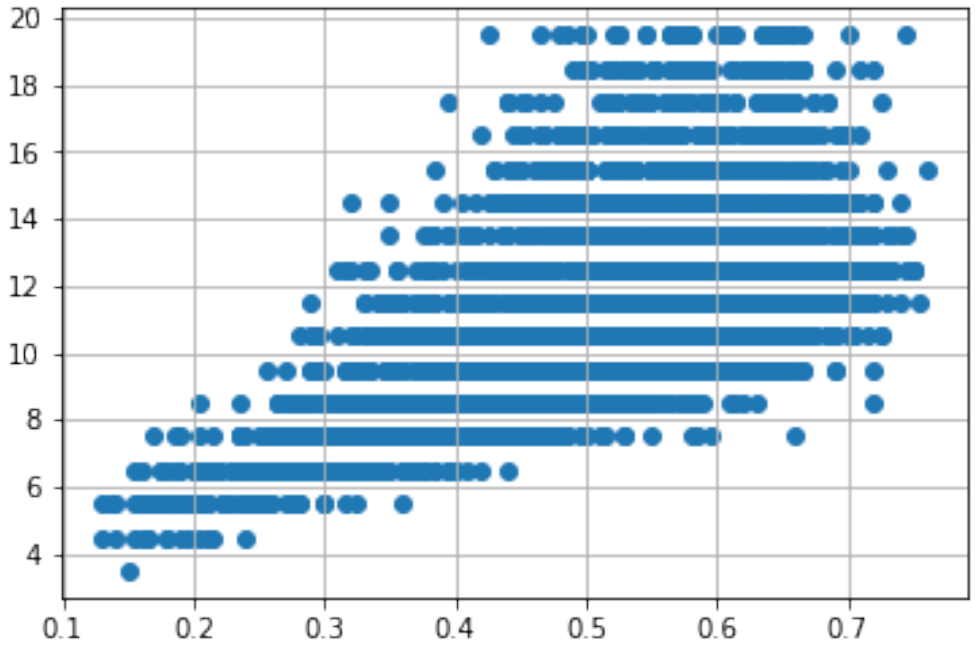



```
var = 'Height'
plt.scatter(x = df[var], y = df['age'])
plt.grid(True)
df.drop(df[(df['Height'] > 0.4) &
           (df['age'] < 15)].index, inplace = True)
df.drop(df[(df['Height']<0.4) & (
df['age'] > 25)].index, inplace = True)
```




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

```
plt.grid(True)

df.drop(df[(df['Length'] <0.1) &
           (df['age'] < 5)].index, inplace = True)
df.drop(df[(df['Length']<0.8) & (
df['age'] > 25)].index, inplace = True)
df.drop(df[(df['Length']>=0.8) & (
df['age'] < 25)].index, inplace = True)
```




```
numerical_features = df.select_dtypes(include = [np.number]).columns
categorical_features = df.select_dtypes(include = [np.object]).columns
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2:
DeprecationWarning: `np.object` is a deprecated alias for the builtin
`object`. To silence this warning, use `object` by itself. Doing this
will not modify any behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
```

```
numerical_features
```

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

```
categorical_features
```

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

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
print(df.Length.value_counts())
```

```
0.575    93
0.625    91
0.580    89
0.550    89
0.620    83
         ..
0.220     2
0.150     1
0.755     1
0.135     1
0.760     1
Name: Length, Length: 126, dtype: int64
```

```
x=df.iloc[:,:5]
x
```

	Length	Diameter	Height	Whole weight	Shucked weight
0	0.455	0.365	0.095	0.5140	0.2245
1	0.350	0.265	0.090	0.2255	0.0995
2	0.530	0.420	0.135	0.6770	0.2565
3	0.440	0.365	0.125	0.5160	0.2155
4	0.330	0.255	0.080	0.2050	0.0895
...	...	...	...	...	...
4172	0.565	0.450	0.165	0.8870	0.3700
4173	0.590	0.440	0.135	0.9660	0.4390
4174	0.600	0.475	0.205	1.1760	0.5255
4175	0.625	0.485	0.150	1.0945	0.5310
4176	0.710	0.555	0.195	1.9485	0.9455

[3995 rows x 5 columns]

```
y=df.iloc[:,5:]
y
```

	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
0	0.1010	0.1500	16.5	0	0	1
1	0.0485	0.0700	8.5	0	0	1
2	0.1415	0.2100	10.5	1	0	0
3	0.1140	0.1550	11.5	0	0	1
4	0.0395	0.0550	8.5	0	1	0
...	...	...	...	...	...	...
4172	0.2390	0.2490	12.5	1	0	0
4173	0.2145	0.2605	11.5	0	0	1
4174	0.2875	0.3080	10.5	0	0	1
4175	0.2610	0.2960	11.5	1	0	0
4176	0.3765	0.4950	13.5	0	0	1

```
[3995 rows x 6 columns]
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)
```

```
LinearRegression()
```

```
x_test.head()
```

	Length	Diameter	Height	Whole weight	Shucked weight
3330	0.385	0.305	0.125	0.3140	0.1460
508	0.560	0.435	0.180	0.8890	0.3600
3988	0.665	0.515	0.165	1.3855	0.6210
2435	0.465	0.380	0.135	0.5790	0.2080
763	0.640	0.510	0.190	1.6130	0.6215

```
y_test.head()
```

	Viscera weight	Shell weight	age	Sex_F	Sex_I	Sex_M
3330	0.0555	0.0800	11.5	1	0	0
508	0.2040	0.2500	12.5	0	0	1
3988	0.3020	0.3445	9.5	0	0	1
2435	0.1095	0.2200	15.5	1	0	0
763	0.3610	0.4700	15.5	0	0	1

```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
x_train=ss.fit_transform(x_train)
```

```
mlrpred=mlr.predict(x_test[0:9])
mlrpred
```

```
from sklearn.metrics import r2_score
r2_score(mlr.predict(x_test),y_test)
```

```
-3.3893282910298814
```