```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

Loading Dataset

```
data = pd.read_csv('abalone.csv')
```

```
data
```

	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

```
data.head()
```

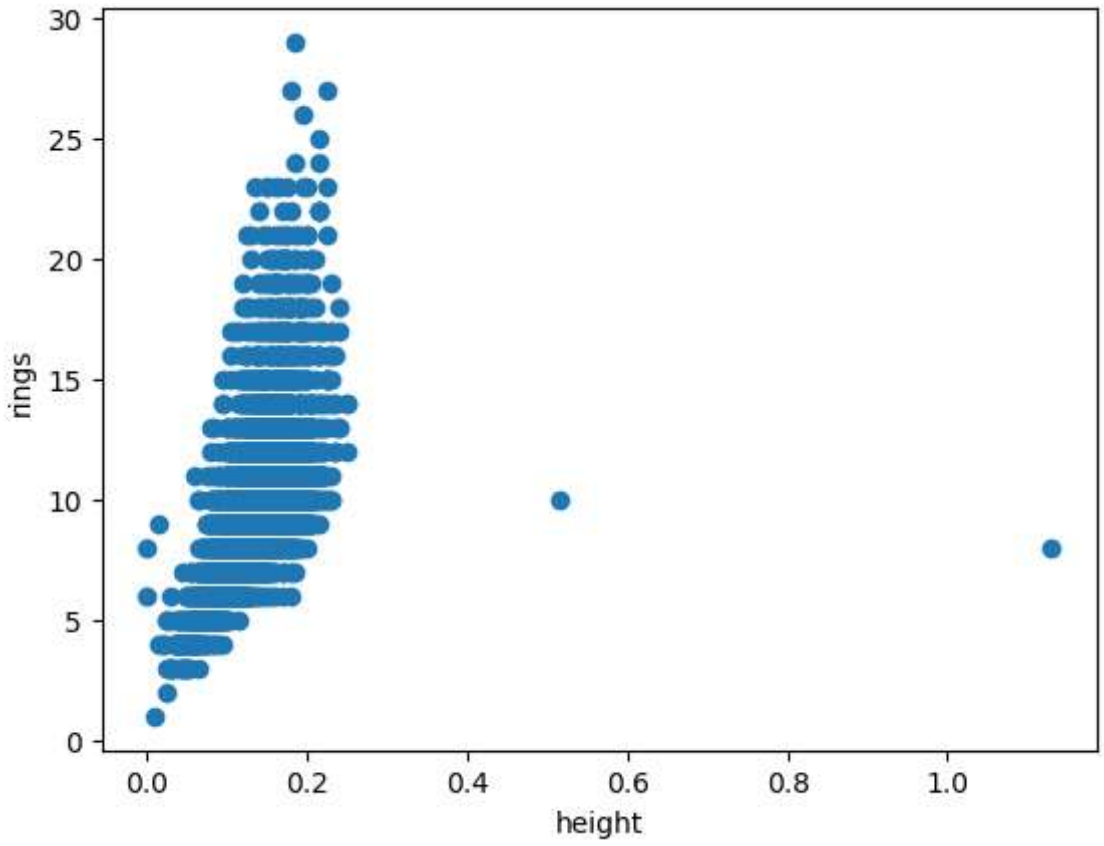
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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
```

```
plt.scatter(data.Height,data.Rings)
plt.xlabel('height')
plt.ylabel('rings')
```

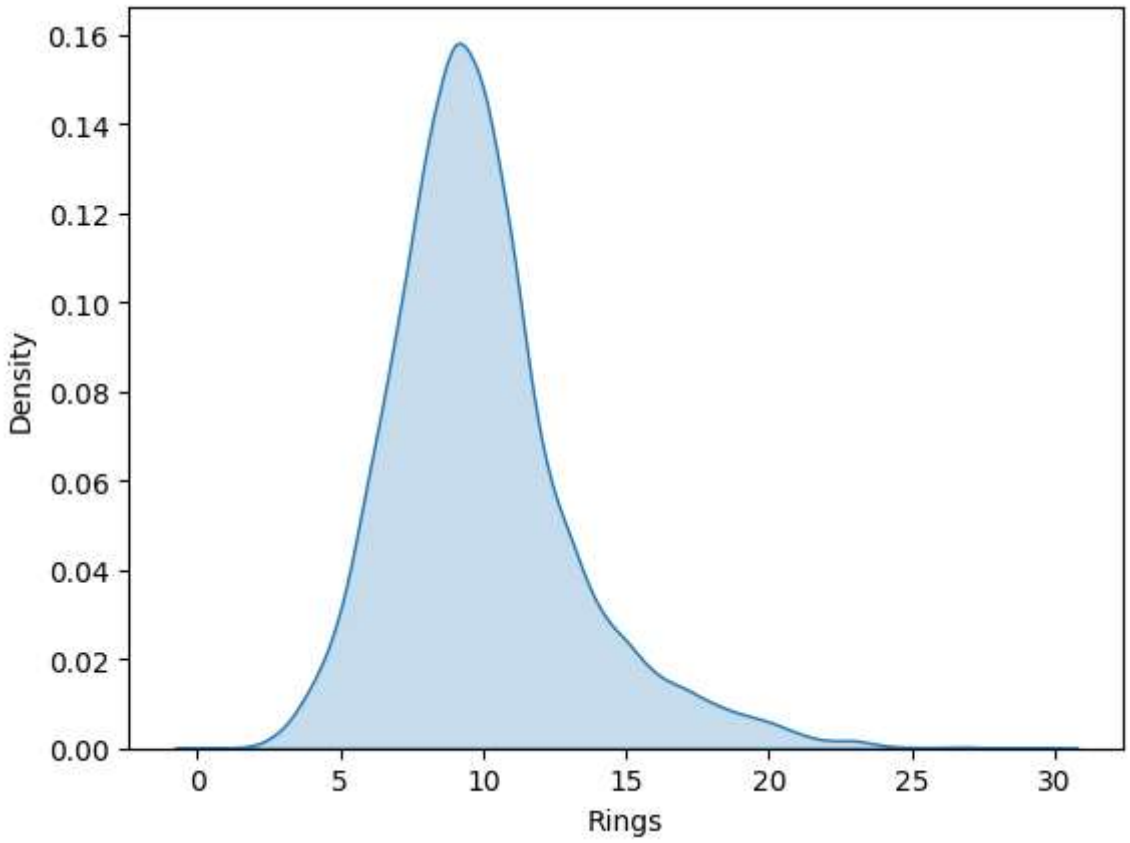
```
Text(0, 0.5, 'rings')
```



univarient analysis

```
sns.kdeplot(data['Rings'],·fill=True)
```

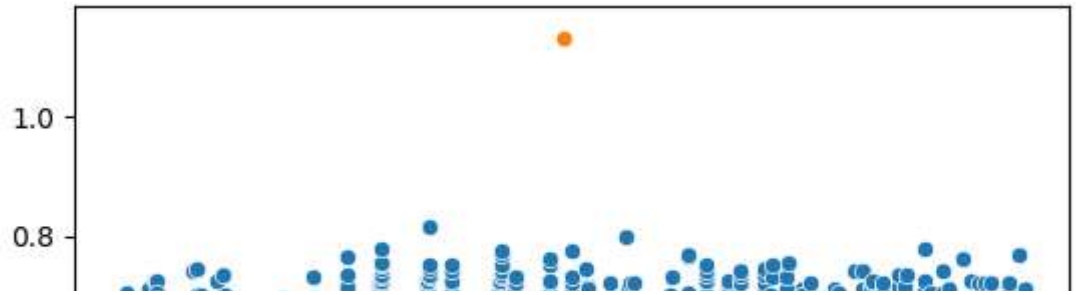
```
<AxesSubplot: xlabel='Rings', ylabel='Density'>
```



Bi-Varient analysis

```
sns.scatterplot(data['Length'])
sns.scatterplot(data["Height"])
```

```
<AxesSubplot: ylabel='Length'>
```



Multi-varient analysis

```
data.describe(include='all')
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	V
count	4177	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.
unique	3	NaN	NaN	NaN	NaN	NaN	
top	M	NaN	NaN	NaN	NaN	NaN	
freq	1528	NaN	NaN	NaN	NaN	NaN	
mean	NaN	0.523992	0.407881	0.139516	0.828742	0.359367	0.
std	NaN	0.120093	0.099240	0.041827	0.490389	0.221963	0.
min	NaN	0.075000	0.055000	0.000000	0.002000	0.001000	0.
25%	NaN	0.450000	0.350000	0.115000	0.441500	0.186000	0.
50%	NaN	0.545000	0.425000	0.140000	0.799500	0.336000	0.
75%	NaN	0.615000	0.480000	0.165000	1.153000	0.502000	0.

Descriptive stats

```
data.describe(include='all')
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight
count	2013.000000	2013.000000	2013.000000	2013.000000	2013.000000	2013.000000
mean	1.056135	0.522755	0.406647	0.138328	0.803753	0.350290

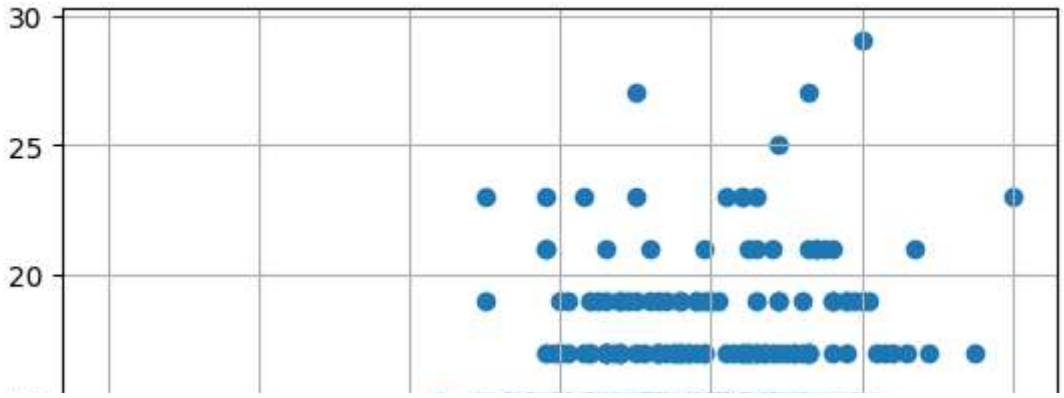
missing values

```
data.dropna()
```

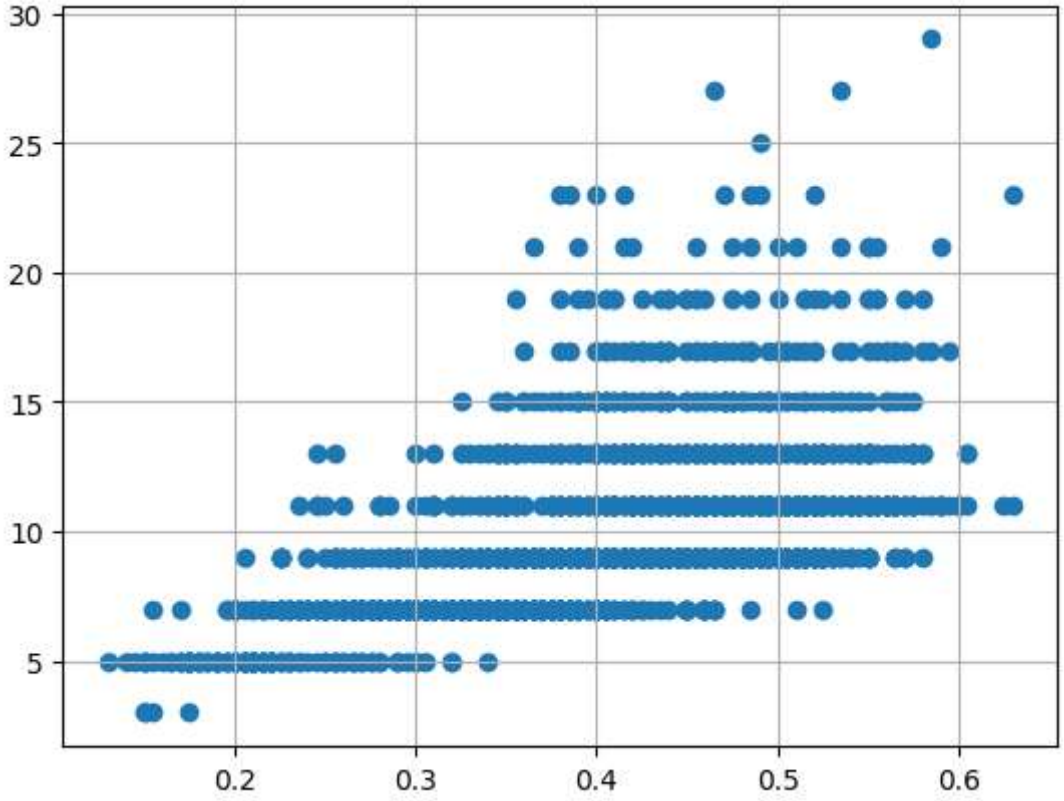
	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
4	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

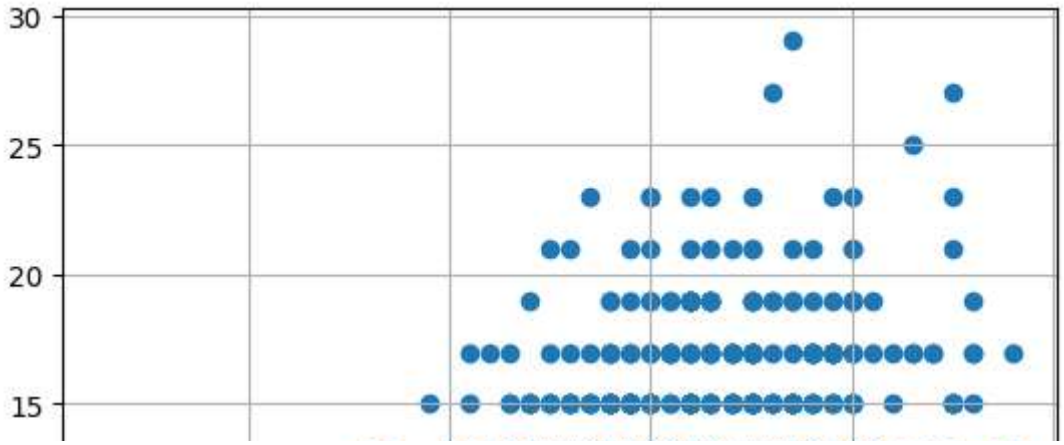
```
data = data[(data['Length'] <= 0.8 ) & (data['Length'] >= 0.2)& (data['Rings'])]


plt.scatter(x = data['Length'], y = data['Rings'])
plt.grid(True)
```




```
plt.scatter(x = data['Diameter'], y = data['Rings'])
plt.grid(True)
```

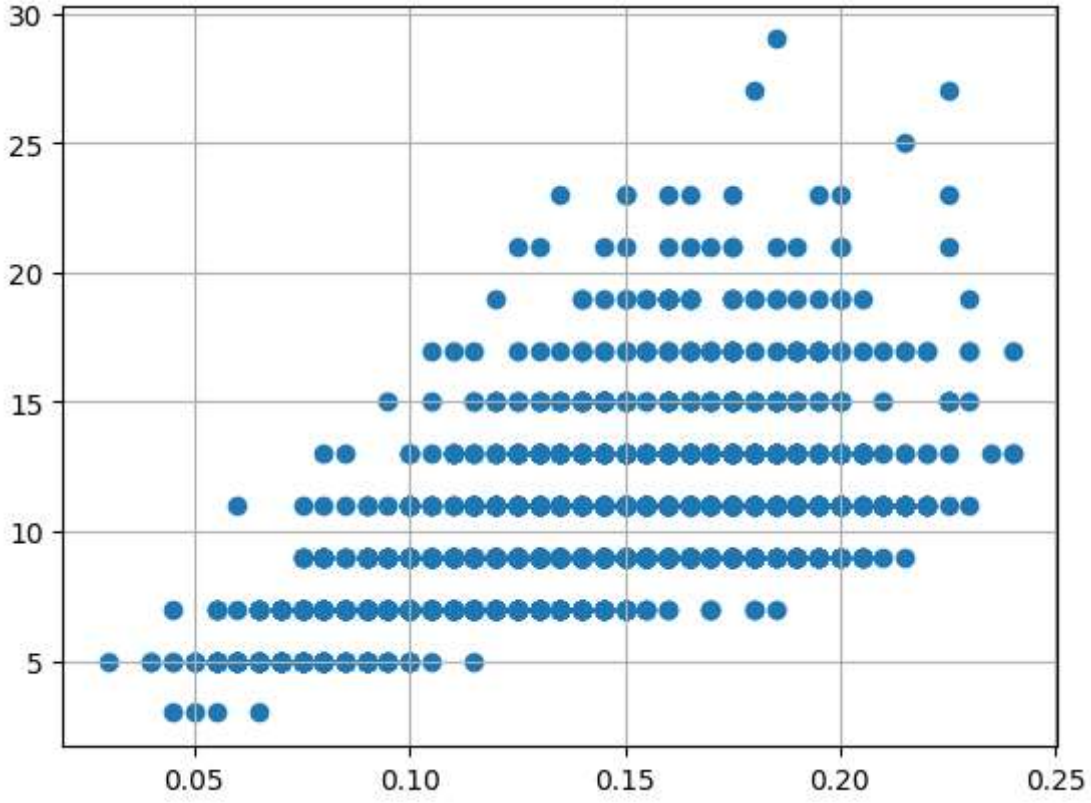



```
plt.scatter(x = data['Height'], y = data['Rings'])
plt.grid(True)
```

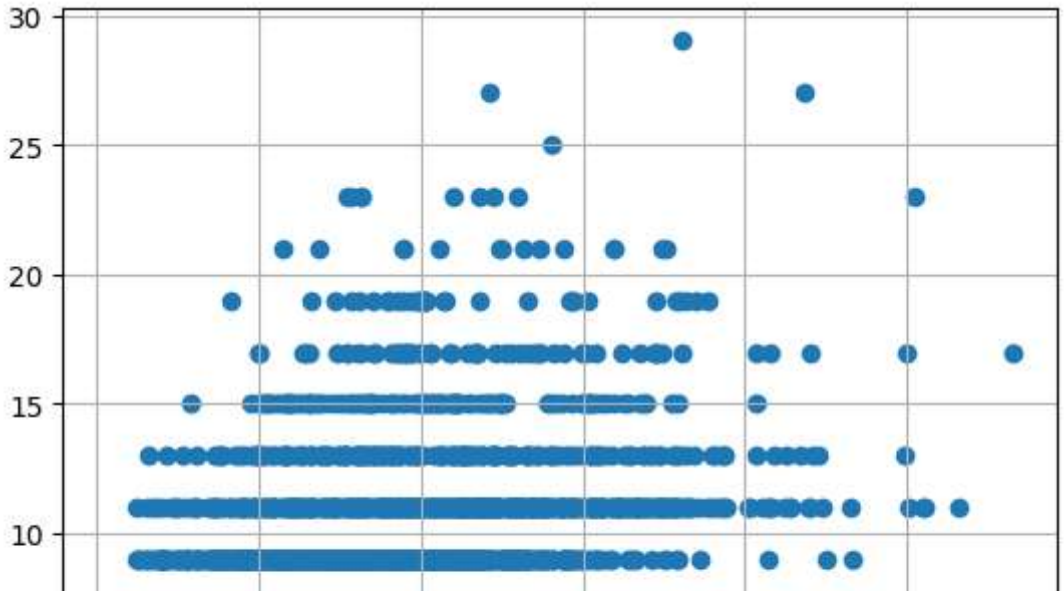



```
data = data[(data['Height'] >= 0.03) & (data['Height'] <= 0.3)]
```


```
plt.scatter(x = data['Height'], y = data['Rings'])
plt.grid(True)
```

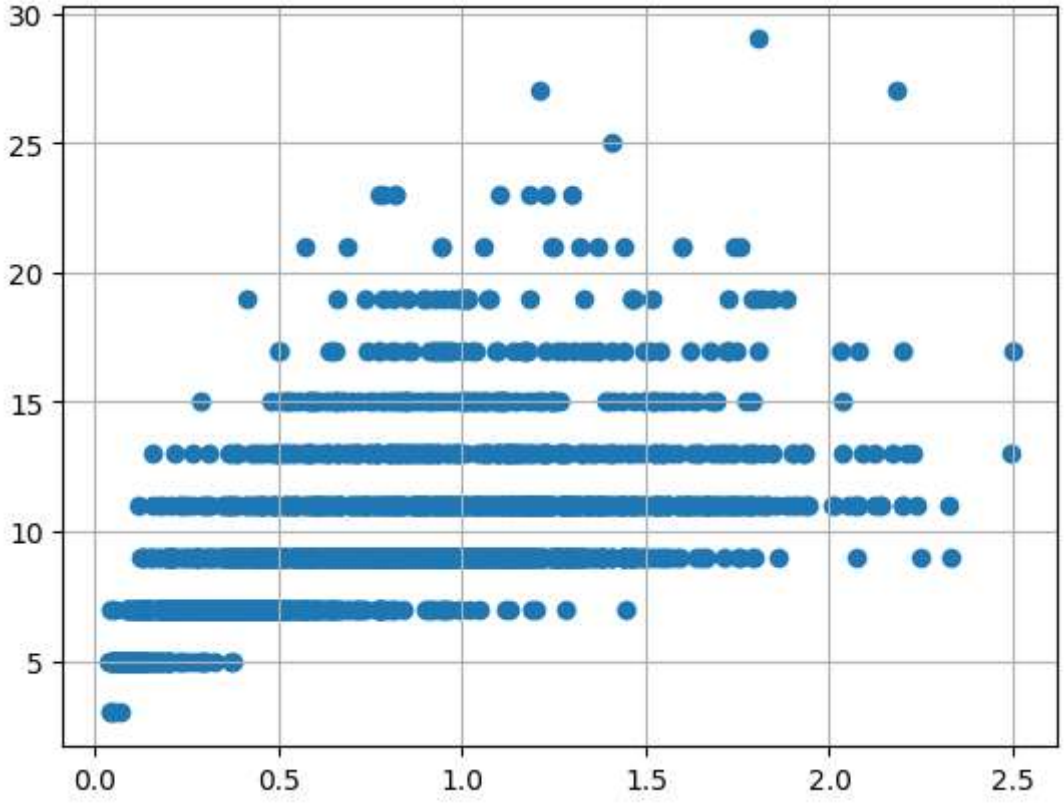



```
plt.scatter(x = data['Whole weight'], y = data['Rings'])
plt.grid(True)
```

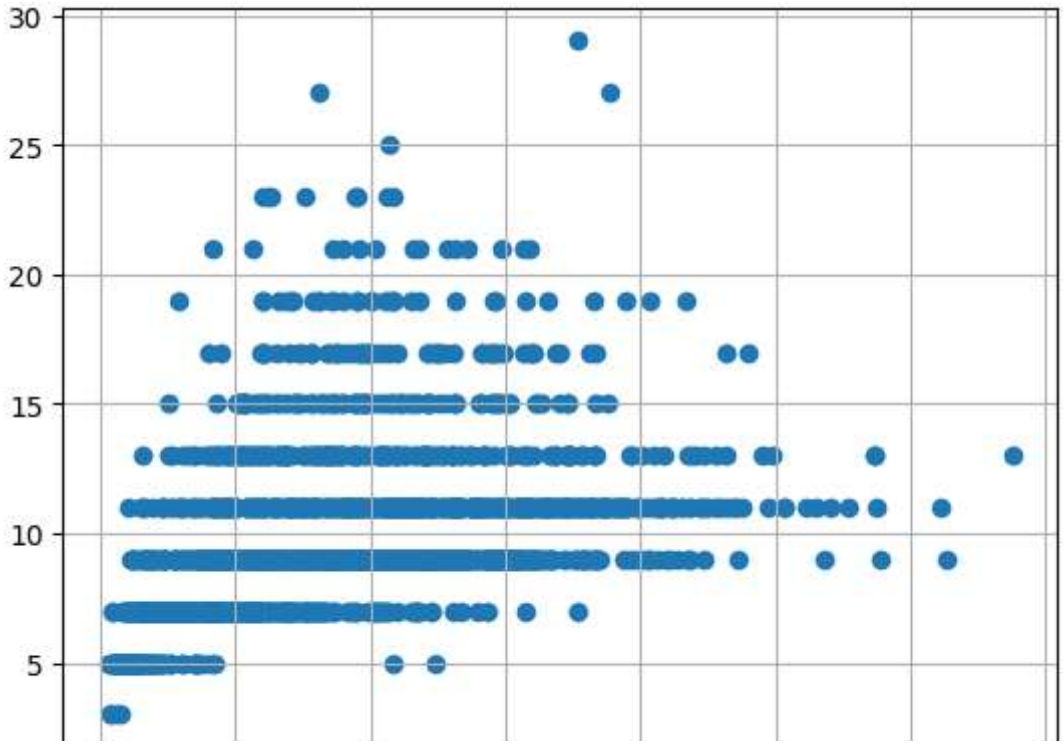



```
data = data [(data['Whole weight'] <= 2.5 )]
```

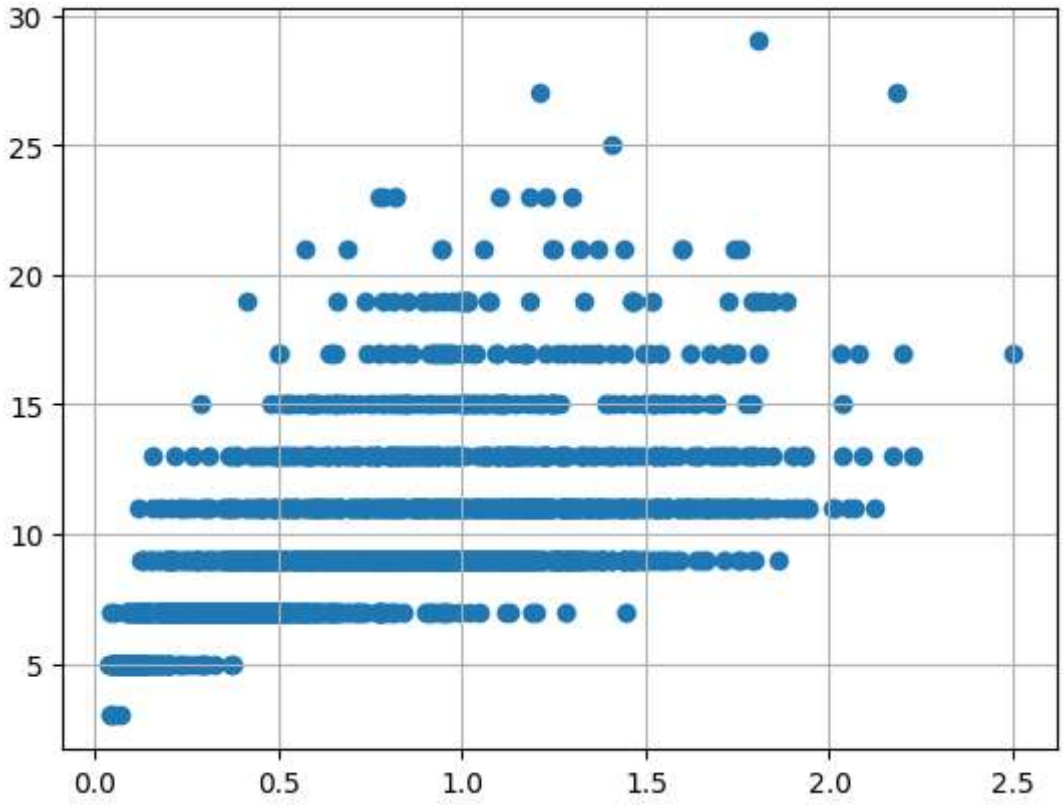
```
plt.scatter(x = data['Whole weight'], y = data['Rings'])
plt.grid(True)
```




```
plt.scatter(x = data['Shucked weight'], y = data['Rings'])
plt.grid(True)
```

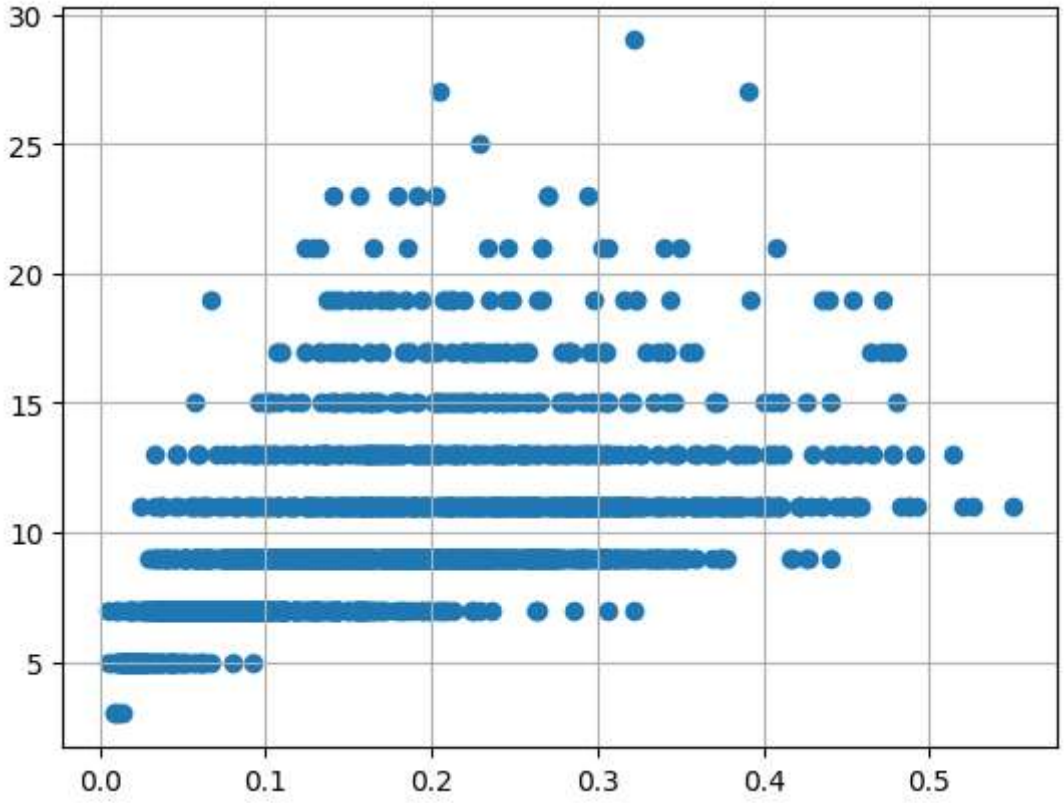



```
data = data [(data['Shucked weight'] <=1)]


plt.scatter(x = data['Whole weight'], y = data['Rings'])
plt.grid(True)
```



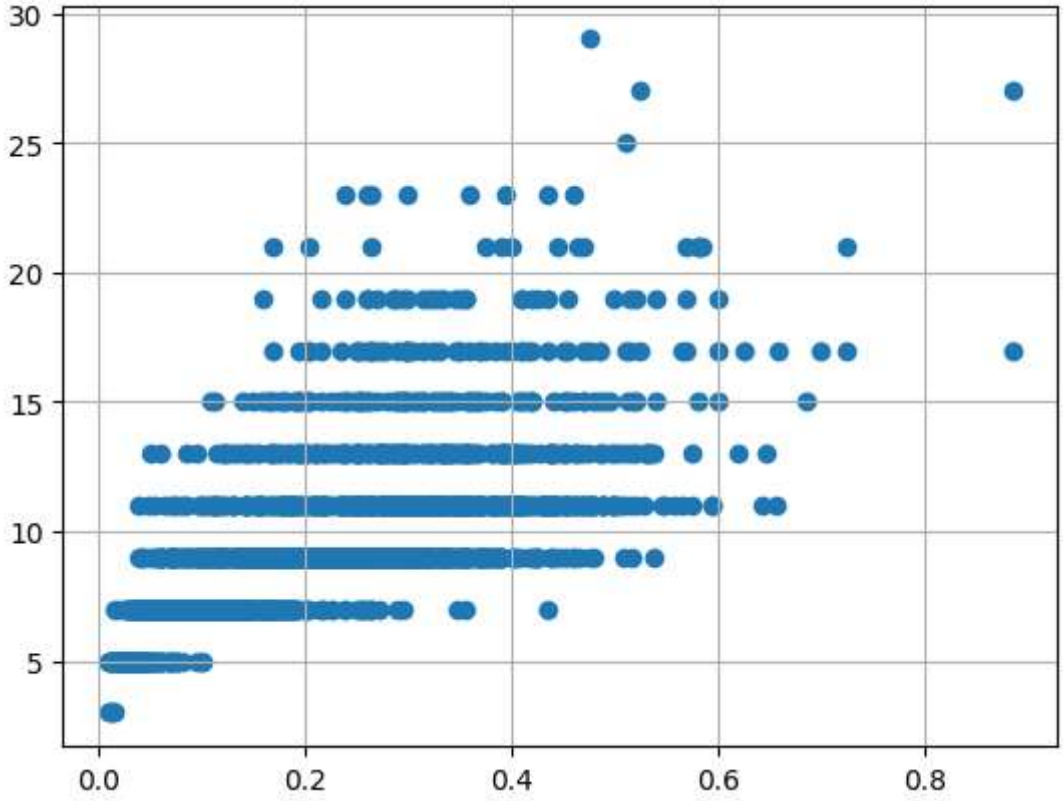

```
plt.scatter(x = data['Viscera weight'], y = data['Rings'])
```

```
plt.grid(True)
```



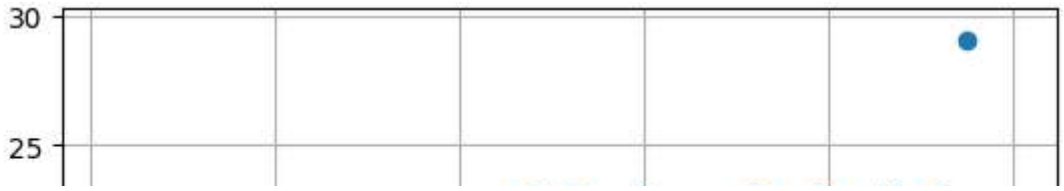

```
data = data [(data['Viscera weight'] <=0.5)]


plt.scatter(x = data['Viscera weight'], y = data['Rings'])
plt.grid(True)
```

```
plt.scatter(x = data['Shell weight'], y = data['Rings'])
plt.grid(True)
```




```
data = data [(data['Shell weight'] <=0.5)]
```

```
plt.scatter(x = data['Shell weight'], y = data['Rings'])
plt.grid(True)
```



Encoding


```
label_encoder = preprocessing.LabelEncoder()
data['Sex'] = label_encoder.fit_transform(data['Sex'])
data['Sex'].unique()
```

```
array([2, 0, 1])
```


```
data.dtypes
```

```
Sex                int32
Length           float64
Diameter         float64
Height           float64
Whole weight     float64
Shucked weight   float64
Viscera weight   float64
Shell weight     float64
Rings              int64
dtype: object
```

Splitting

```
x = data
y = data['Rings'] + 1.5
x
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	2	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
1	2	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
2	0	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9

y

```
0       16.5
1        8.5
2       10.5
4        8.5
8       10.5
        ...
4164     8.5
4165     8.5
4167    10.5
4172    12.5
4174    10.5
Name: Rings, Length: 2013, dtype: float64
```

Building model (train & testing)

```
standardScale = StandardScaler()
x = standardScale.fit_transform(x)
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size= 0.25)
```

```
regressor = RandomForestRegressor(n_estimators= 500, random_state=0)
```

```
regressor.fit(x_train, y_train)
regressor.score(x_train, y_train)*100
```

```
99.95402799307415
```

```
regressor.fit(x_test, y_test)
regressor.score(x_test, y_test)*100
```

```
99.970912084474
```

Colab paid products - Cancel contracts here