In [2]:

```
import mysql.connector
```

In [3]:

```
import pandas as pd    #connecting with Mysql and getting the data student marks data.
con=mysql.connector.connect(
    host="localhost",
    port=3306,
    user="root",
    password="welcome01",
    database="studentdb"

)
```

In [4]:

```
df=pd.read_sql_query("select * from student",con)
```

In [5]:

```
import numpy
print("Numpy ::", numpy.__version__)
import pandas
print("Pandas ::", numpy.__version__)
import numpy as np
```

```
Numpy :: 1.18.1
Pandas :: 1.18.1
```

In [50]:

```
df
```

Out[50]:

	Unnamed: 0	Gender	DOB	Maths	Physics	Chemistry	English	Biology	Economics	Histo
0	John	M	05-04-1988	55	45	56	87	21	52	
1	Suresh	M	04-05-1987	75	96	78	64	90	61	
2	Ramesh	M	25-05-1989	25	54	89	76	95	87	
3	Jessica	F	12-08-1990	78	96	86	63	54	89	
4	Jennifer	F	02-09-1989	58	96	78	46	96	77	
5	Annu	F	05-04-1988	45	87	52	89	55	89	
6	pooja	F	04-05-1987	55	64	61	58	75	58	
7	Ritesh	M	25-05-1989	54	76	87	56	25	56	
8	Farha	F	12-08-1990	55	63	89	75	78	75	
9	Mukesh	M	02-09-1989	96	46	77	83	58	83	

In [14]:

```
df.tail(5)
```

Out[14]:

	Unnamed: 0	Gender	DOB	Maths	Physics	Chemistry	English	Biology	Economics	Histo
5	Annu	F	05-04-1988	45	87	52	89	55	89	
6	pooja	F	04-05-1987	55	64	61	58	75	58	
7	Ritesh	M	25-05-1989	54	76	87	56	25	56	
8	Farha	F	12-08-1990	55	63	89	75	78	75	
9	Mukesh	M	02-09-1989	96	46	77	83	58	83	

In [15]:

```
df['Maths'].mean()
```

Out[15]:

```
59.6
```

In [16]:

```
df['Physics'].mean()
```

Out[16]:

```
72.3
```

In [17]:

```
df['Chemistry'].mean()
```

Out[17]:

```
75.3
```

In [18]:

```
df['English'].mean()
```

Out[18]:

```
69.7
```

In [19]:

```
df['Biology'].mean()
```

Out[19]:

```
64.7
```

In [20]:

```
df['Economics'].mean()
```

Out[20]:

```
72.7
```

In [21]:

```
df['History'].mean()
```

Out[21]:

```
69.7
```

In [22]:

```
df['Civics'].mean()
```

Out[22]:

```
60.5
```

In [23]:

```
df['Maths'].median()
```

Out[23]:

```
55.0
```

In [24]:

```
df['Maths'].var()
```

Out[24]:

```
378.71111111111105
```

In [25]:

```
df['Maths'].std()
```

Out[25]:

```
19.46050130677807
```

In [28]:

```
np.percentile(a,100)
```

Out[28]:

```
10.0
```

In [29]:

```
np.percentile(a,75)
```

Out[29]:

```
8.6
```

In [30]:

```
np.percentile(a,50)
```

Out[30]:

```
7.0
```

In [31]:

```
np.percentile(a,60)
```

Out[31]:

```
7.6
```

In [32]:

```
np.percentile(a,40)
```

Out[32]:

```
5.6000000000000001
```

In [33]:

```
df.describe()  #understanding the dataset.
```

Out[33]:

	Maths	Physics	Chemistry	English	Biology	Economics	History	Civic
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000
mean	59.600000	72.300000	75.300000	69.700000	64.700000	72.700000	69.700000	60.500
std	19.460501	20.661559	14.000397	14.453373	26.998148	14.629119	14.453373	25.343
min	25.000000	45.000000	52.000000	46.000000	21.000000	52.000000	46.000000	2.000
25%	54.250000	56.250000	65.000000	59.250000	54.250000	58.750000	59.250000	52.250
50%	55.000000	70.000000	78.000000	69.500000	66.500000	76.000000	69.500000	63.000
75%	70.750000	93.750000	86.750000	81.250000	87.000000	86.000000	81.250000	76.250
max	96.000000	96.000000	89.000000	89.000000	96.000000	89.000000	89.000000	89.000

Doing the EDA process

In [34]:

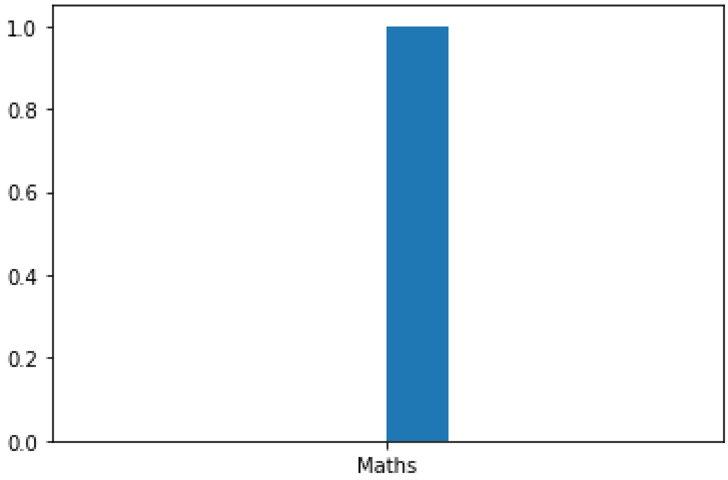
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [35]:

```
plt.hist('Maths')
```

Out[35]:

```
(array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.]),
 array([-0.5, -0.4, -0.3, -0.2, -0.1,  0. ,  0.1,  0.2,  0.3,  0.4,  0.
5]),
 <a list of 10 Patch objects>)
```



In [36]:

```
max_m=df['Maths'].max()
max_m
```

Out[36]:

```
96
```

In [37]:

```
min_m=df['Maths'].min()
min_m
```

Out[37]:

```
25
```

In [38]:

```
r=max_m-min_m
```

In [39]:

```
r
```

Out[39]:

```
71
```

In [40]:

```
q1=df['Maths'].quantile(0.25)
```

In [41]:

```
q1
```

Out[41]:

```
54.25
```

In [42]:

```
q3=df['Maths'].quantile(0.75)
```

In [43]:

```
q3
```

Out[43]:

```
70.75
```

In [44]:

```
IQR=q3-q1
```

In [45]:

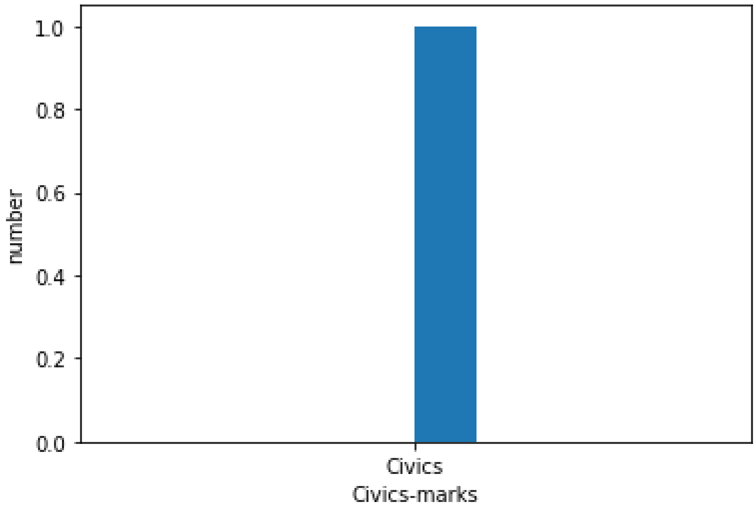
```
IQR
```

Out[45]:

```
16.5
```

In [46]:

```
plt.hist('Civics')
plt.xlabel('Civics-marks')
plt.ylabel('number')
plt.show()
```



In [47]:

```
dfc=df.corr() #Finding the correaltion between the variables.
dfc
```

Out[47]:

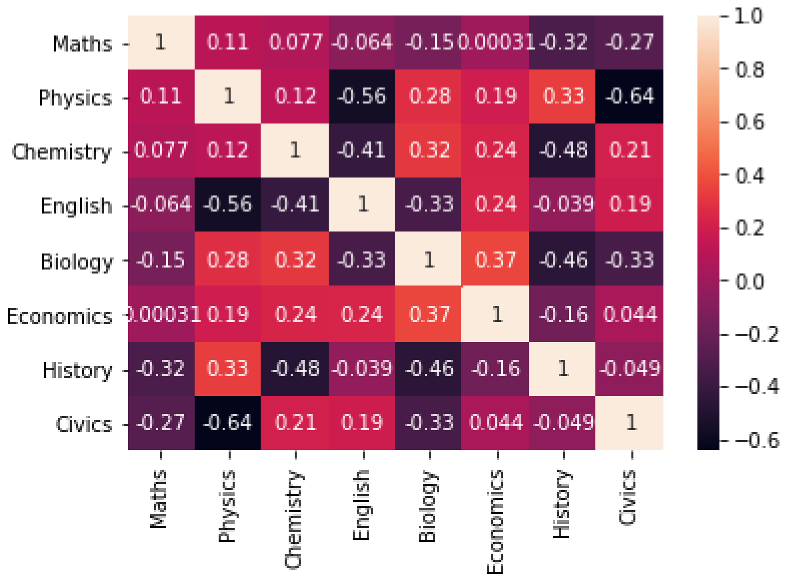
	Maths	Physics	Chemistry	English	Biology	Economics	History	
Maths	1.000000	0.113354	0.076751	-0.064074	-0.146598	0.000312	-0.320846	-0.
Physics	0.113354	1.000000	0.117192	-0.562608	0.279638	0.191851	0.334082	-0.
Chemistry	0.076751	0.117192	1.000000	-0.409680	0.318032	0.235390	-0.482710	0.
English	-0.064074	-0.562608	-0.409680	1.000000	-0.330844	0.237577	-0.038775	0.
Biology	-0.146598	0.279638	0.318032	-0.330844	1.000000	0.370531	-0.456131	-0.
Economics	0.000312	0.191851	0.235390	0.237577	0.370531	1.000000	-0.164953	0.
History	-0.320846	0.334082	-0.482710	-0.038775	-0.456131	-0.164953	1.000000	-0.
Civics	-0.274629	-0.643266	0.210908	0.186705	-0.330060	0.043905	-0.048686	1.

In [48]:

```
sns.heatmap(dfc,annot=True)
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcbcc74fc8>
```

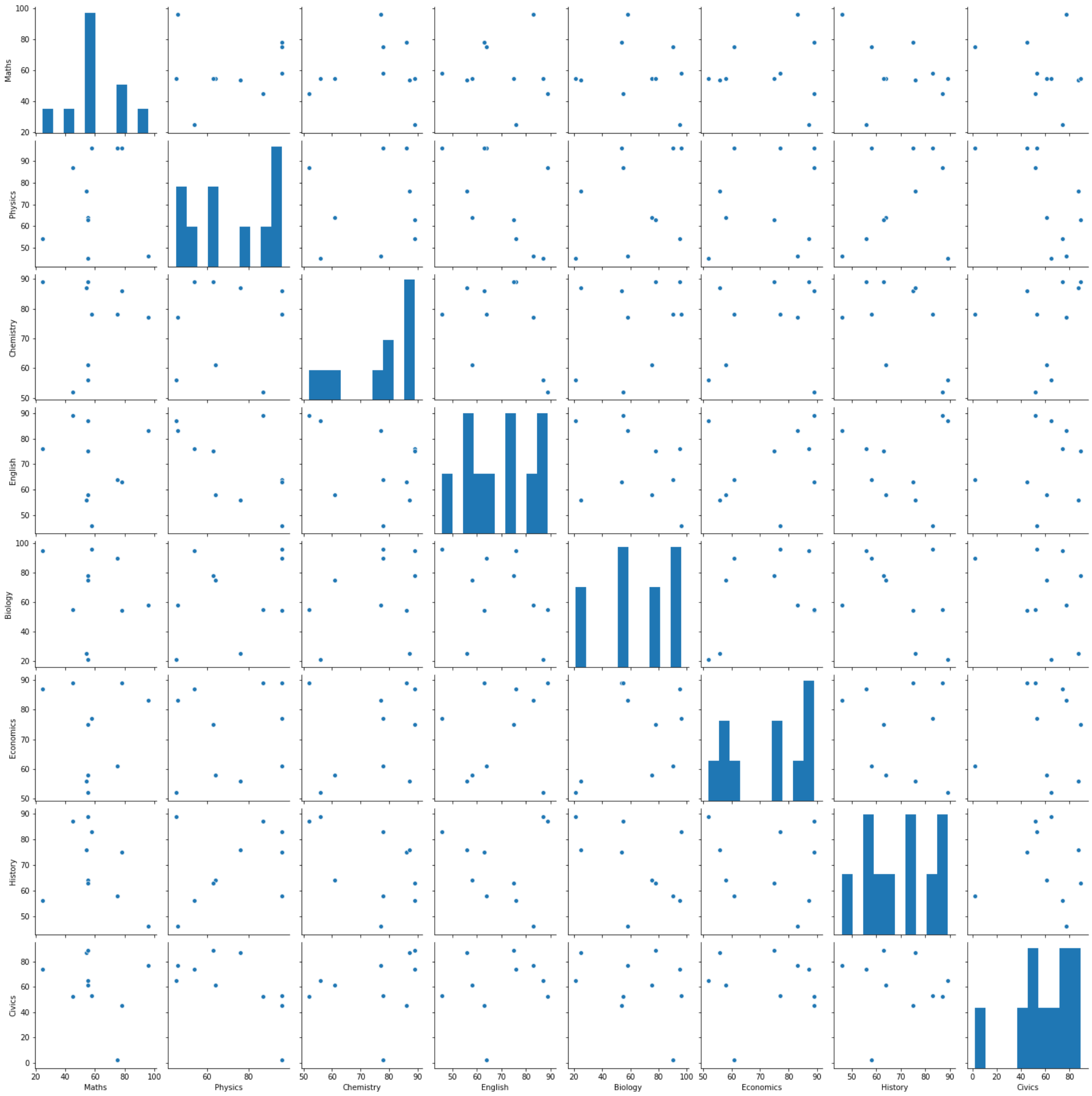


In [49]:

```
sns.pairplot(df) #Bivariate distribution analysis between two numerical variables.
```

Out[49]:

```
<seaborn.axisgrid.PairGrid at 0x1bcbcdd9a08>
```

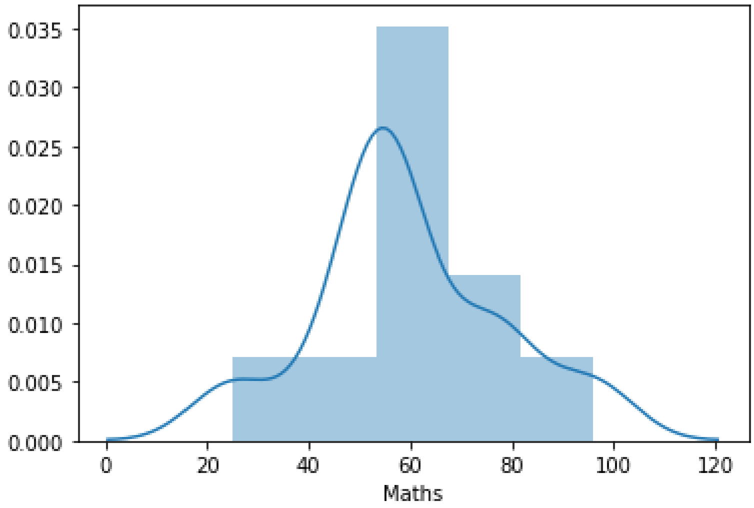


In [51]:

```
sns.distplot(df['Maths']) #finding the univariate distribution of data.
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcbcd2ea08>
```

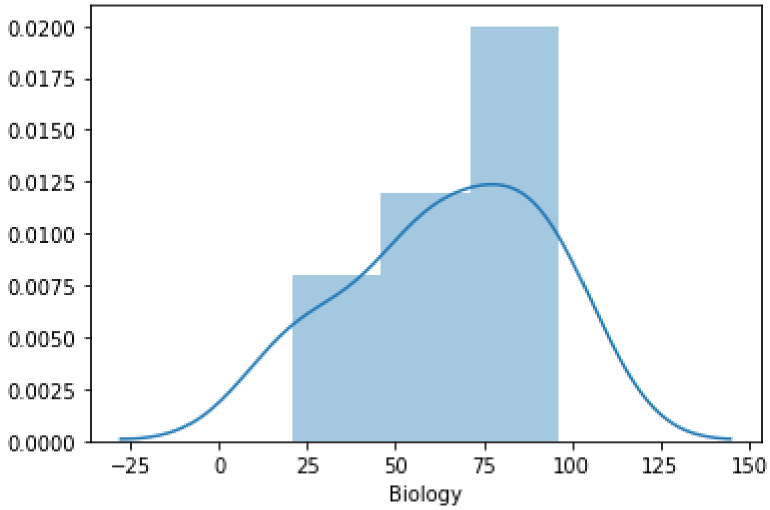


In [52]:

```
sns.distplot(df['Biology'])
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcbff05f88>
```

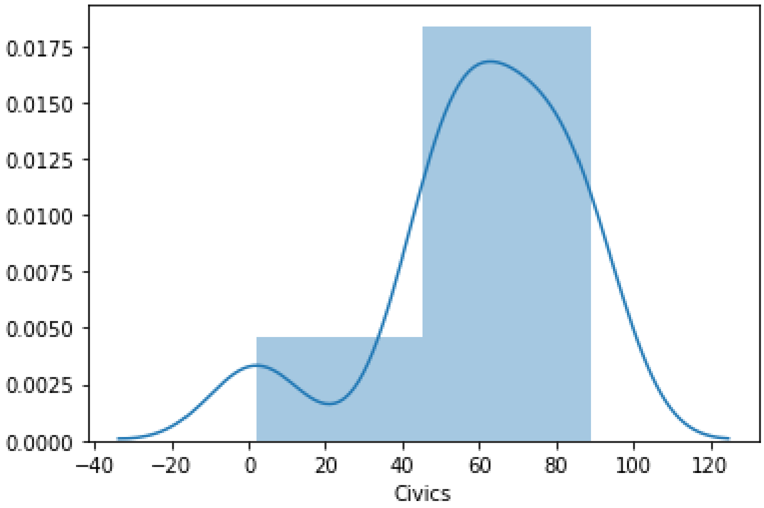


In [53]:

```
sns.distplot(df['Civics'])
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcc078a248>
```

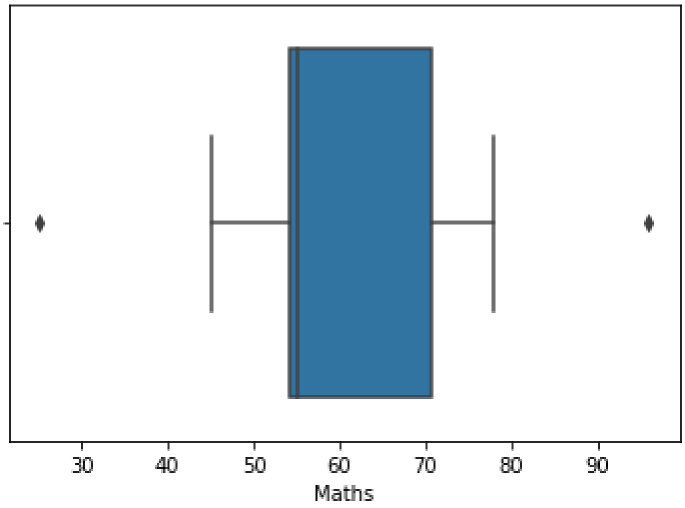


In [55]:

```
sns.boxplot(df['Maths'])
```

Out[55]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcc0943dc8>
```

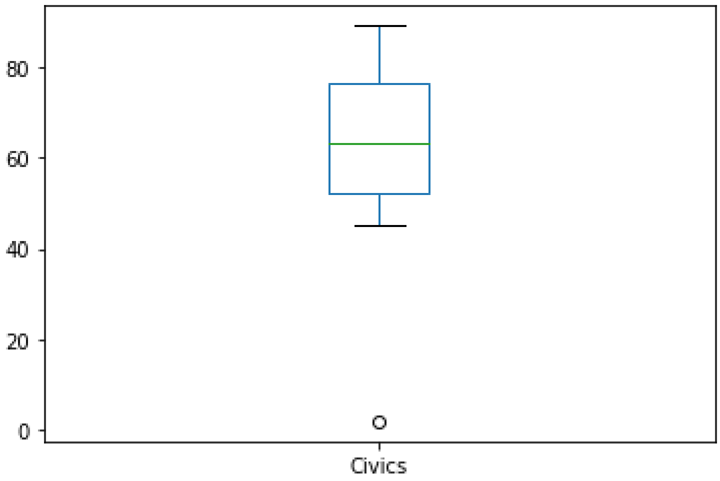


In [76]:

```
df['Civics'].plot.box() #fivepoint summary analyais
```

Out[76]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bef18cde08>
```

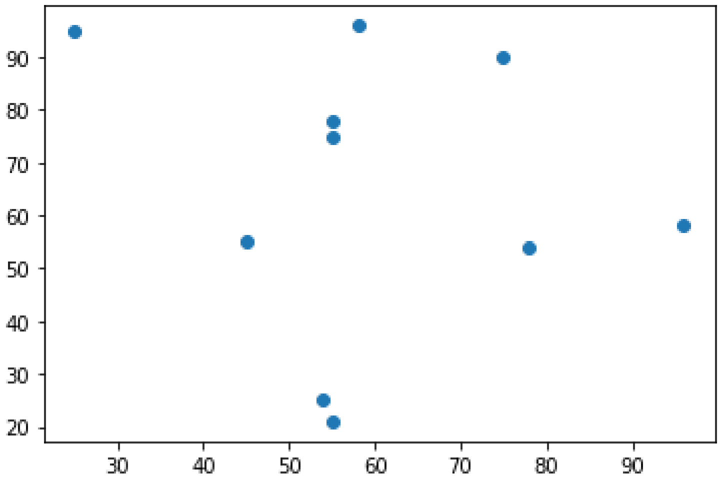


In [77]:

```
plt.scatter(df['Maths'],df['Biology']) #finding out if their any relationship between these two numeric vairables.
```

Out[77]:

```
<matplotlib.collections.PathCollection at 0x1bef19af788>
```

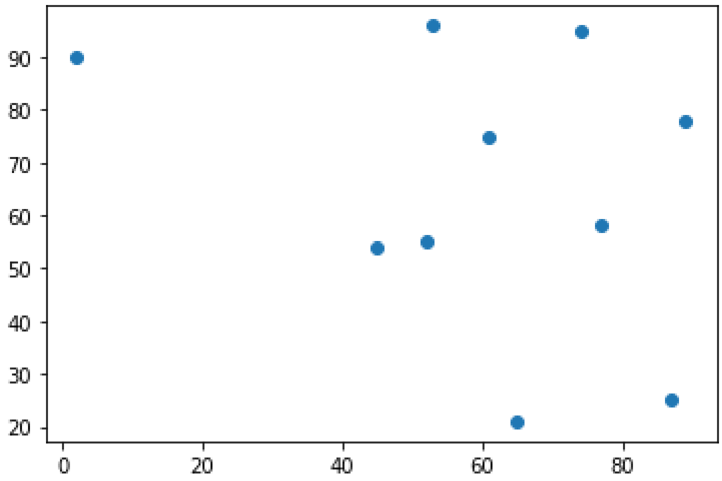


In [78]:

```
plt.scatter(df['Civics'],df['Biology'])
```

Out[78]:

```
<matplotlib.collections.PathCollection at 0x1bef1a1cc88>
```



Conclusion: We have loaded the data and then i have done the data mining process by fetching the student marksdata into the notebook and performed exploratory data analysis on the dataset.