In [ ]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [6.0, 3.0]

# Reading Data
data= pd.read_csv('realdataOne.csv')
# data= pd.read_csv('realdataTwo.csv')
print(data.shape)
data.head()
```

```
(10, 2)
```

Out[ ]:

	X_Values	Y_Values
0	1.0	1.8
1	2.0	2.4
2	3.3	2.3
3	4.3	3.8
4	5.3	5.3

In [ ]:
```
#Collecting X and Y
X= data['X_Values'].values
Y= data['Y_Values'].values
```

In [ ]:
```
y_values=np.array(Y)
print("Sum of all the Y_Values: ", y_values.sum())

xx = [x_values * x_values for x_values, x_values in zip(X, X)]
xx_values=np.array(xx)
print("Sum of all the XX_Values: ", xx_values.sum())
```

```
Sum of all the Y_Values:  32.5
Sum of all the XX_Values:  121.34
```

In [ ]:
```
xy = [xx_values * y_values for xx_values, y_values in zip(xx, Y)]
xy_values=np.array(xy)
print("Sum of all the XY_Values: ", round(xy_values.sum(),2))

xxx = [xx_values * xx_values for xx_values, xx_values in zip(xx, xx)]
xxx_values=np.array(xxx)
print("Sum of all the XX_Values: ", xxx_values.sum())
```

```
Sum of all the XY_Values:  509.76
Sum of all the XX_Values:  2329.9862
```

In [ ]:
```
# Using the formula to calculate slope(b)
n= len(X)
b = round(((n* xy_values.sum())-(xx_values.sum()*y_values.sum()))/ ((n*xxx_values.sum(
print("slope(b) is: " , b)
```

```
slope(b) is:  0.1346
```

In [ ]:
```
# # Using the formula to calculate intercept(a)
a = round((y_values.sum() - (b*xx_values.sum()))/n ,4)
```

```
print("intercept(a): ", a)
```

```
intercept(a):  1.6168
```

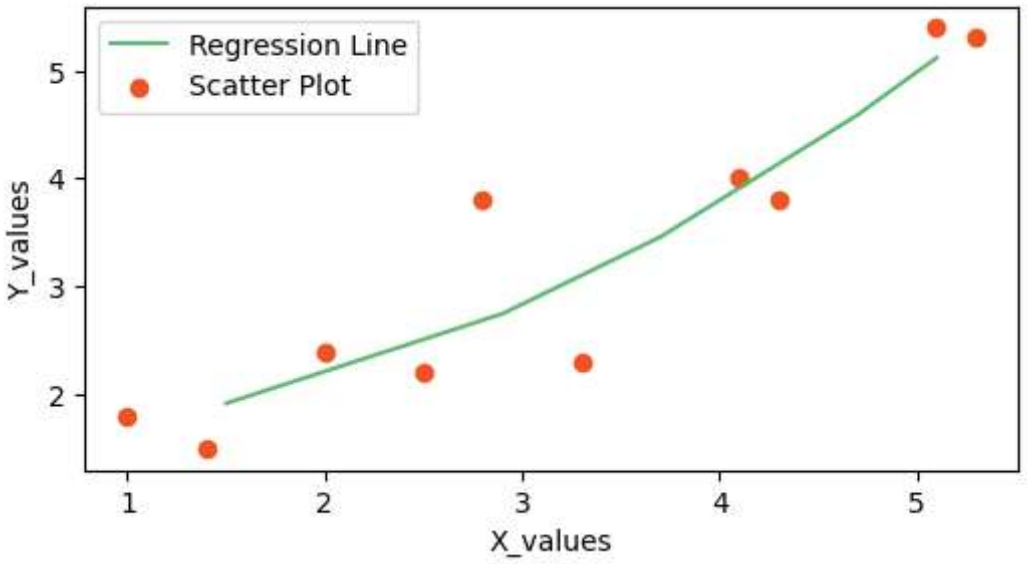
*** Training phase **

In [ ]:
```
# Then substitute Intercept(a) and Slope(b) in regression equation formula
# Regression Equation(y) = a + bx
x = np.array(X)
y=a+b*x**2
print(y)
```

```
[1.7514   2.1552   3.082594 4.105554 5.397714 1.880616 2.45805  2.672064
 3.879426 5.117746]
```

In [ ]:
```
plt.plot(x,y, color='#58b970', label='Regression Line')
plt.scatter(X,Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('X_values')
plt.ylabel('Y_values')
plt.legend()
plt.show()
```



** validation phase ***

In [ ]:
```
# After calculating a1, b1, a2, b2 in Training Phase, the values are not changed with
# Only ŷ values are changed with the new Real Data Sets.
# Regression Equation(y) = a + bx
x = np.array([1.5,2.9,3.7,4.7,5.1])
y=a+b*x**2
print(y)
```

```
[1.91965  2.748786 3.459474 4.590114 5.117746]
```