Regression

www.hbpatel.in

Regression analysis is a form of predictive modelling technique which investigates the **relationship** between a dependent and independent variable

Three major **uses** for Regression Analysis:

- Determining the strength of predictors
- Forecasting an effect
- Trend forecasting

Linear Regression

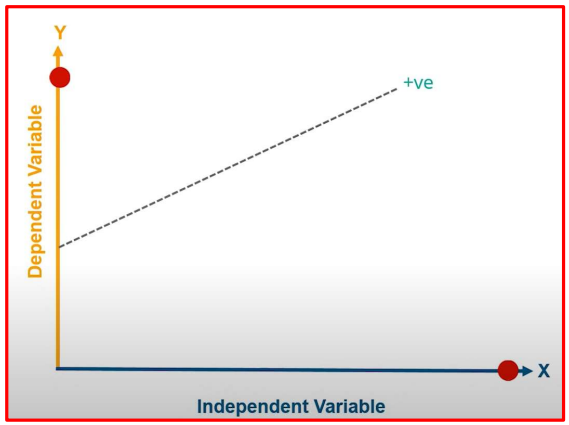
www.hbpatel.in

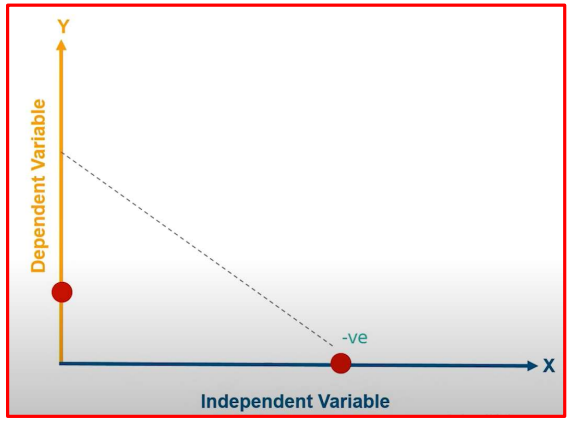
- Data is modelled using a straight line
- It is used with continuous variable
- Value of a variable is either an output or predicted
- Accuracy is measured by loss

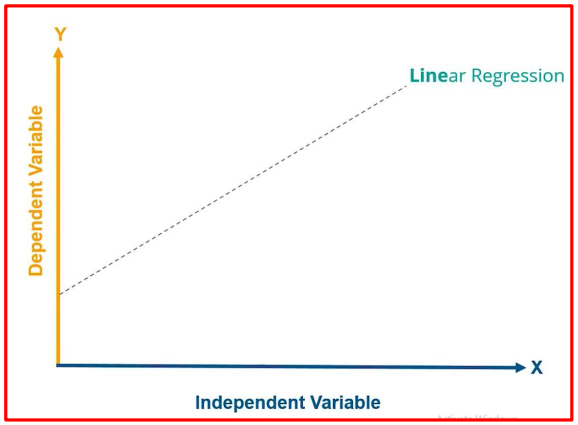
Linear Regression

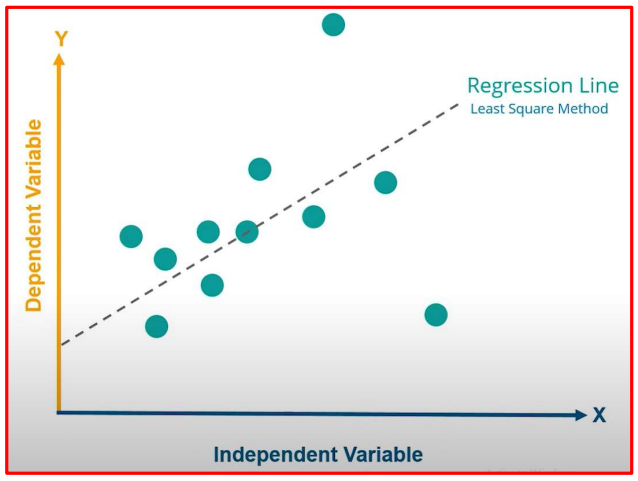
Source: Linear Regression | Edureka: https://youtu.be/E5RjzSK0fvY

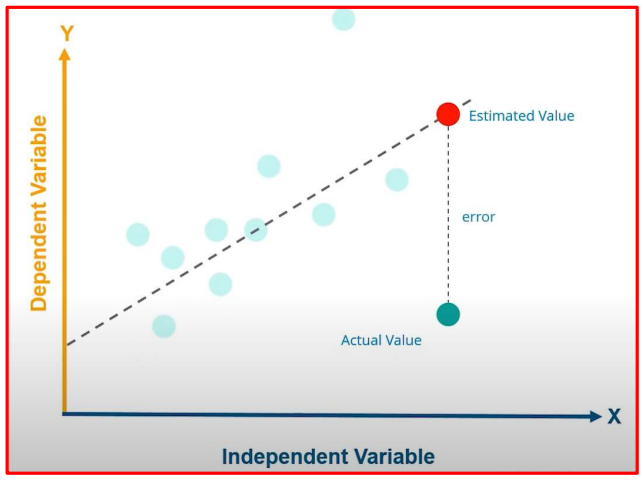
www.hbpatel.in

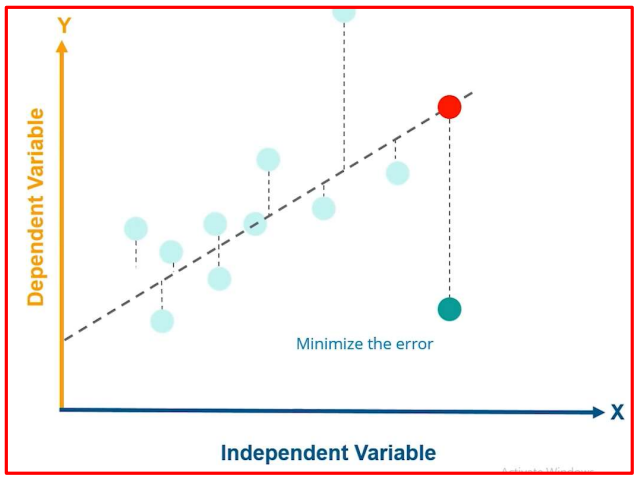








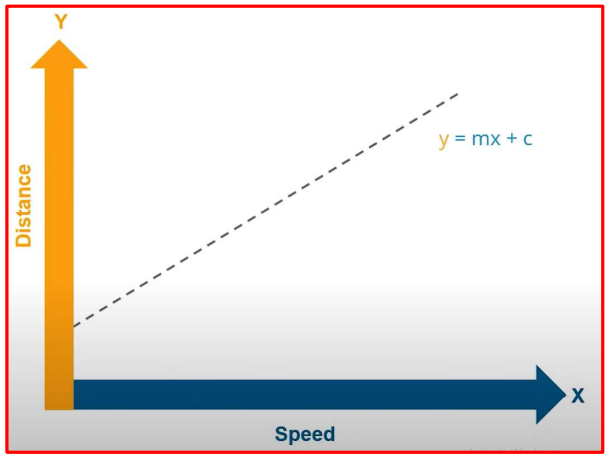




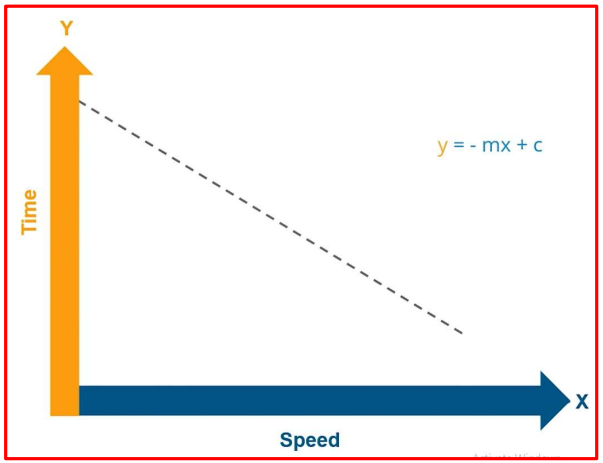
Linear Regression

Source: Linear Regression | Edureka: https://youtu.be/E5RjzSK0fvY

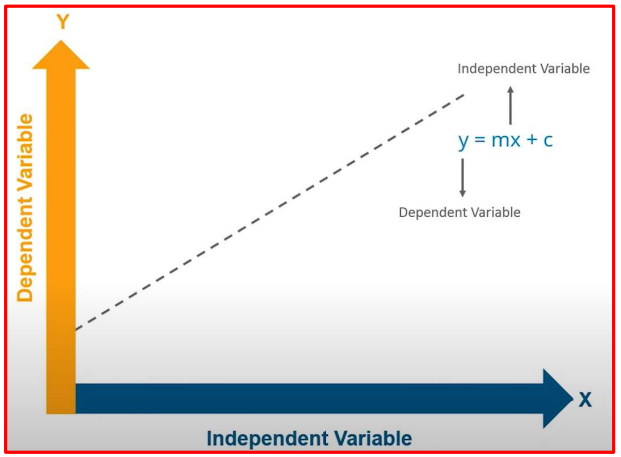
www.hbpatel.in



Time t is constant
x = Speed of vehicle
y = Distance travelled in a fixed duration of time
m = positive slope of the line
c = y intercept of the line



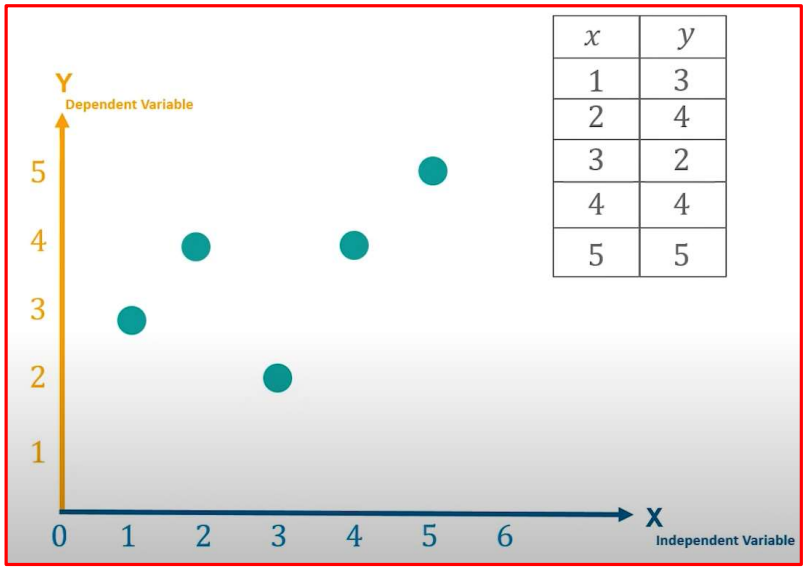
Time t is constant
x = Speed of vehicle
y = Time taken to travel a fixed distance
m = negative slope of the line
c = y intercept of the line



Linear Regression

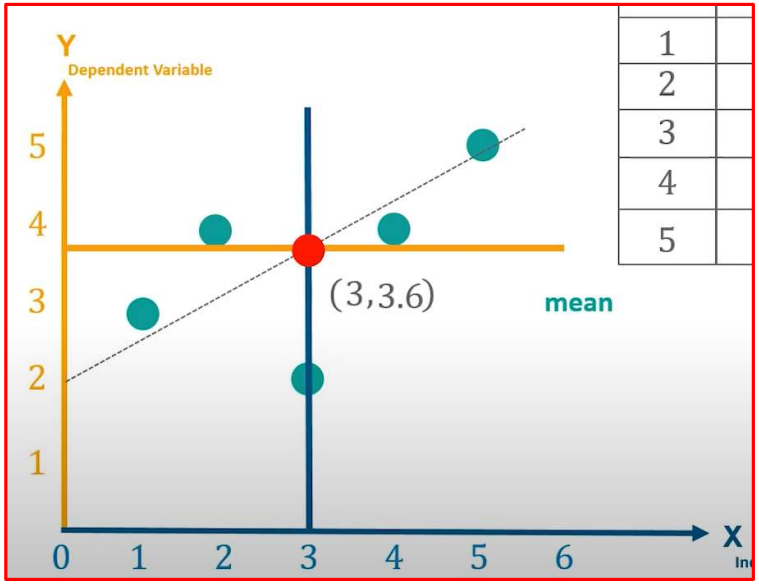
Source: Linear Regression | Edureka: https://youtu.be/E5RjzSK0fvY

www.hbpatel.in



x	y
1	3
2	4
3	2
4	4
5	5

	x	y
	1	3
	2	4
	3	2
	4	4
	5	5
Mean	**3**	**3.6**



1
2
3
4
5

$$m = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

x	y	x – xbar	y - ybar	(x - xbar) x (y - ybar)	square (x - xbar)	
1	3	-2	-0.6	1.2	4	
2	4	-1	0.4	-0.4	1	
3	2	0	-1.6	0	0	
4	4	1	0.4	0.4	1	
5	5	2	1.4	2.8	4	
3	**3.6**			**4**	**10**	**0.4**
Xbar	**ybar**			**∑ (x - xbar) x (y - ybar)**	**∑ square (x - xbar)**	**m**

y = 3.6
x = 3
m = 0.4
y = mx + c
c = y - mx
c = 3.6 - 3x0.4
c = 2.4

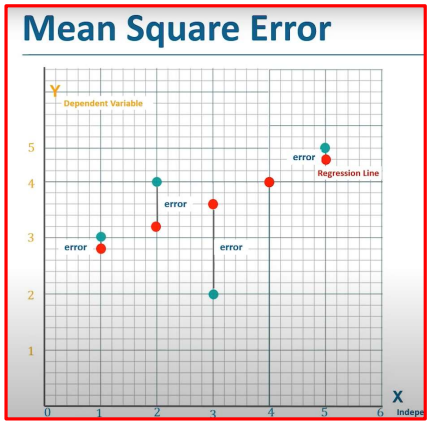
Linear Regression

Source: Linear Regression | Edureka: https://youtu.be/E5RjzSK0fvY

www.hbpatel.in

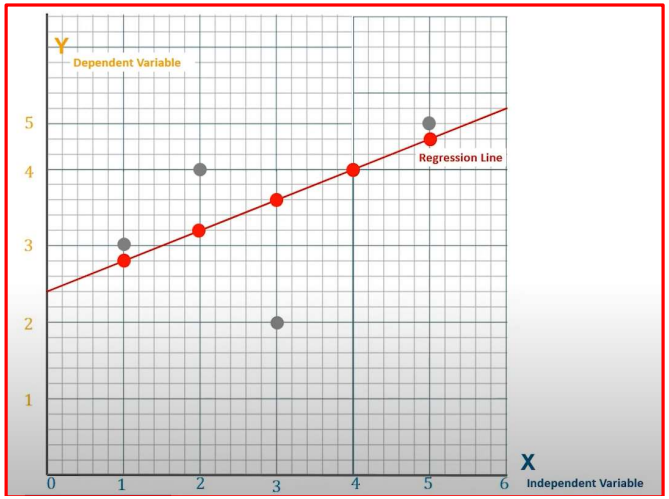
```
y  = mx + c
y  = 0.4x + 2.4
```

x	y = 0.4x + 2.4
1	2.8
2	3.2
3	3.6
4	4
5	4.4



R-Square: Coefficient of Determination OR Coefficient of Multiple Determination

$$R^2 = \frac{\sum (y_p - \bar{y})^2}{\sum (y - \bar{y})^2}$$



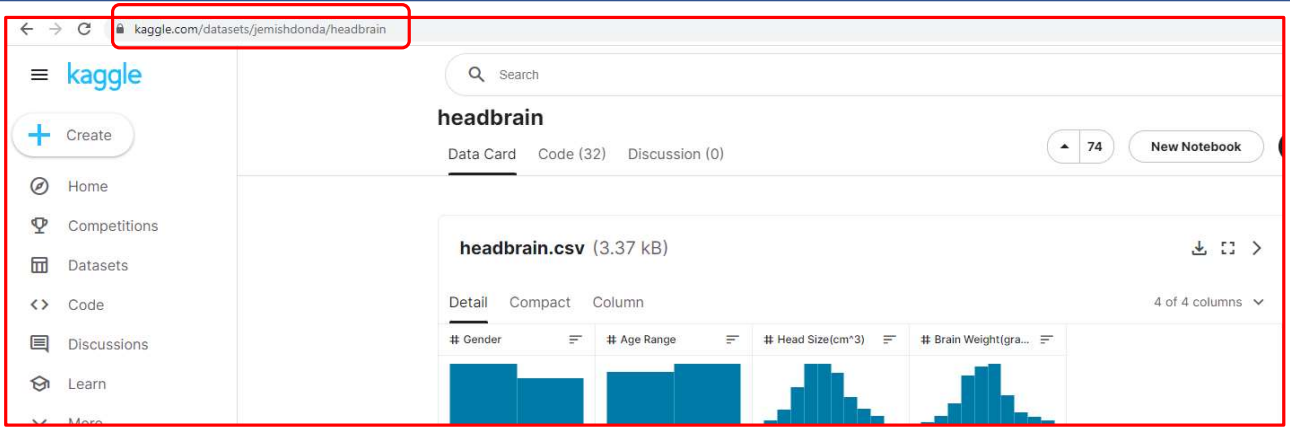
x	y	$y - \bar{y}$	$(y - \bar{y})^2$	y_p	$(y_p - \bar{y})$	$(y_p - \bar{y})^2$
1	3	- 0.6	0.36	2.8	-0.8	0.64
2	4	0.4	0.16	3.2	-0.4	0.16
3	2	−1.6	2.56	3.6	0	0
4	4	0.4	0.16	4.0	0.4	0.16
5	5	1.4	1.96	4.4	0.8	0.64

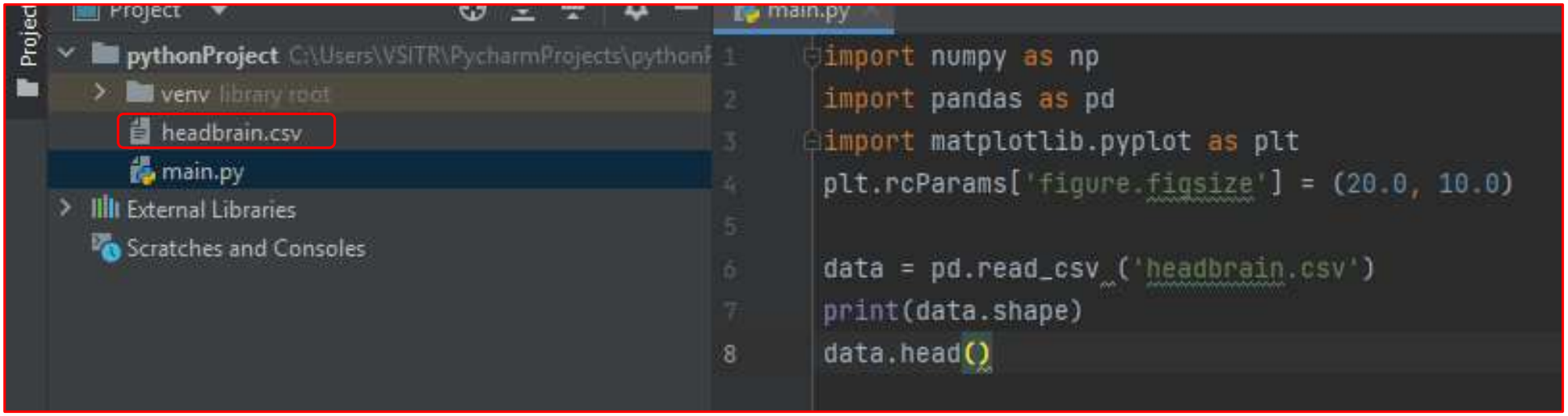
mean y 3.6

$$R^2 = \frac{\sum (y_p - \bar{y})^2}{\sum (y - \bar{y})^2}$$

Linear Regression using Python

www.hbpatel.in





Linear Regression using Python

www.hbpatel.in

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)

data = pd.read_csv('headbrain.csv')
print(data.shape)
print(data)
```

```
(237, 4)
     Gender  Age Range  Head Size(cm^3)  Brain Weight(grams)
0         1          1             4512                 1530
1         1          1             3738                 1297
2         1          1             4261                 1335
3         1          1             3777                 1282
4         1          1             4177                 1590
..      ...        ...              ...                  ...
232       2          2             3214                 1110
233       2          2             3394                 1215
234       2          2             3233                 1104
235       2          2             3352                 1170
236       2          2             3391                 1120

[237 rows x 4 columns]
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)

data = pd.read_csv('headbrain.csv')

X = data['Head Size(cm^3)'].values
Y = data['Brain Weight(grams)'].values

mean_x = np.mean(X)
mean_y = np.mean(Y)

n = len(X)

numer = 0
denom = 0

for i in range (n):
    numer += (X[i] - mean_x) * (Y[i] - mean_y)
    denom += (X[i] - mean_x) ** 2

b1 = numer / denom
b0 = mean_y - (b1 * mean_x)

print(b1, b0)
```

```
0.26342933948939945 325.57342104944223
```

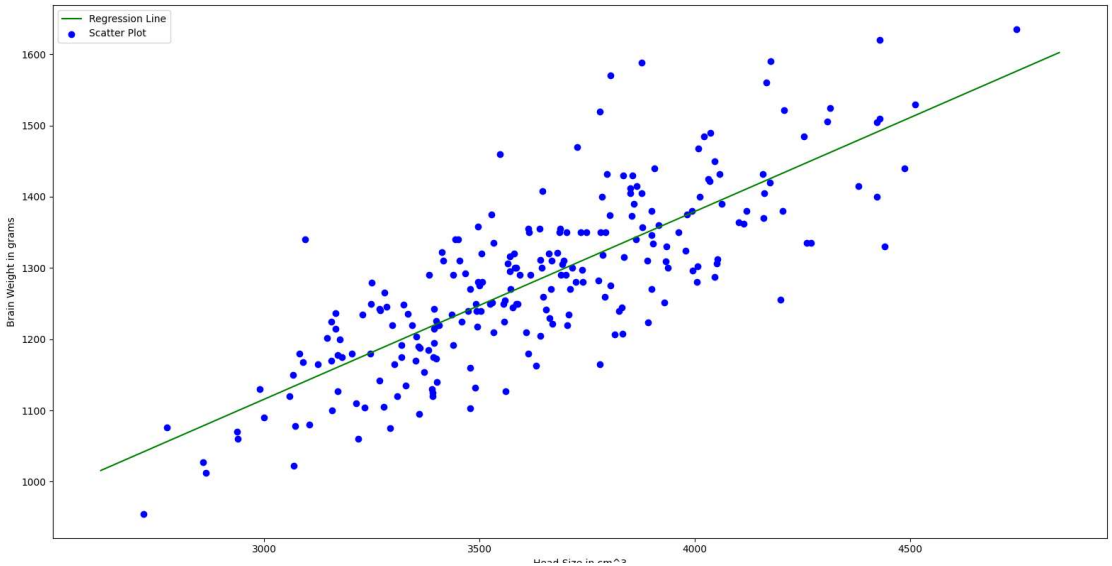
Linear Regression using Python

www.hbpatel.in

```
max_x = np.max(X) + 100
min_x = np.min(X) - 100

x = np.linspace(min_x, max_x, 1000)
y = b0 + b1 * x

#plotting line
plt.plot (x, y, color='green', label='Regression Line')
plt.scatter (X, Y, color='blue', label='Scatter Plot')
plt.xlabel("Head Size in cm^3")
plt.ylabel("Brain Weight in grams")
plt.legend()
plt.show()
```



Linear Regression using Python

How good our model is (using R^2 value)?

```
ss_t = 0
ss_r = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    ss_t += (Y[i] - mean_y) ** 2
    ss_r += (Y[i] - y_pred) ** 2
r2 = 1 - (ss_r / ss_t)
print(r2)
```

```
0.6393117199570003
```