VIT-AP UNIVERSITY

Linear Regression

Dr. Kuppusamy .P
Associate Professor / SCOPE

Linear Regression

- Linear Regression analysis is a statistical (quantitative analysis) tool
- Predictive modeling method to investigate the mathematical relationship between an independent variable (predictor – x) and continuous dependent variable (outcome – y) and
- Predictor shows the changes in Dependent variable (y axis) to the changes in explanatory variables in X axis.
- It uses current information about a phenomenon to predict its future behavior.
- Involves the graphical lines over a set of data points that most closely towards all shape of the data.
- When the data form a set of pairs of numbers, it is interpreted as the observed values of an independent (or predictor) variable X and a dependent (or response) variable Y.

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

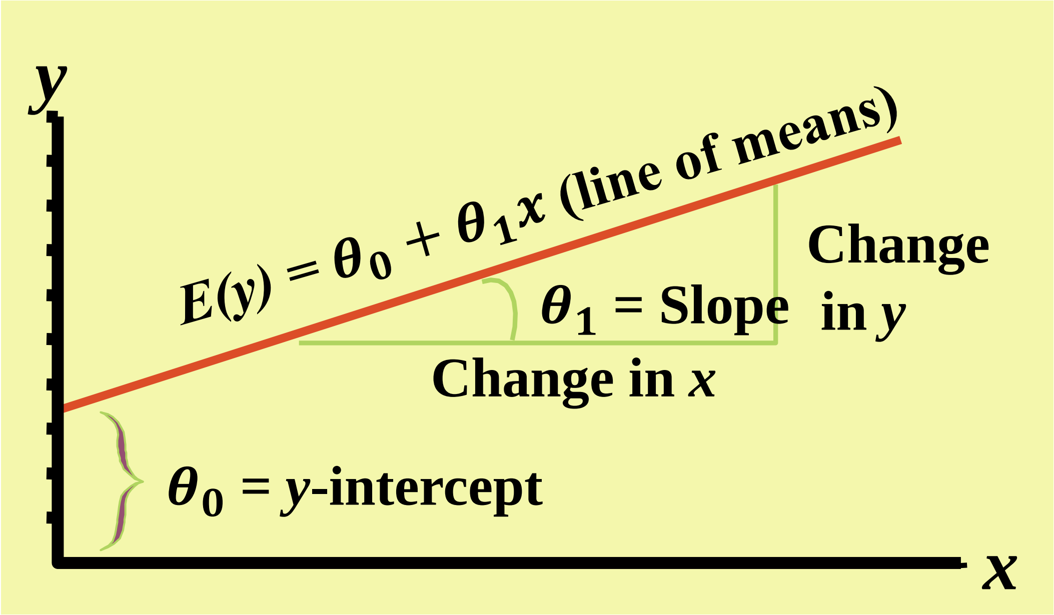
Data model in Simple Linear Regression

- Data is modelled using a straight line with continuous variable
- Relationship between variables is a linear function.
- Linear equation representation with single feature is given:

Population y-intercept　**Population Slope**　**Random Error**

$$y = \theta_0 + \theta_1 x + \varepsilon$$

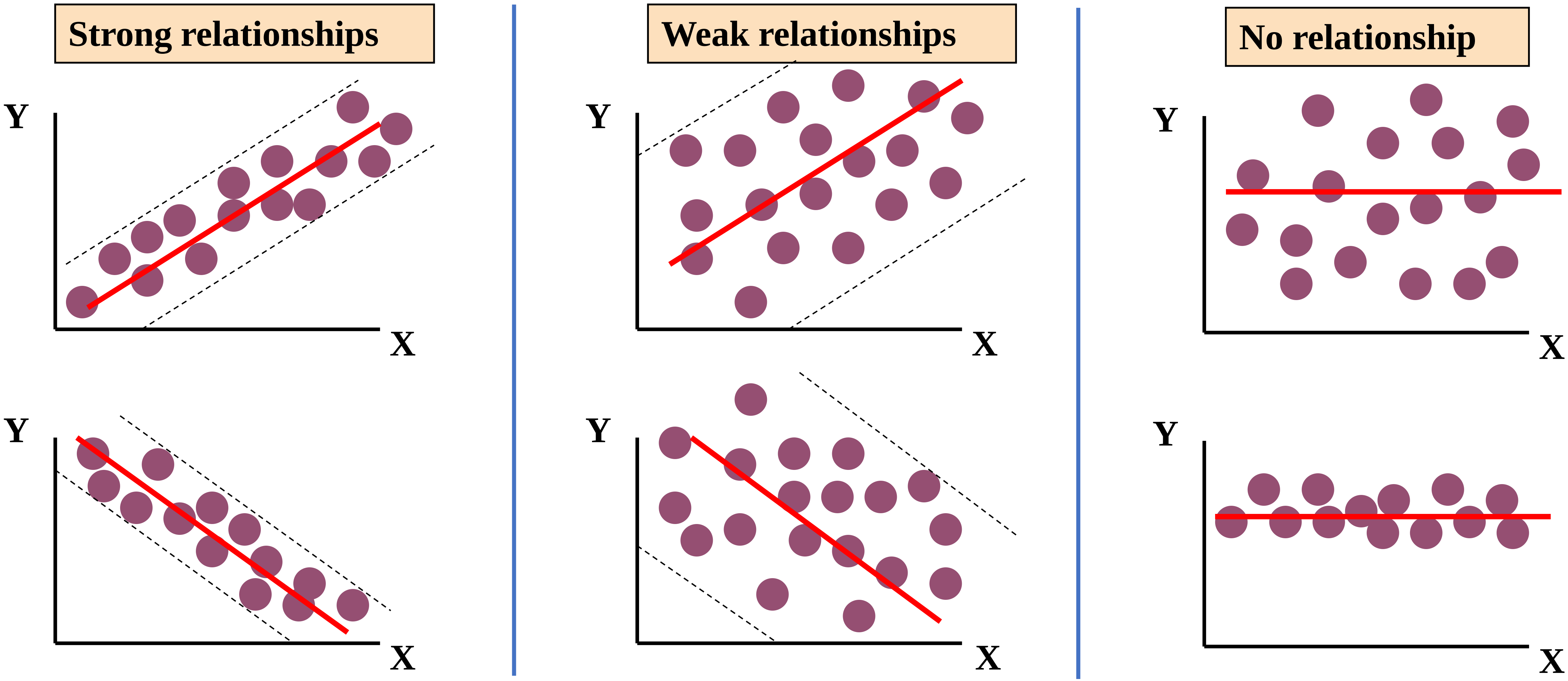
Dependent (Response) Variable　**Independent (Explanatory) Variable**



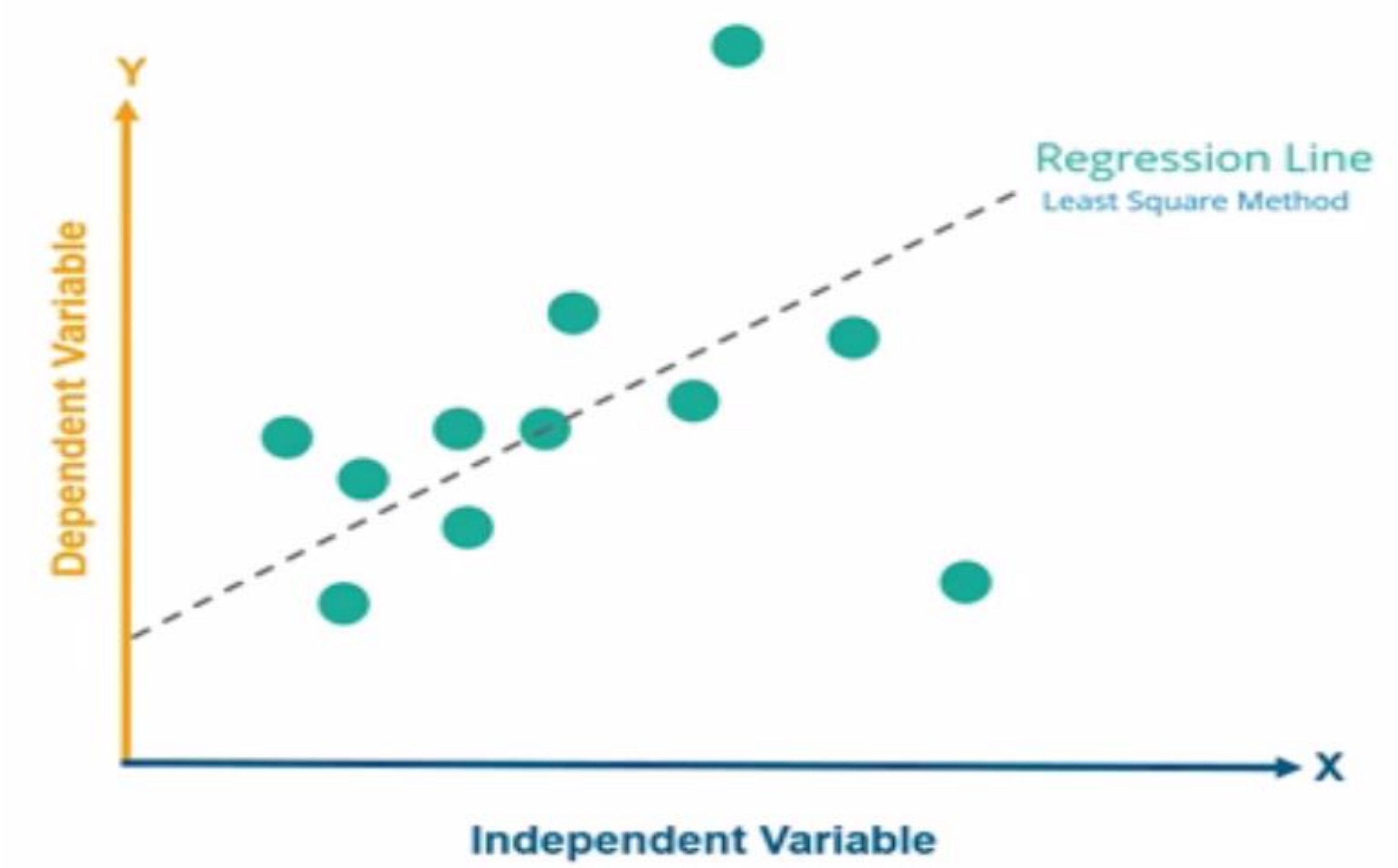
- It is reduced into

$$y' = h_\theta(x) = \theta^T . X$$

Types of Relationships

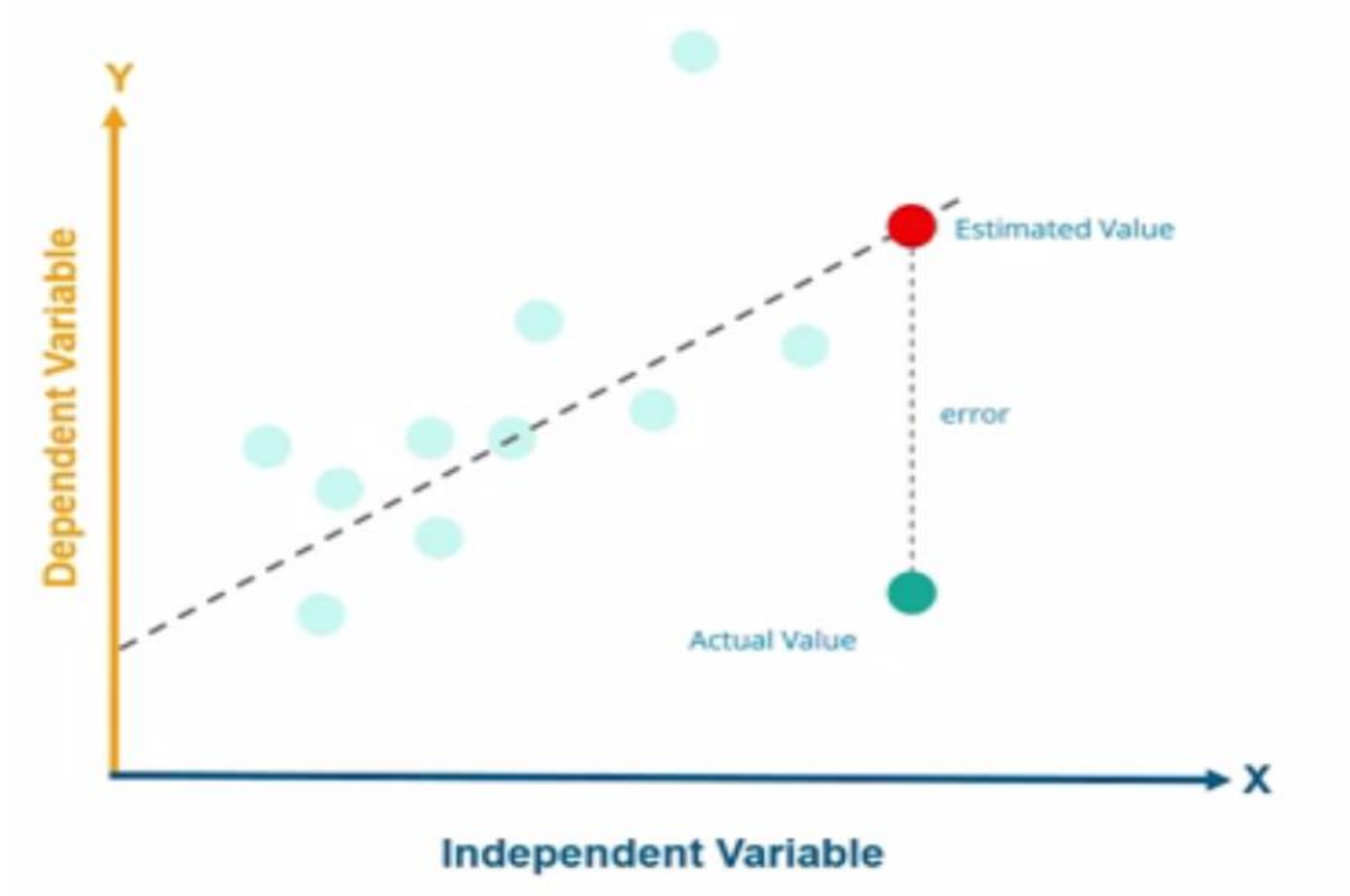


Plot the graph using x and actual y (target) values



Random Error Identification

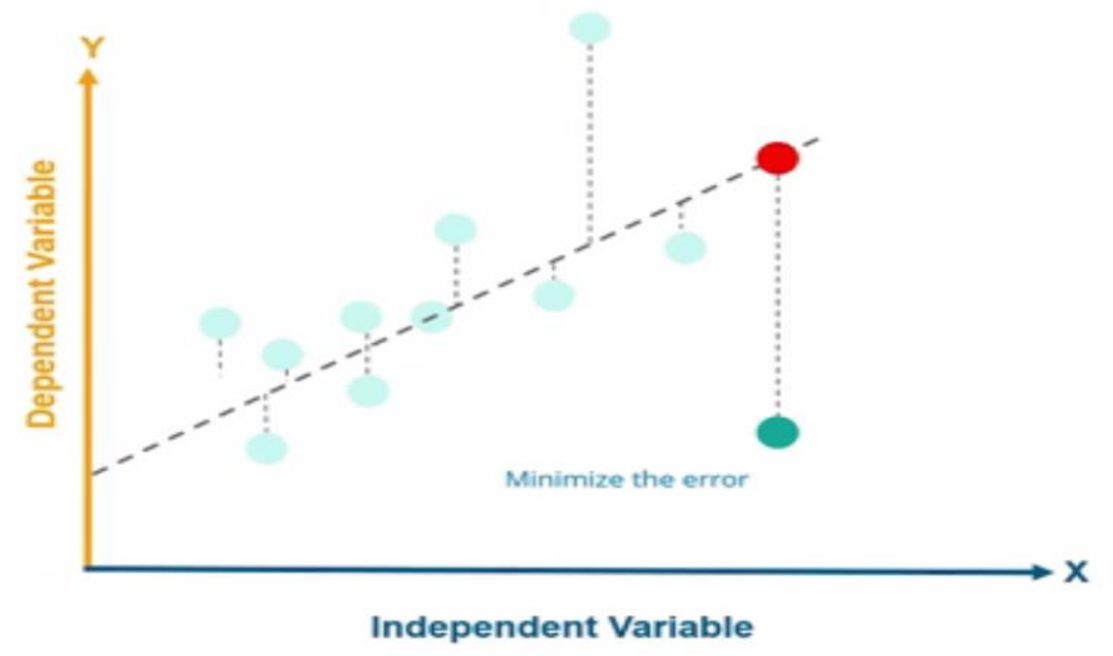
- Random Error ε = Estimated Value (y_i') – Actual Value (y_i)



Minimize the Random Error using Cost Function Least squared (LSE) Method

- Reduce the distance between estimated (predicted) and actual (target) value
- Find the best fit of the line using **least square method.**

Least Squares Method to Minimize the Error



- **Best fit'** means difference between actual *y* values and predicted *y* values are a **minimum**
  - *But* positive differences off-set negative

$$\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n}\hat{\varepsilon}_i^2$$

- $y' = h_\theta(x) = \theta^T . X$
- Least Squares minimizes the Sum of the Squared Differences (SSE)

Minimize the Random Error using Cost Function

- Normal Equation to calculate the weights

$$\theta = (X^T.X)^{-1}.(X^T.y)$$

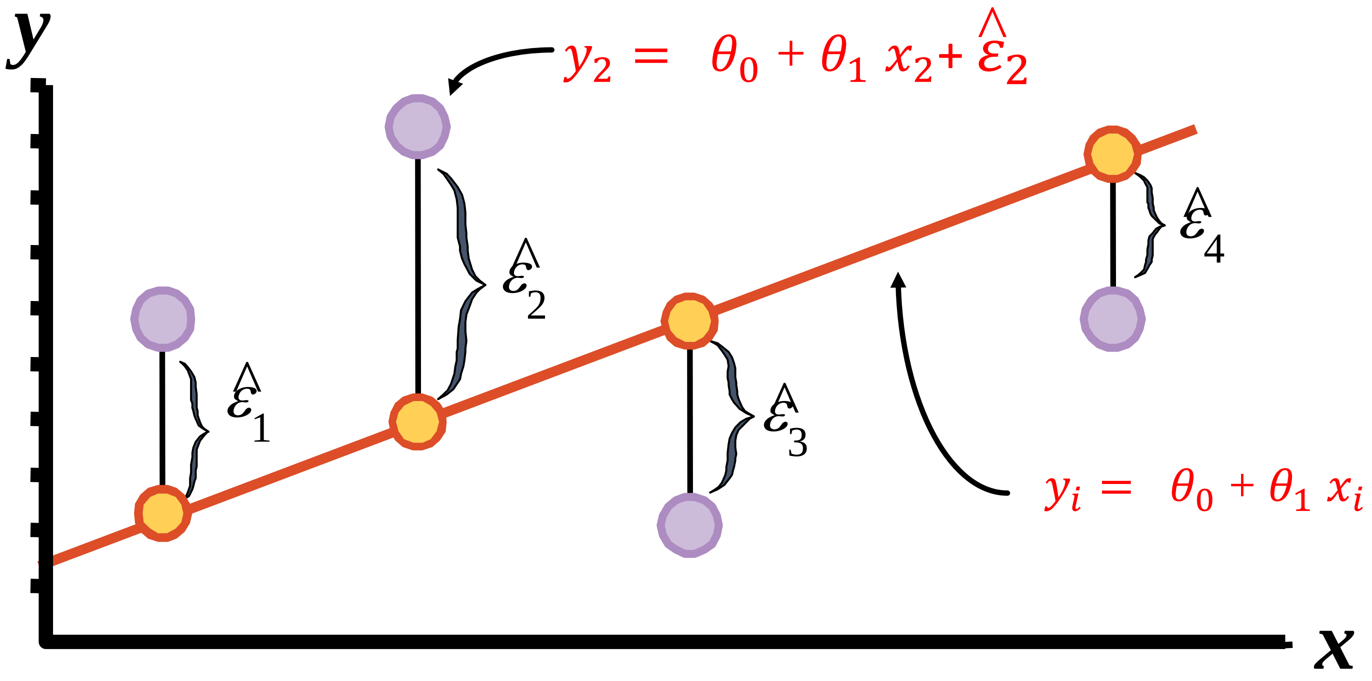
- Compute the inverse of the matrix $(X^T.X)^{-1}$ and $X^T.y$ (dot) matrix vector multiplication.

Computational complexity

- In least squares regression, Assume N training examples and C features. It consumes
  - $O(C^2N)$ to multiply X^T by X
  - $O(CN)$ to multiply X^T by Y
  - $O(N^3$ i.e., C=N) to compute the LU (or Cholesky) factorization of X^TX and use that to compute the product $(X^T.X)^{-1}.(X^T.y)$
- The computational complexity with the normal equation about $O(n^{2.4})$ to $O(n^3)$.
- For large set (100 000) of features, normal equation computation is slow.
- However linear equation handles large training sets well. So, model can fit into memory.
- Once the model is trained the new predictions will be fast.

Least Squares Graphically

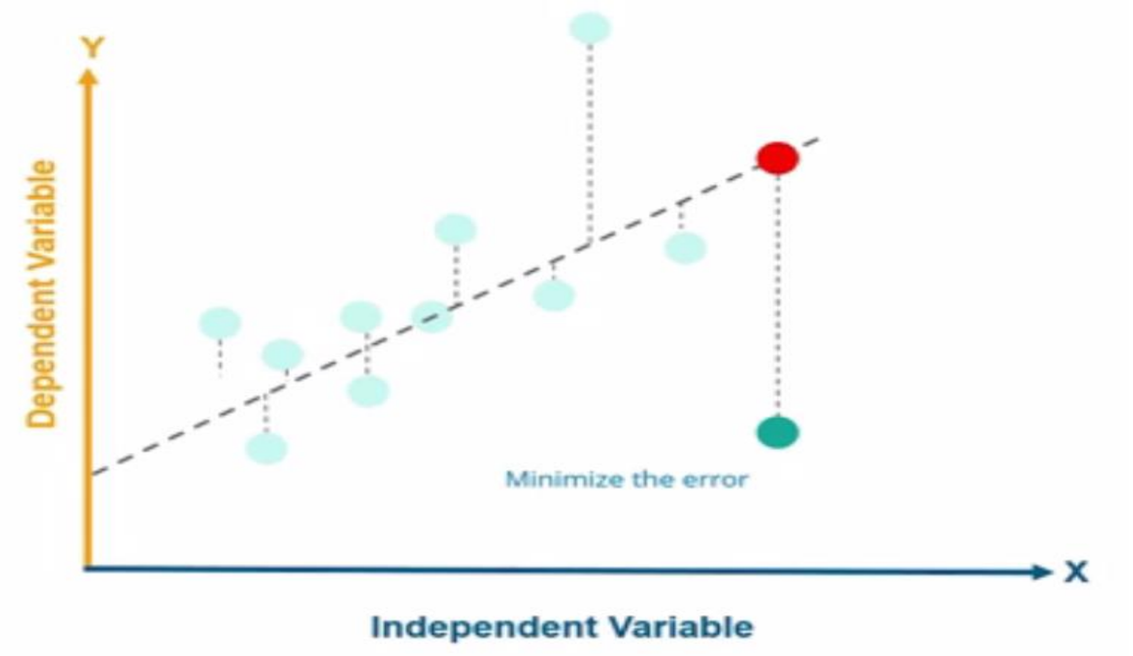
$$\text{LS minimizes} \sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \hat{\varepsilon}_1^2 + \hat{\varepsilon}_2^2 + \hat{\varepsilon}_3^2 + \hat{\varepsilon}_4^2$$



Minimize the Random Error using Mean squared Error (MSE)

- Reduce the distance between estimated (predicted) and actual (target) value
- Find the best fit of the line using **Mean squared Error (MSE) method.**

Mean Squared Error to Minimize the Error



- **Best fit'** means difference between actual *y* values and predicted *y* values are a **minimum**
  - *But* positive differences off-set negative

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_1} - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left((h_\theta(x_i) - y_i)\right)^2$$

- $y' = h_\theta(x) = \theta^T . X$

Multi-variate Linear Regression

- **Multi-variate Linear Regression**
- Linear equation representation with multiple features

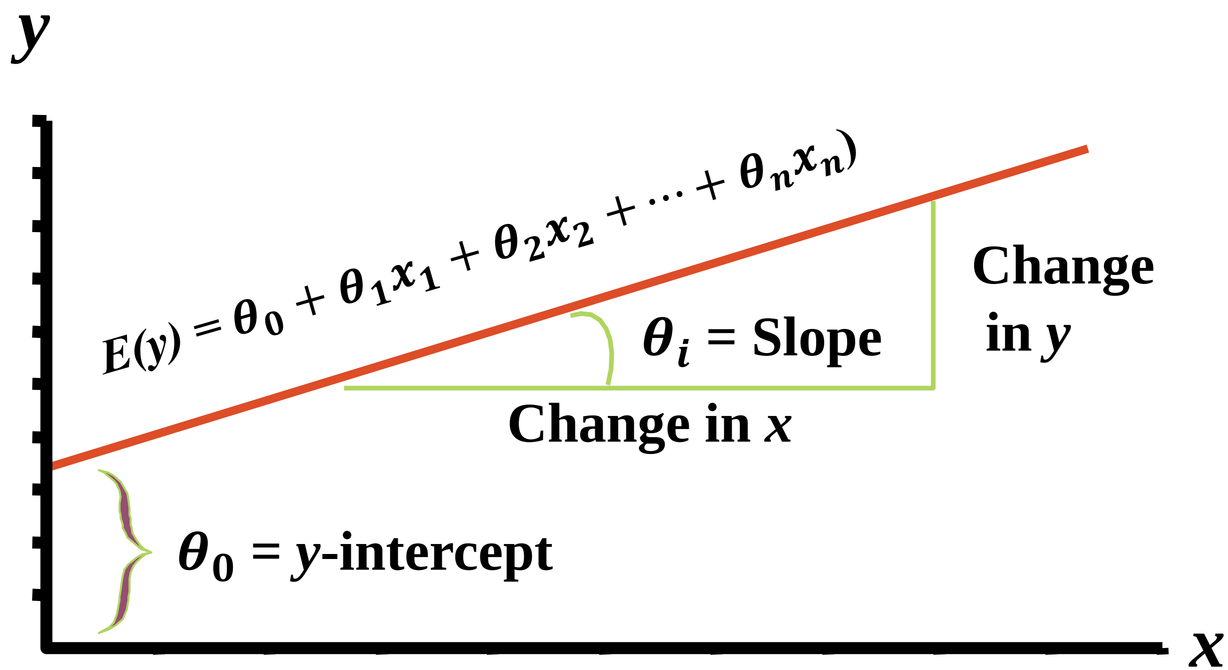
$$y = \theta_0 \, x_0 + \theta_1 \, x_1 + \theta_2 \, x_2 + \ldots\ldots + \theta_n \, x_n$$

- It is reduced into

$$y' = h_\theta(x) = \theta^T . X$$

$$Cost\ Function\ J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\theta^T x^{(i)} - y^i)^2$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\left(\sum_{j=0}^{n} \theta_j x_j^{(i)} \right) - y^{(i)} \right)^2$$

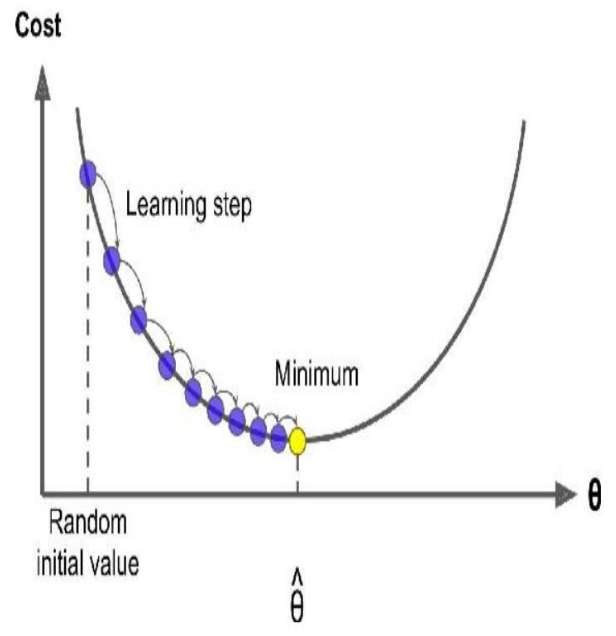


Gradient Descent

- Gradient descent is an optimization algorithm tweak the parameters iteratively in order to find the optimal cost function.

Idea of Gradient Descent

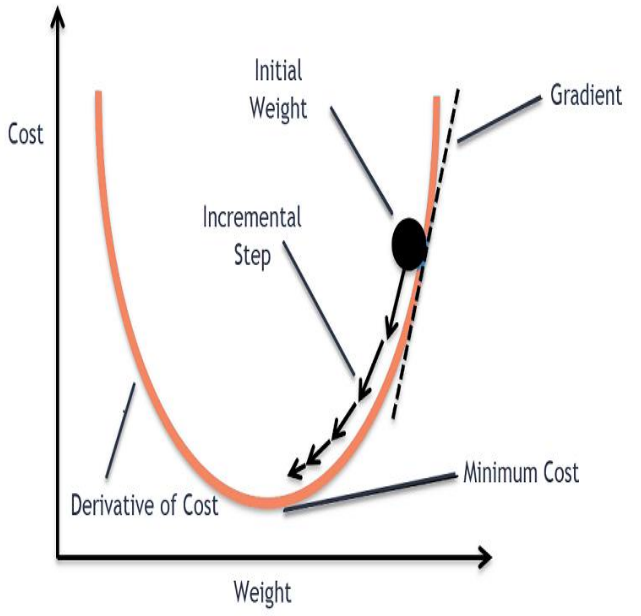
- Let consider you are on the mountain peak in the dense fog, and you want to go down. You can only feel the slope of the ground below your feet. A good strategy to reach the bottom of the valley quickly is to go downhill in the direction of the steepest slope.
- Gradient descent does the above context. It measures the local gradient of the error function with regards to the parameter vector θ, and it goes in the direction of descending gradient. Once the gradient is close to zero or zero, you have reached a **minimum.**
- Initialize θ as random values and then improve one step at a time to decrease cost function.



- The size of the steps determined by the learning rate parameter. If choose learning rate is too small, the algorithm takes many iterations to achieve the minimum.
- Convergence theorem for gradient descent is $\theta_j = \boldsymbol{\theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta_0, \theta_1)}$; $j = 0 \; and \; j = 1$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\widehat{y_1} - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left((h_\theta(x_i) - y_i) \right)^2$$

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



Learning Parameter in Gradient Descent

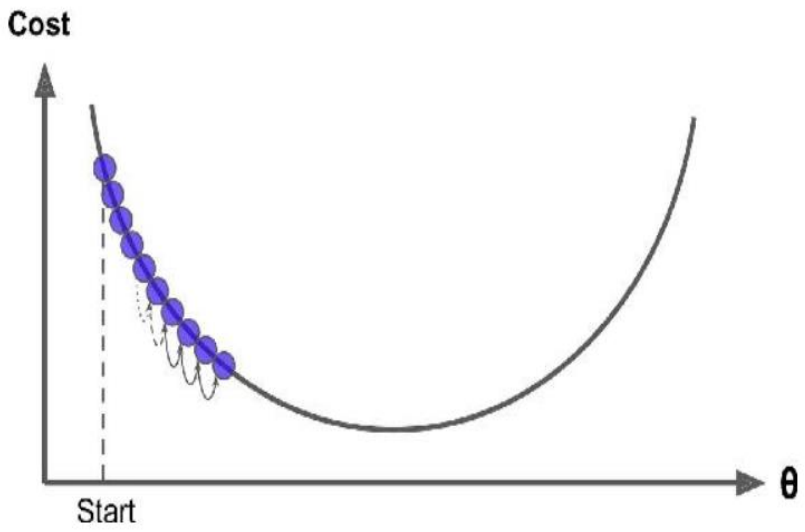
- The size of the steps determined by the learning rate parameter.

For Small Learning rate:

- If choose learning rate is too small, the algorithm takes many iterations to achieve the minimum.

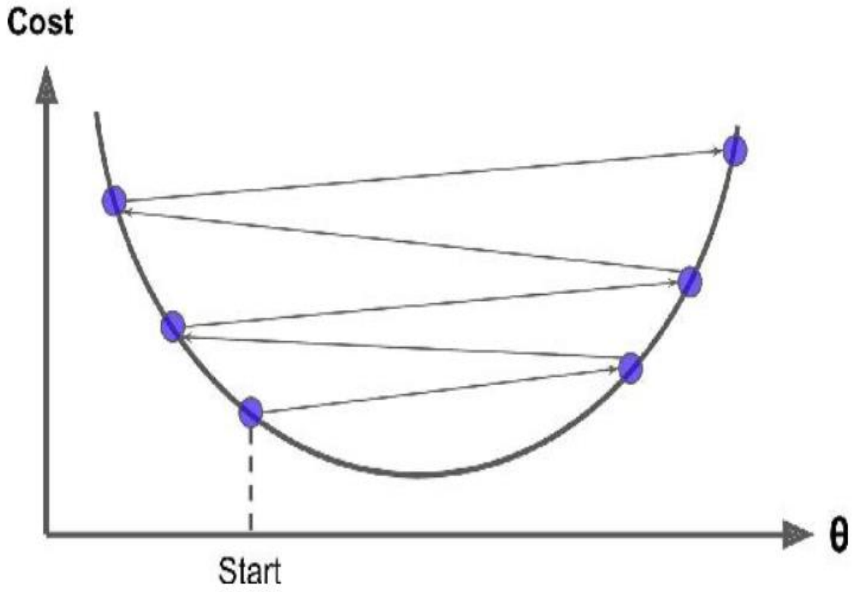
$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$



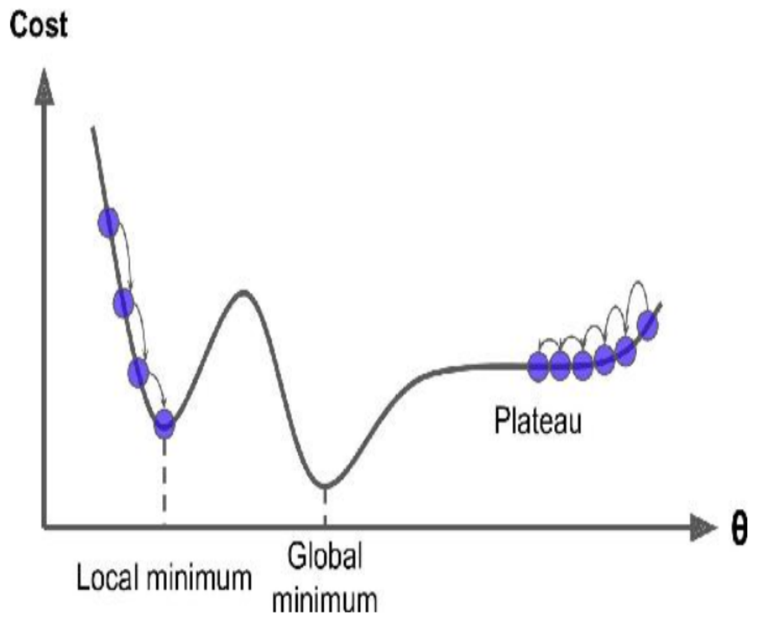
For High Learning rate:

- If choose learning rate is too high, the algorithm may jump across the valley in finding solution even higher than before.



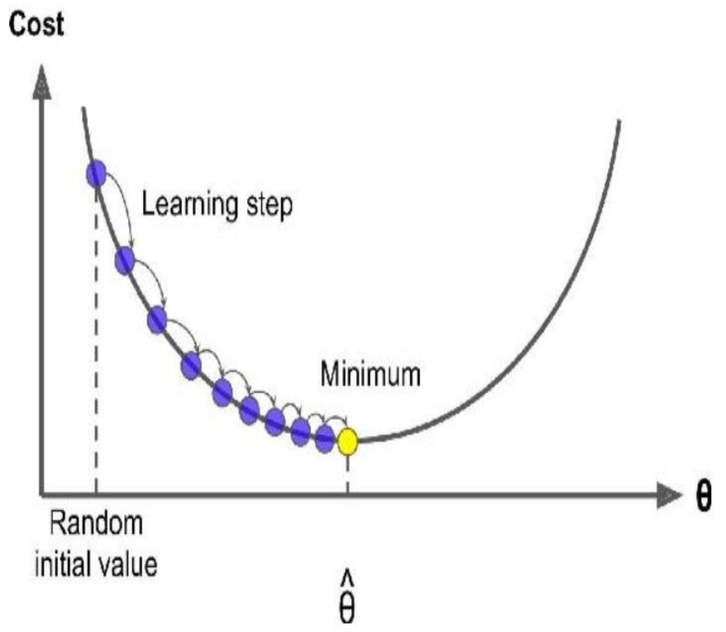
Learning Parameter in Gradient Descent

- However, all functions does not contain regular shape.
- Few functions may contain local minimums which are not as good as the global minimum.



MSE cost function in Linear Regression

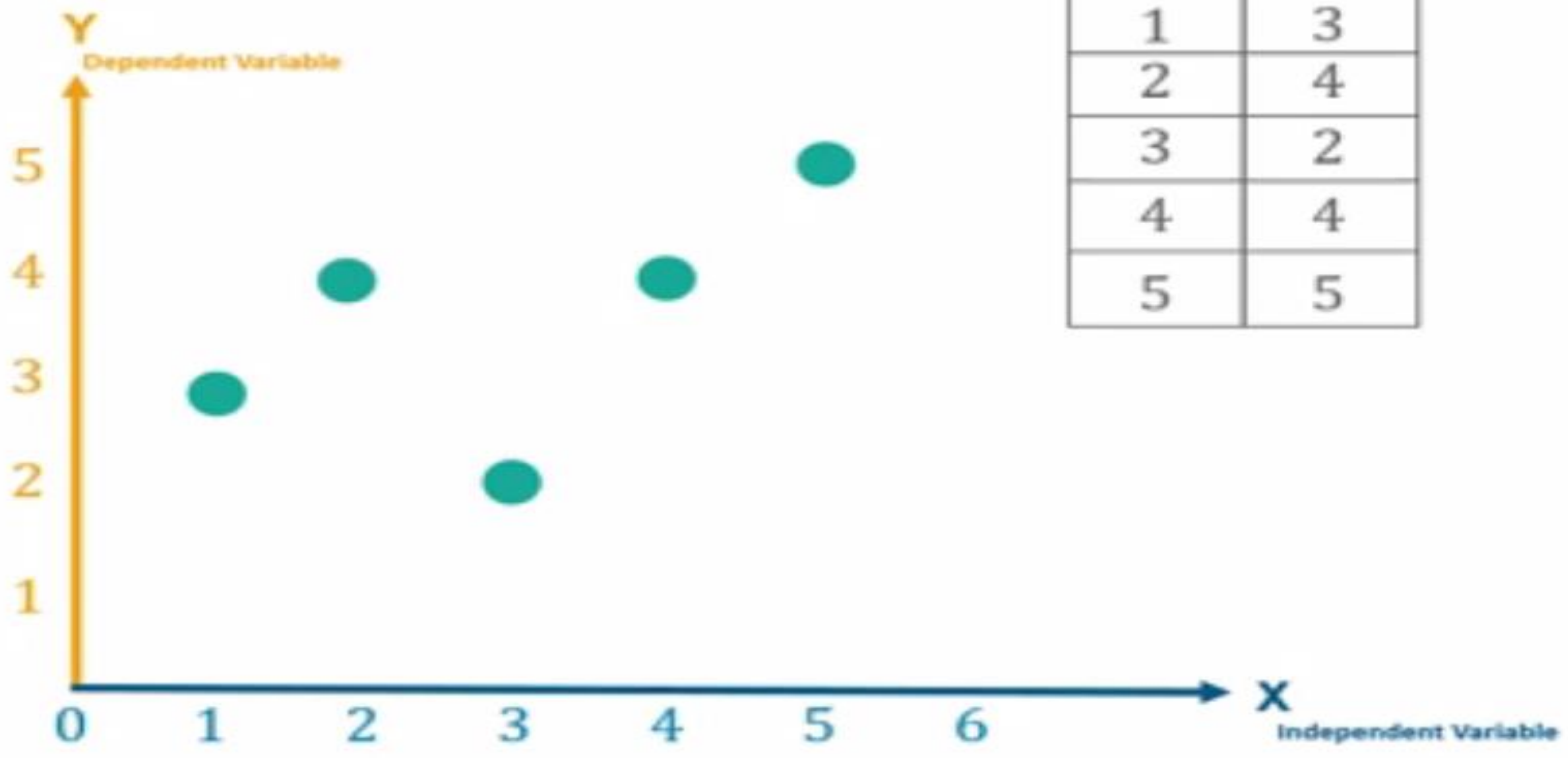
$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(\widehat{y_1} - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left((h_\theta(x_i) - y_i)\right)^2$$



- But MSE cost function for a Linear Regression model contains convex function i.e., if two points selected on the curve, the line segment joining them never crosses the curve.
- This implies that there are no local minima, just one global minimum. It is also a continuous function with a slope that never changes abruptly.
- It leads that GD guarantees to approach close to the global minimum.

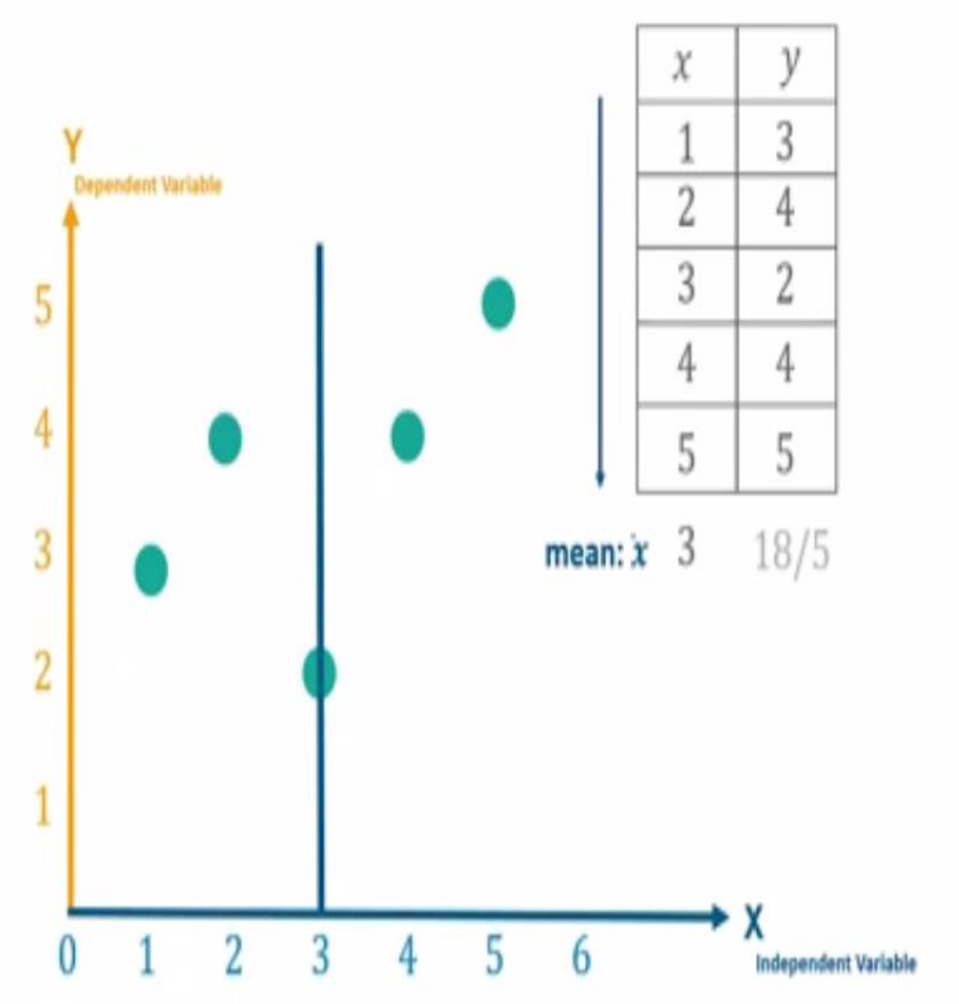
Case Study – Simple Linear Regression (i.e., Single feature x_1)

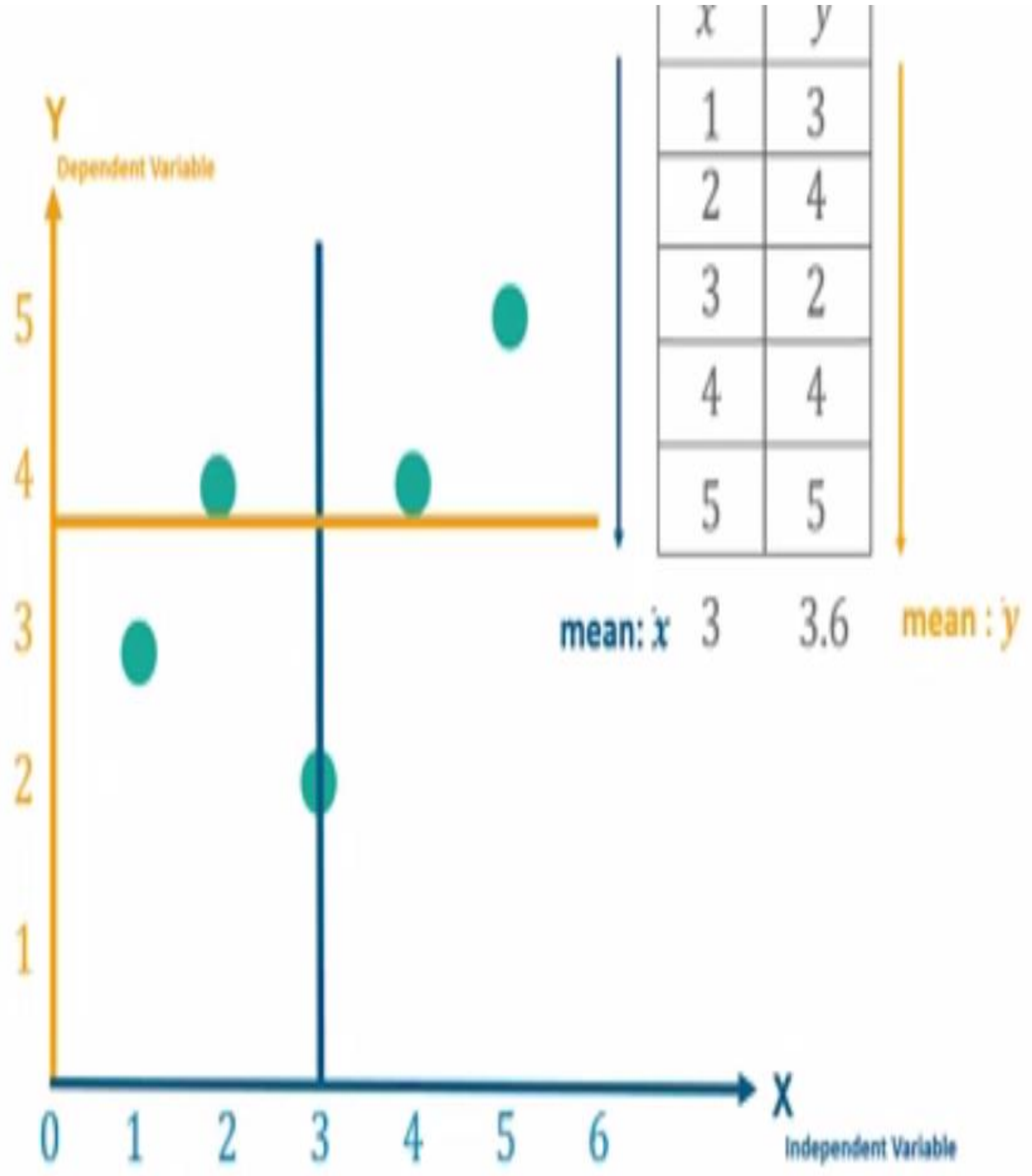
- Let consider x and y values and mark in scatter plot
- Calculate slope m for the regression line y=mx+c.
- $m = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}$
- $\bar{x}$ is mean of all x
- $\bar{y}$ is mean of all y.



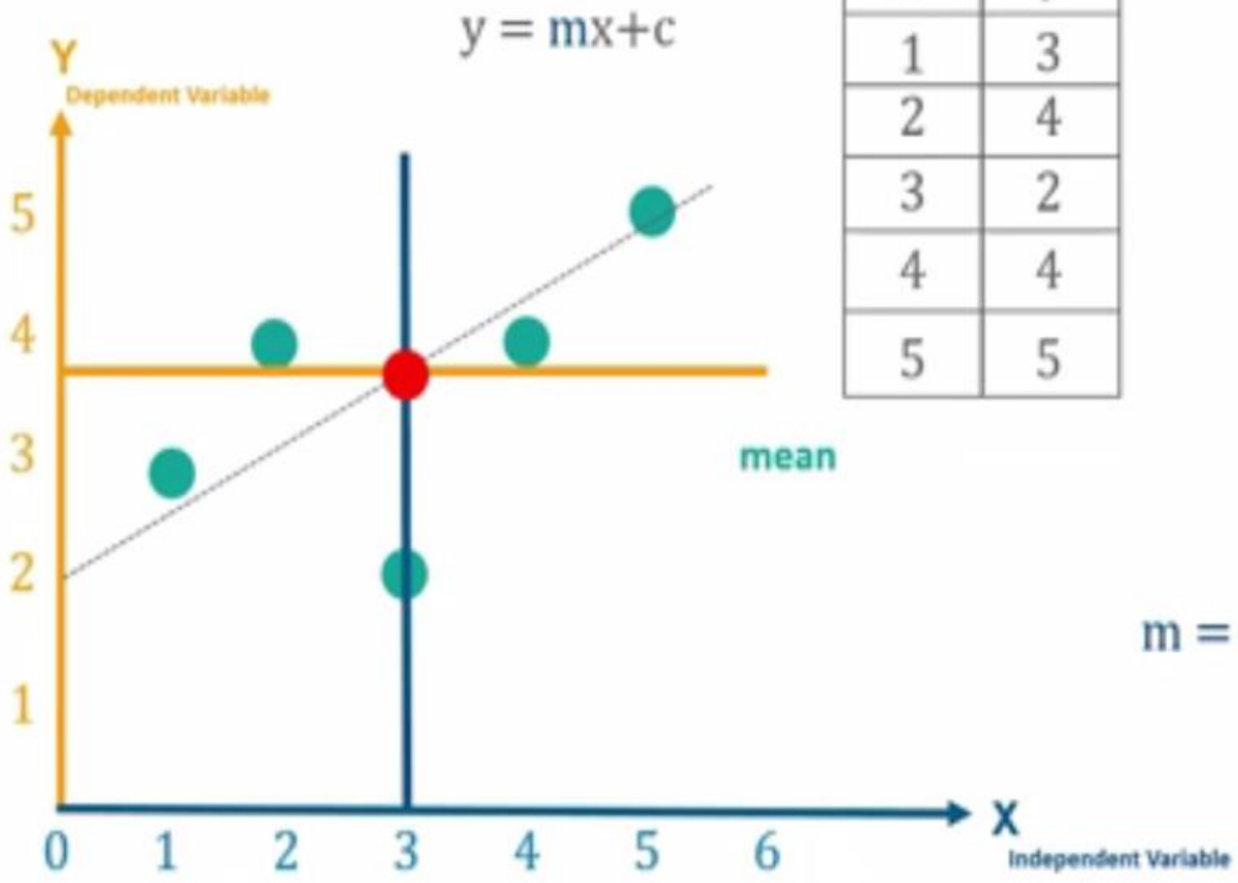
x	y
1	3
2	4
3	2
4	4
5	5

- Find mean of x, and mean of y





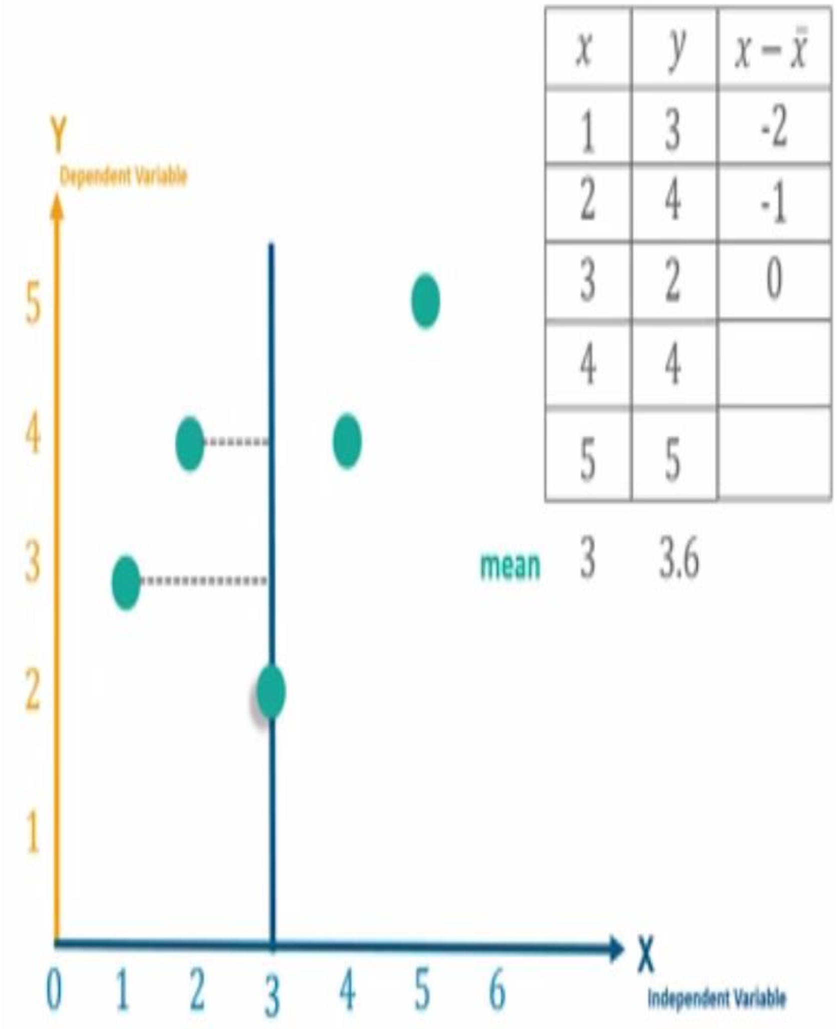
- Find the coefficients m and c in the straight line y = mx+c

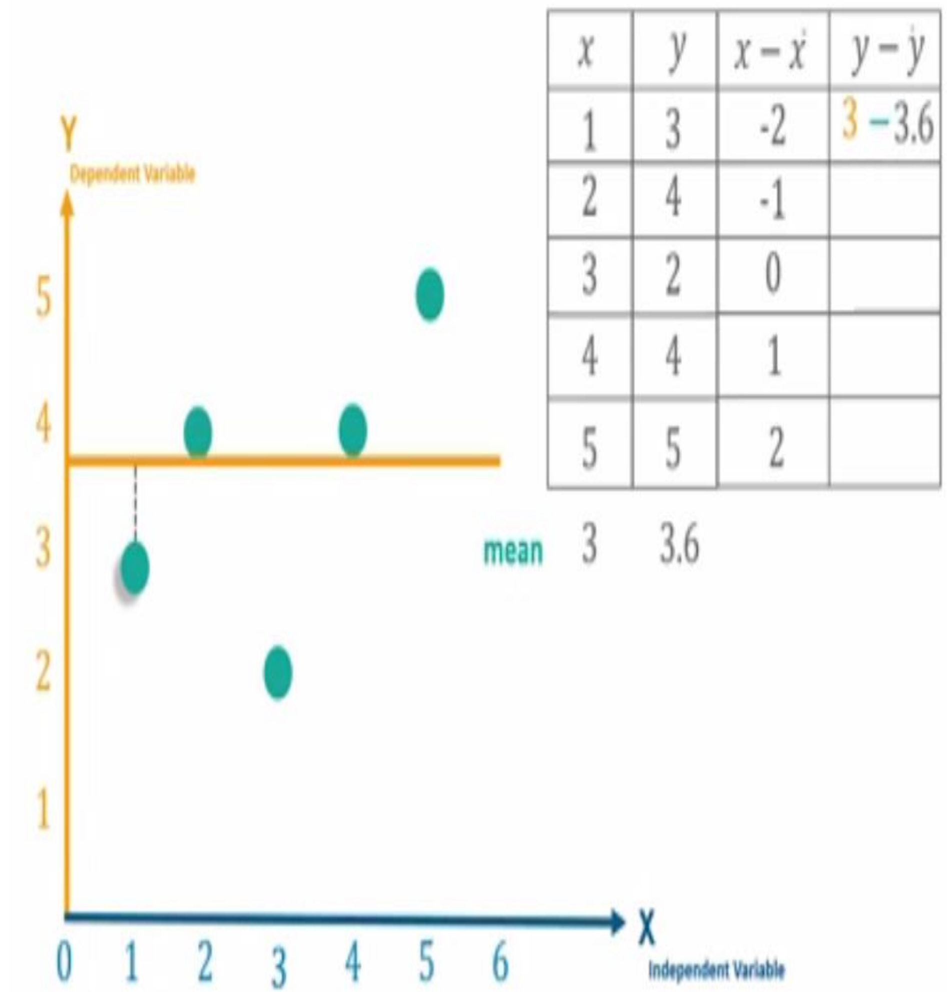


x	y
1	3
2	4
3	2
4	4
5	5

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

- Find $x-x^1$, and $y-y^1$



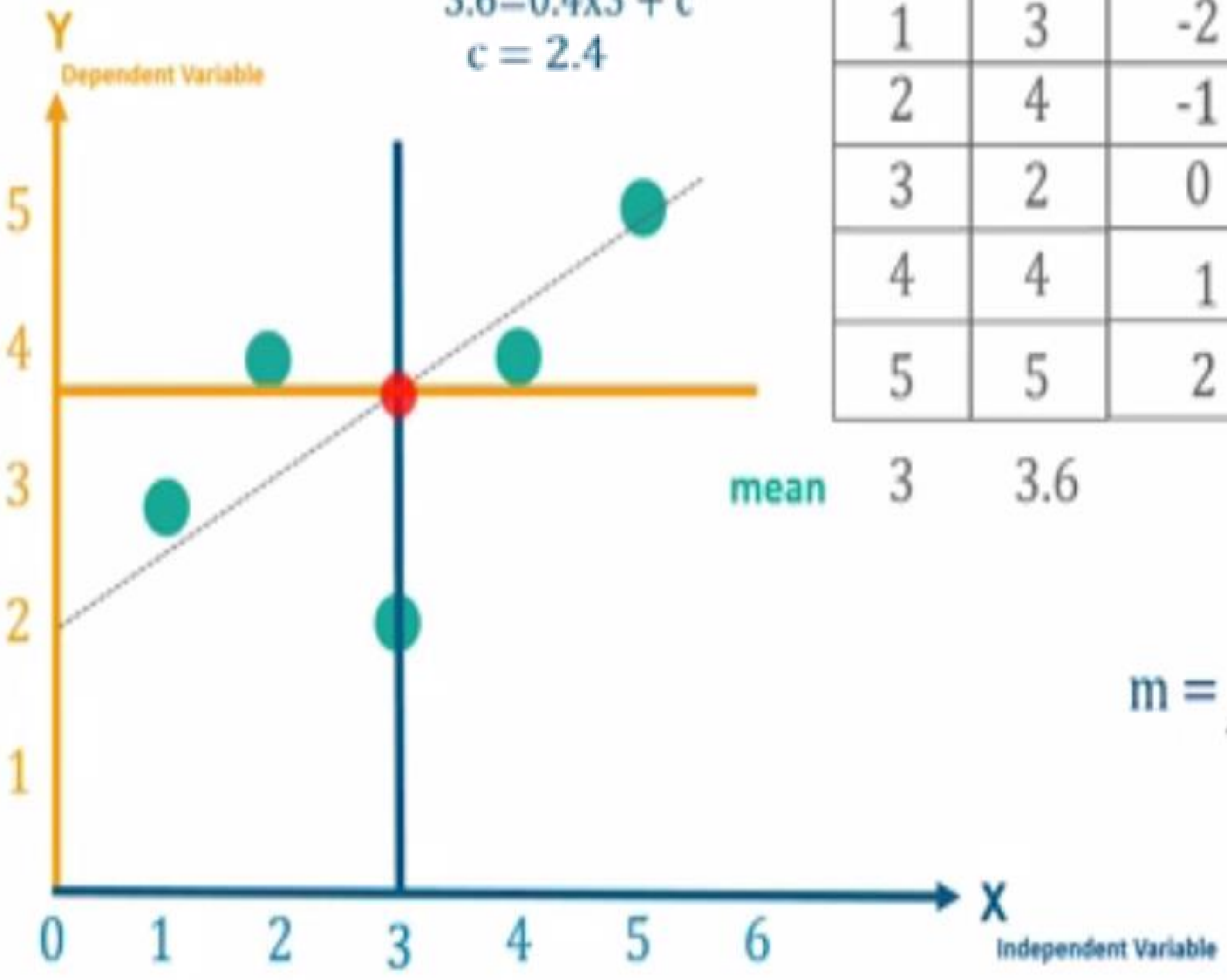


- Find slope m and y-intercept c.

$$y = mx + c$$

$$3.6 = 0.4 \times 3 + c$$

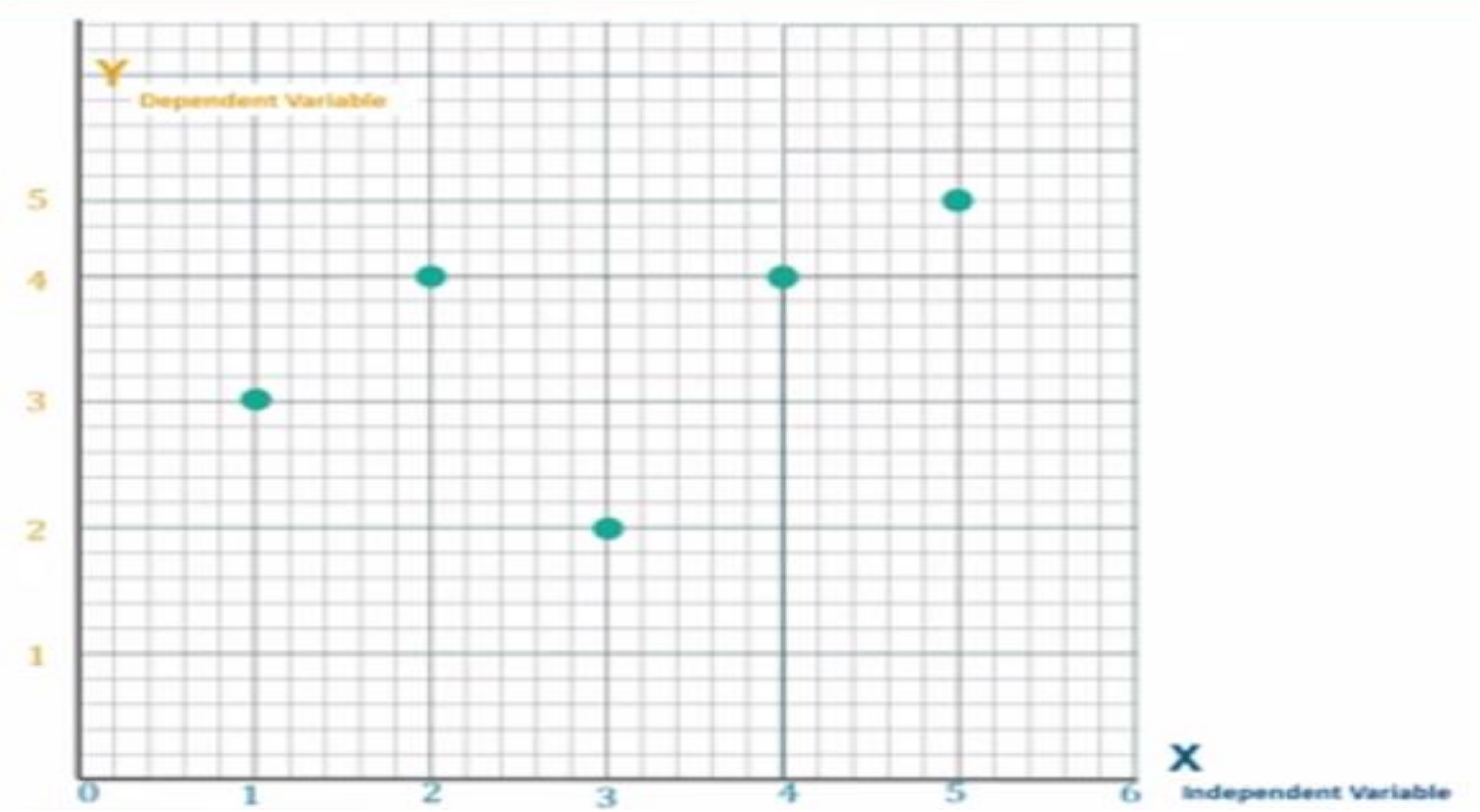
$$c = 2.4$$



x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x})(y-\bar{y})$
1	3	-2	-0.6	4	1.2
2	4	-1	0.4	1	-0.4
3	2	0	-1.6	0	0
4	4	1	0.4	1	0.4
5	5	2	1.4	4	2.8
mean 3	3.6			$\Sigma = 10$	$\Sigma = 4$

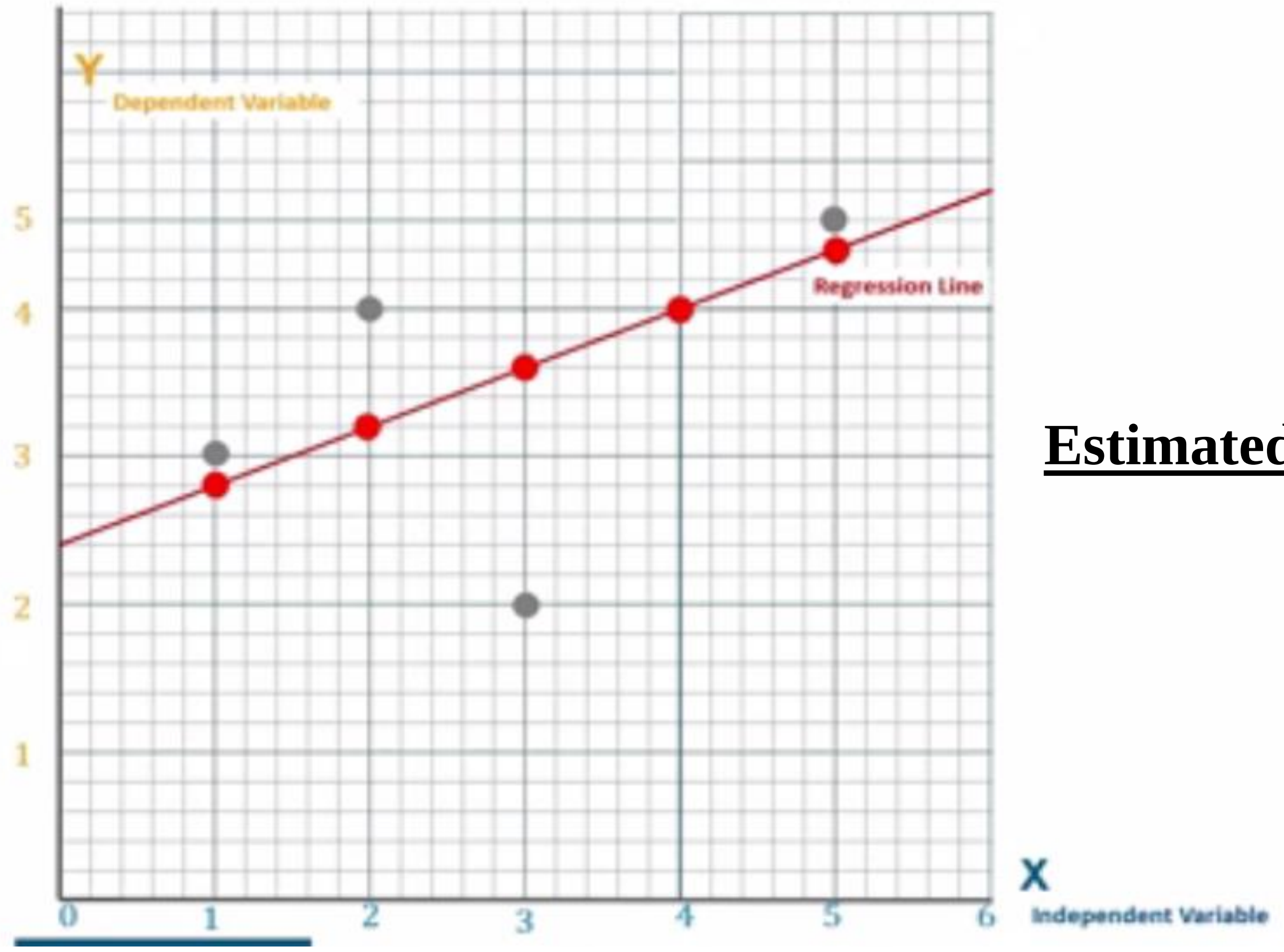
$$m = \sum \frac{(x-\bar{x})(y-\bar{y})}{(x-\bar{x})^2} = \frac{4}{10}$$

Plot the x, y values in the graph   x = {1, 2, 3, 4, 5}   y = {3, 4, 2, 4, 5}



Plot the regression line using estimated y = (mx+c) values

x = {1, 2, 3, 4, 5}   y = {2.8, 3.2, 3.6, 4, 4.4}



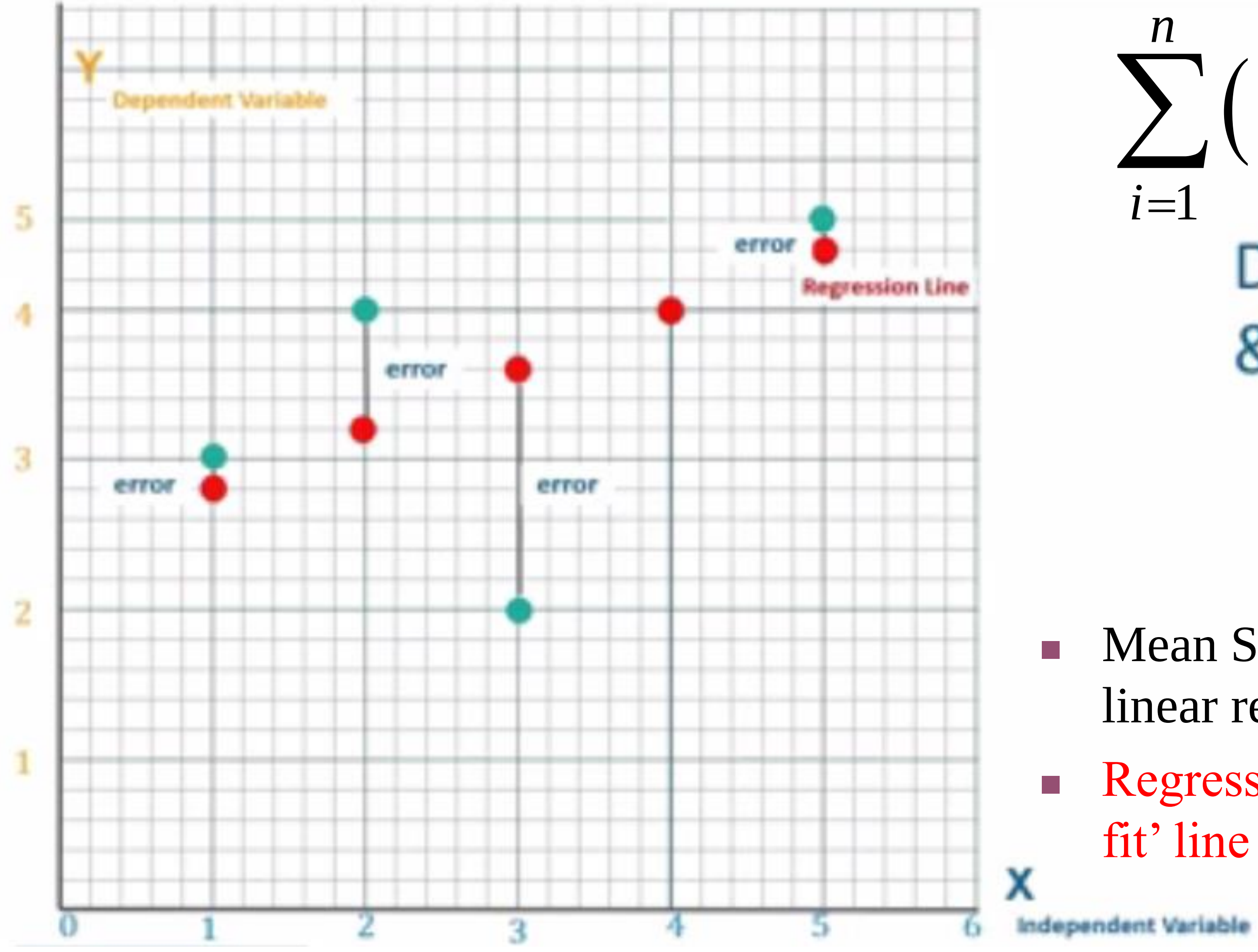
m = 0.4
c = 2.4
y = 0.4x + 2.4

For given m = 0.4 & c = 2.4, lets predict values for y for x = {1,2,3,4,5}

<u>Estimated y values</u>

y = 0.4 x 1+ 2.4 = 2.8
y = 0.4 x 2+ 2.4= 3.2
y = 0.4 x 3+ 2.4= 3.6
y = 0.4 x 4+ 2.4= 4.0
y = 0.4 x 5+ 2.4= 4.4

Find the Error $\boldsymbol{\varepsilon}$



$$\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2=\sum_{i=1}^{n}\hat{\varepsilon}_i^2$$

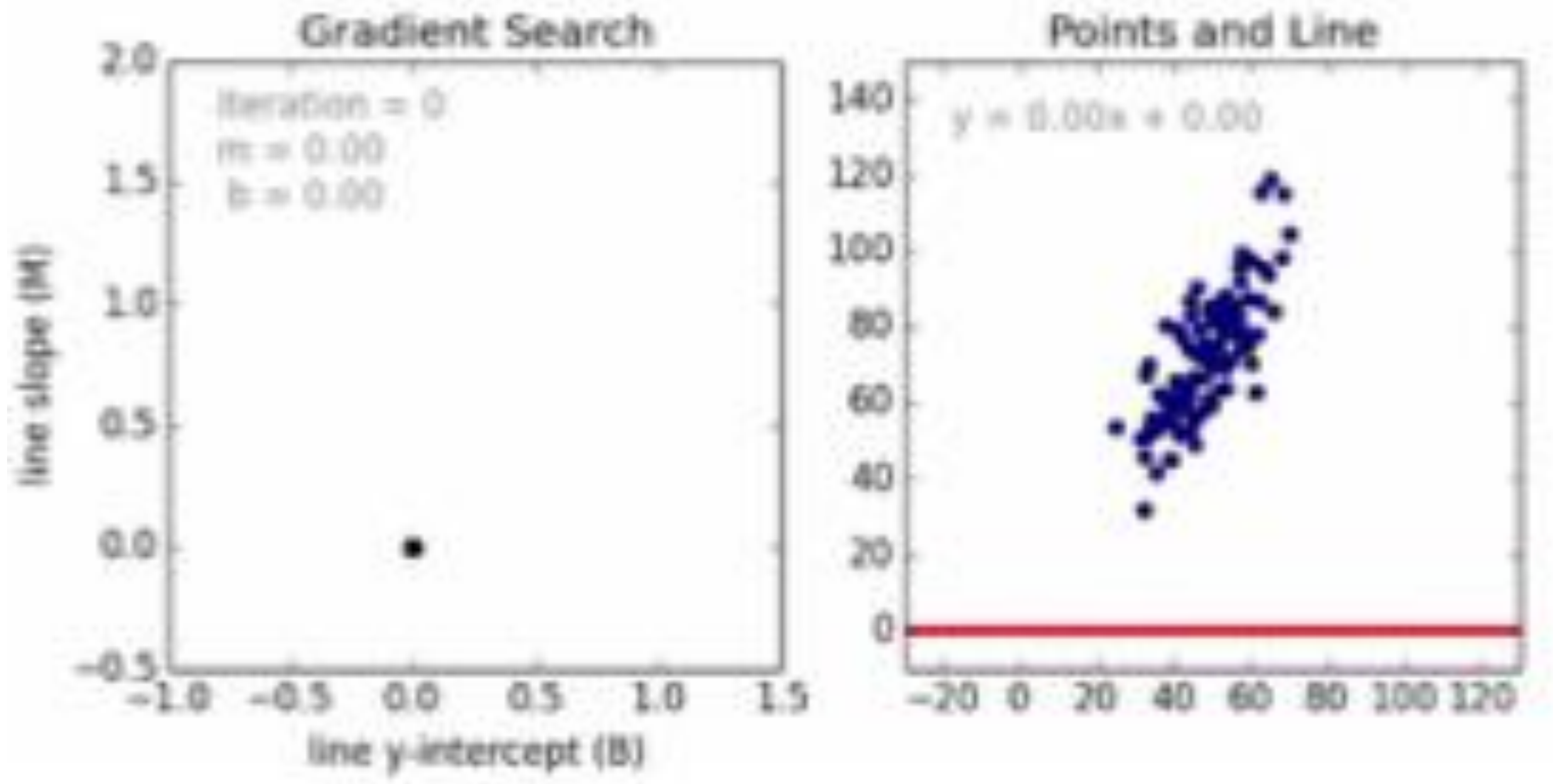
Distance between actual & predicted value

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2.$$

- Mean Square Error minimizes the error in the linear regression.
- Regression Line with least error is the ‘best fit’ line

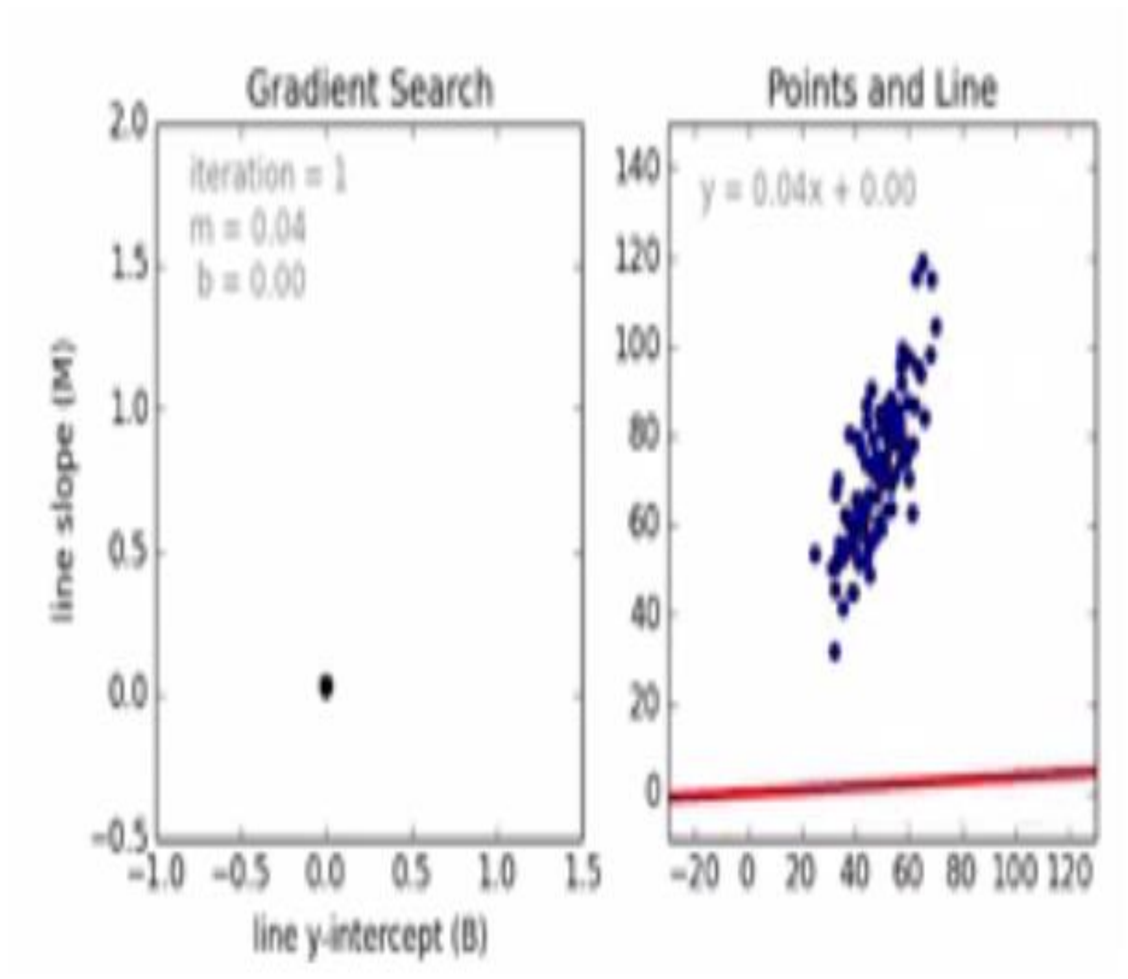
How would you draw a line through the points in real time?

- Initial values (iteration 0) for slope m = 0 and y-intercept b = 0

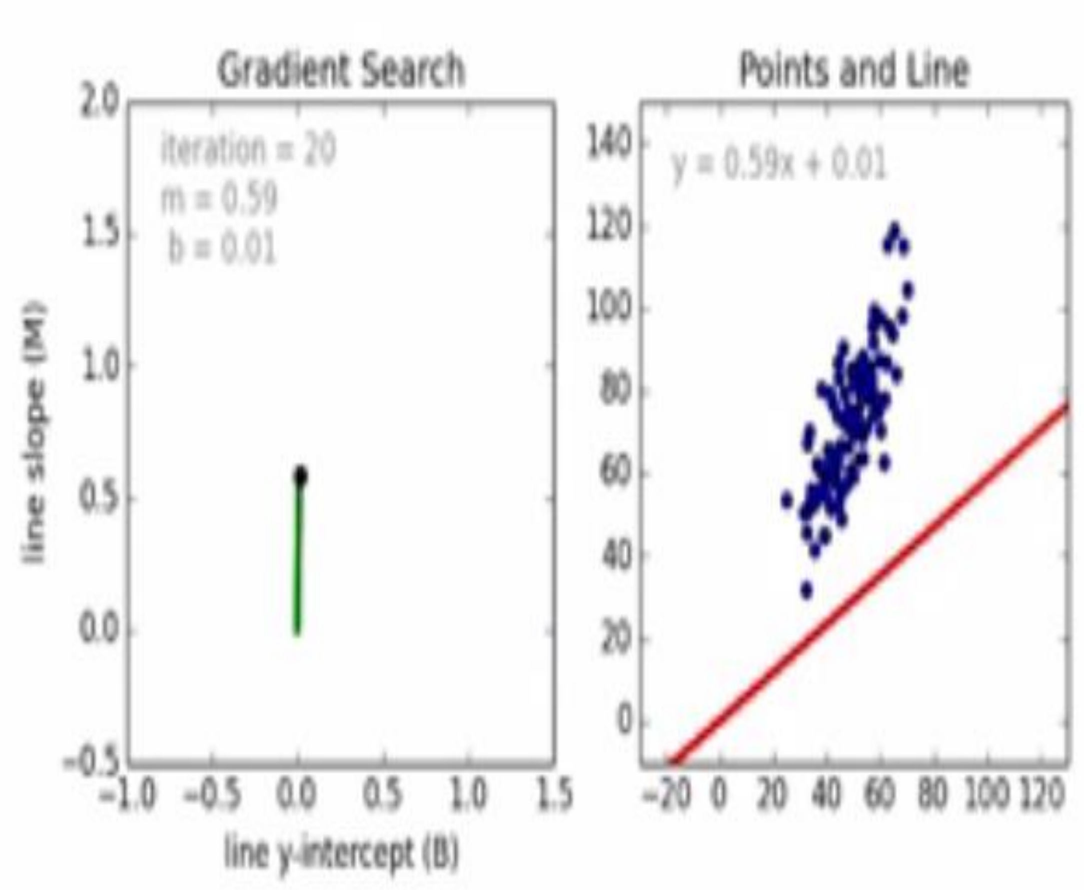


How would you draw a line through the points?

iteration 1, slope m = 0.04 and y-intercept b = 0

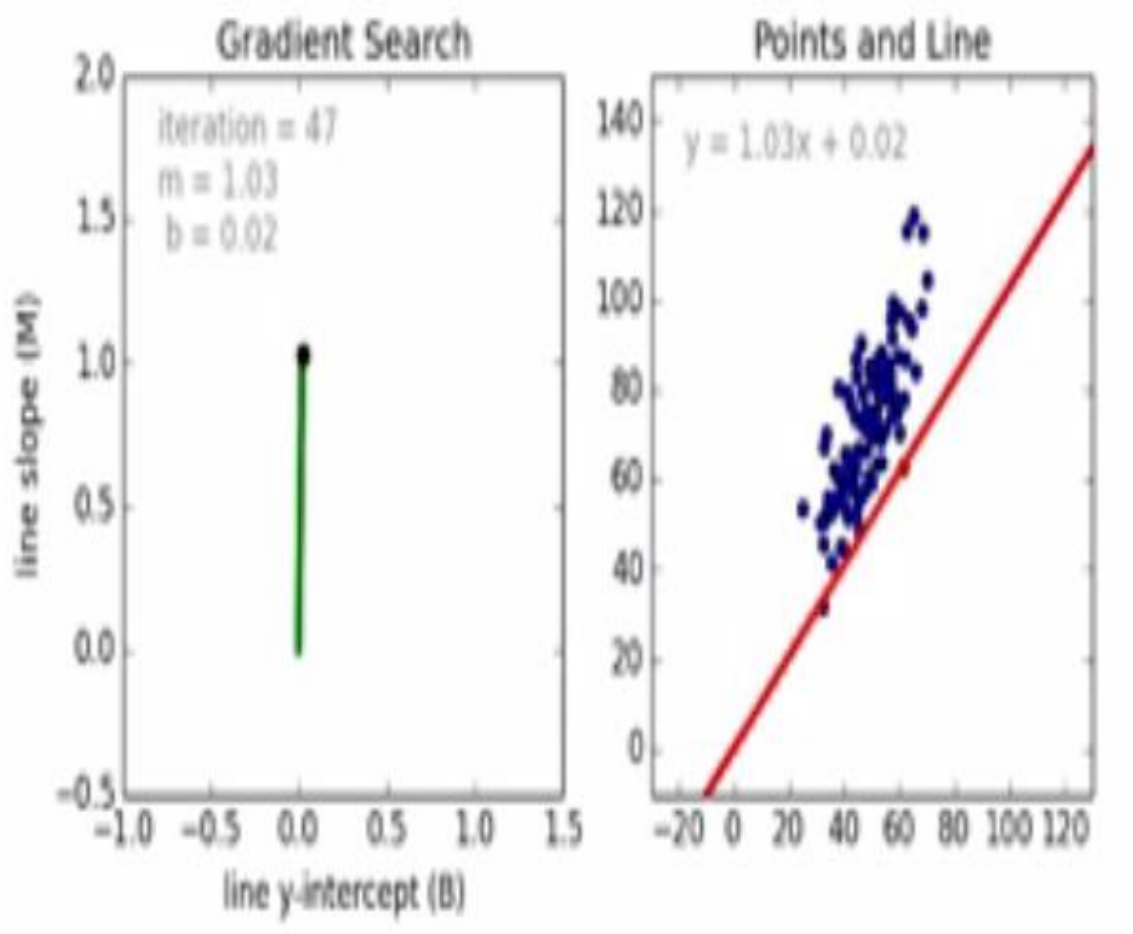


iteration 20, slope m = 0.59 and y-intercept b = 0.01

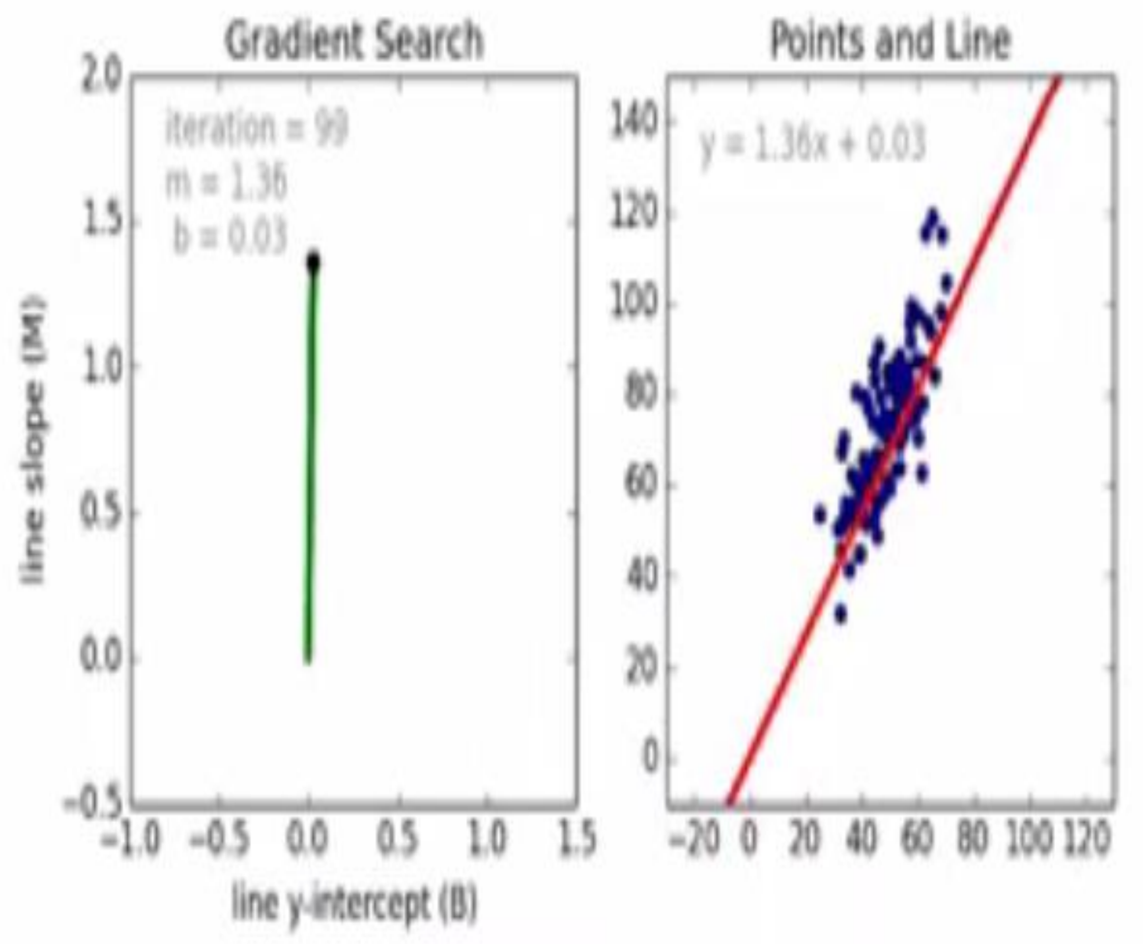


Determine which line ‘fits best’ in 100 iterations

iteration 47, slope m = 1.03 and y-intercept b = 0.02



iteration 99, slope m = 1.36 and y-intercept b = 0.03



Applications of Regression

- Forecasting an effect
- Trend forecasting and sales estimates
- Analyze the impact of price changes
- Insurance domain

VIT-AP UNIVERSITY

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.

2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.

3. Wikipedia