Supervised

unsuper.

reg

classific

clustering

ML → Linear

simple

vanilla

pointers → algo

+1 room → +50$k

Housing Prices

House 1	House 2	House 3	House 4	House 5
1 room	2 rooms	3 rooms	4 rooms	5 rooms
$150K	$200K	???	$300K	$350K

+50 +50 +50 +50

250K

one integer

sqft

locality

sf

$y \neq 0 \rightarrow x = 0$

$y = mx + c$

Housing Prices

House 1	House 2	House 3	House 4	House 5
1 room	2 rooms	3 rooms	4 rooms	5 rooms
$150K	$200K	???	$300K	$350K
		$250K		

0 1

1LR → $100K

$x_0 = 0$

$y = ?$

β_0

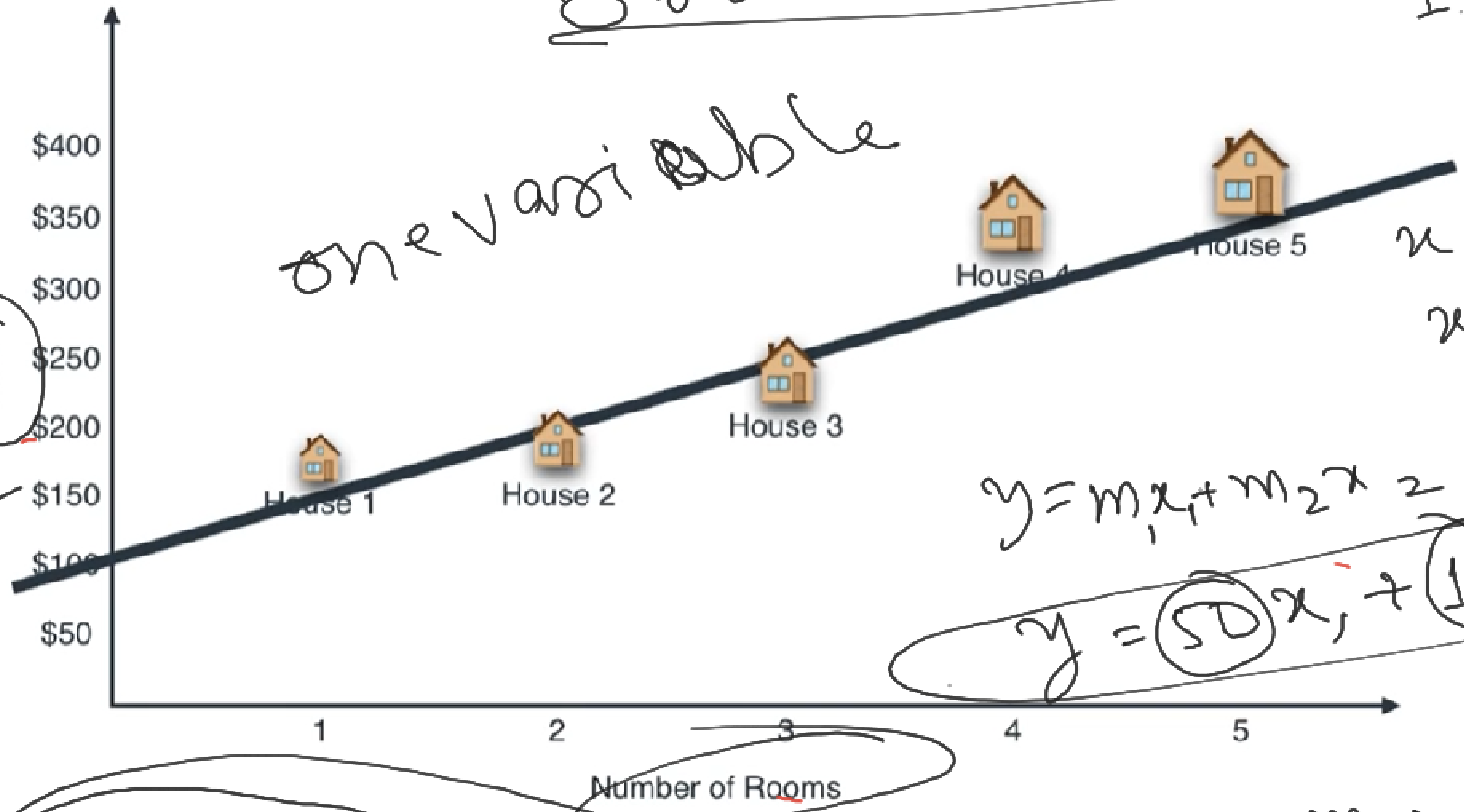
$x_i = 0$

+1 room + 50

0 room → 0

$1\,ft^2 = \$1$

$y = m x + c$



one variable

$x_1 \to$ rooms

$x_2 \to$ sqft

$50 = 50 \times 1$

$100 = 50 \times 2$

$y = m_1 x_1 + m_2 x_2$

$y = 50 x_1 + 1 \times x_2$

pred

$y = m_1 x_1 + m_2 x_2 + m_3 x_3 + c$

multi

m → slope

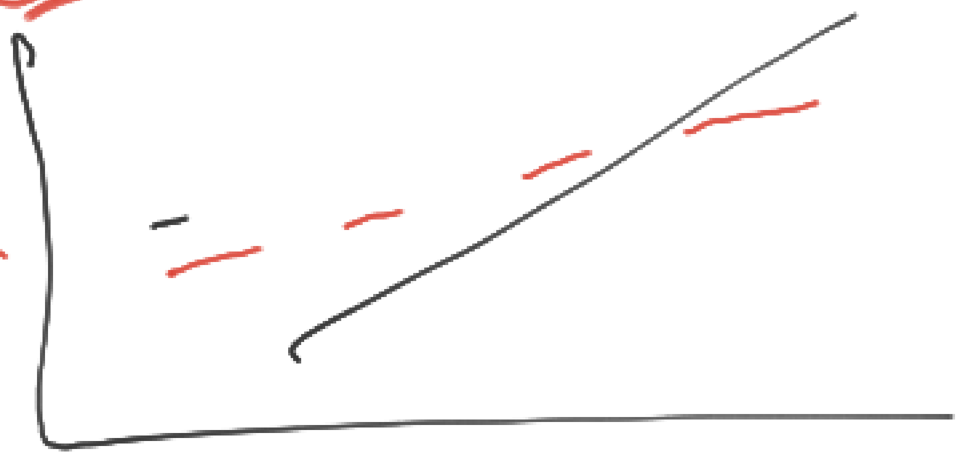
c → intercept

y = all other x_0

$c = \beta_0$

$m_p \to \beta_1 \to \beta_n$

$y = mx + c$



$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots + \epsilon$

x_1 "m" β_1 y_1

bias

NM

data → features

auto → boarding → β_1

$\overset{x_1}{\text{Km}}$ = 5 times Km

$\overset{x_2}{\text{time}}$ = 1xt

β_0 / c

fare

$y = 5x_1 + 1 \times t + \beta_1$

1BR → +50K

$y = 50x_1$

$x = 0$

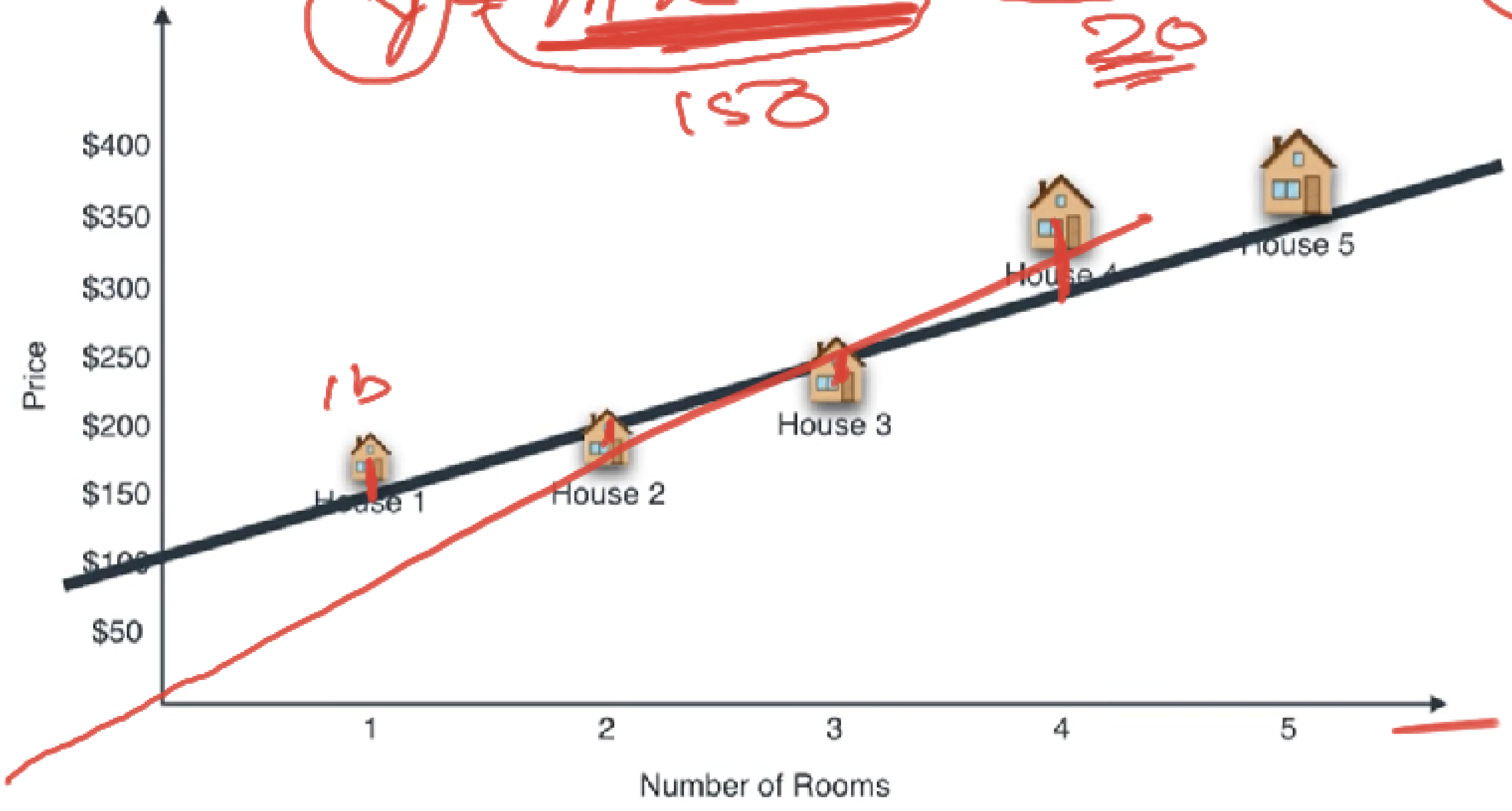
$t = 0$

$\beta_0 =$

170

$$y = mx + c + \epsilon$$

150   20



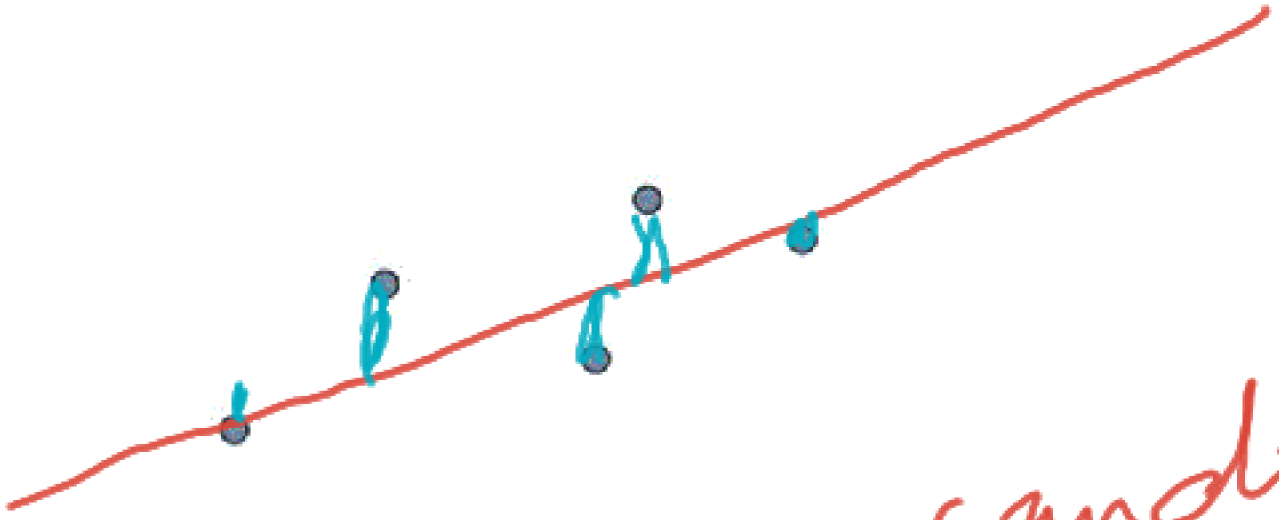
min $+\epsilon$ error

$x = 0 \quad y = 0$

$$y = \beta_0 + \beta_1 x_1$$

How does Line of best fit happen

Linear Regression



ϵ

R = 100%

99%

error

ϵ = noise

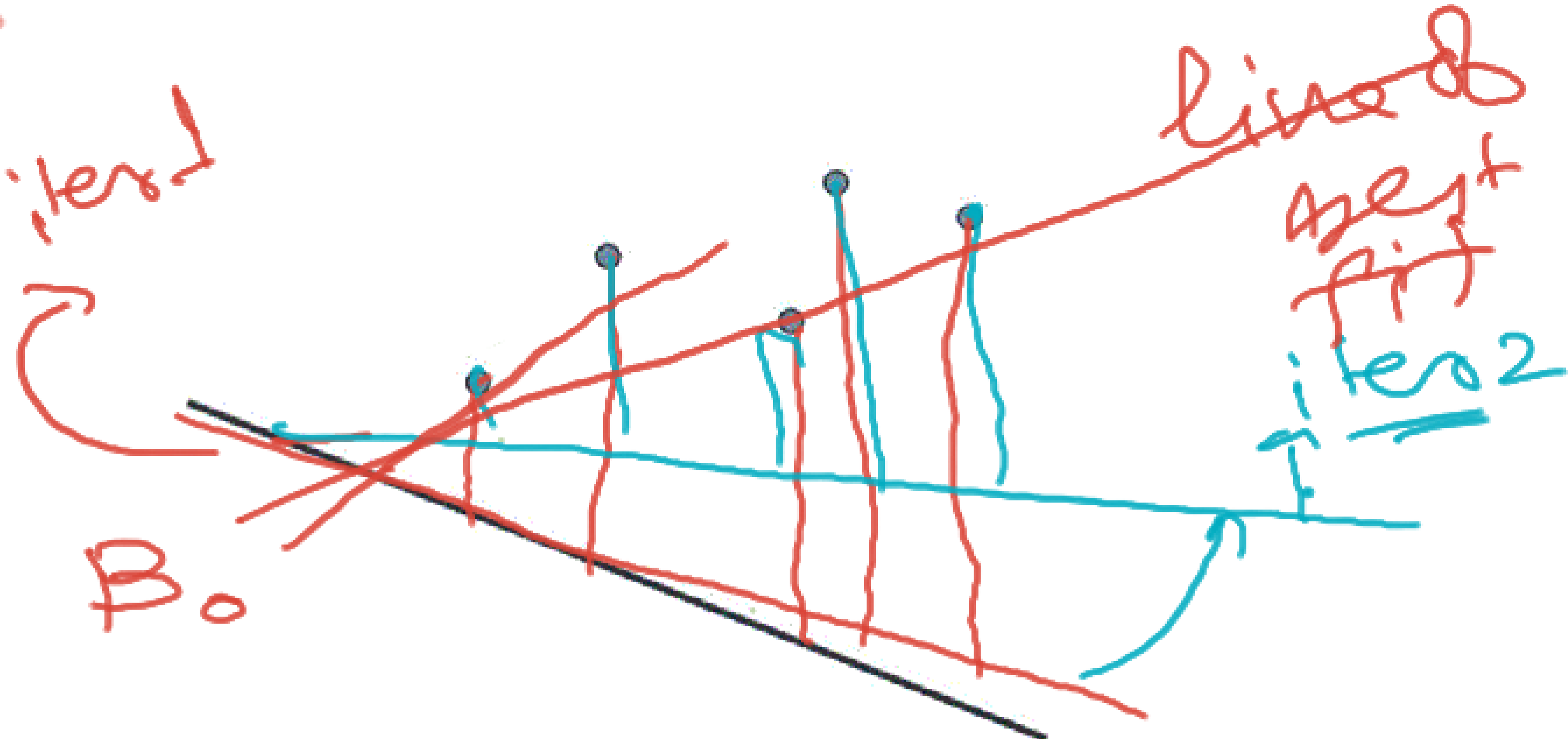
→ random

error = SSE

blue < red

Let us break down what is happening

Linear Regression



B_1, B_0

error

$(y_a - y_p) = +ve$

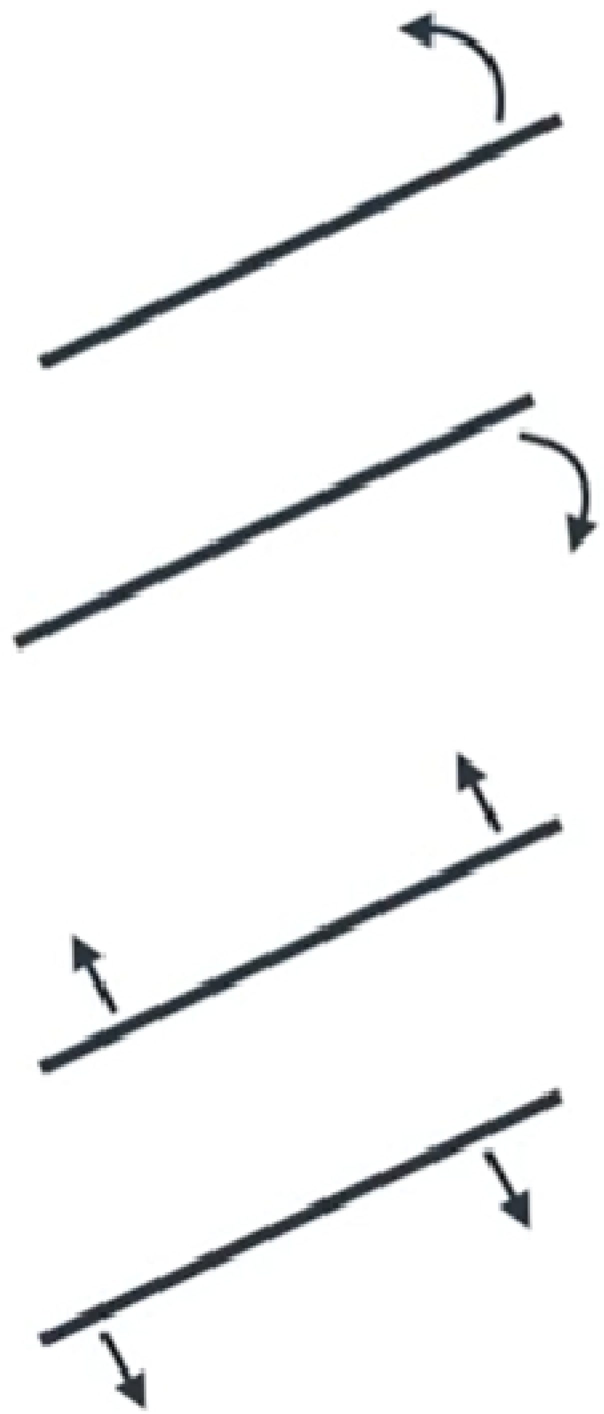
$(y_a - y_p)^2 = -ve$

(+ve)

$|y_a - y_p|$

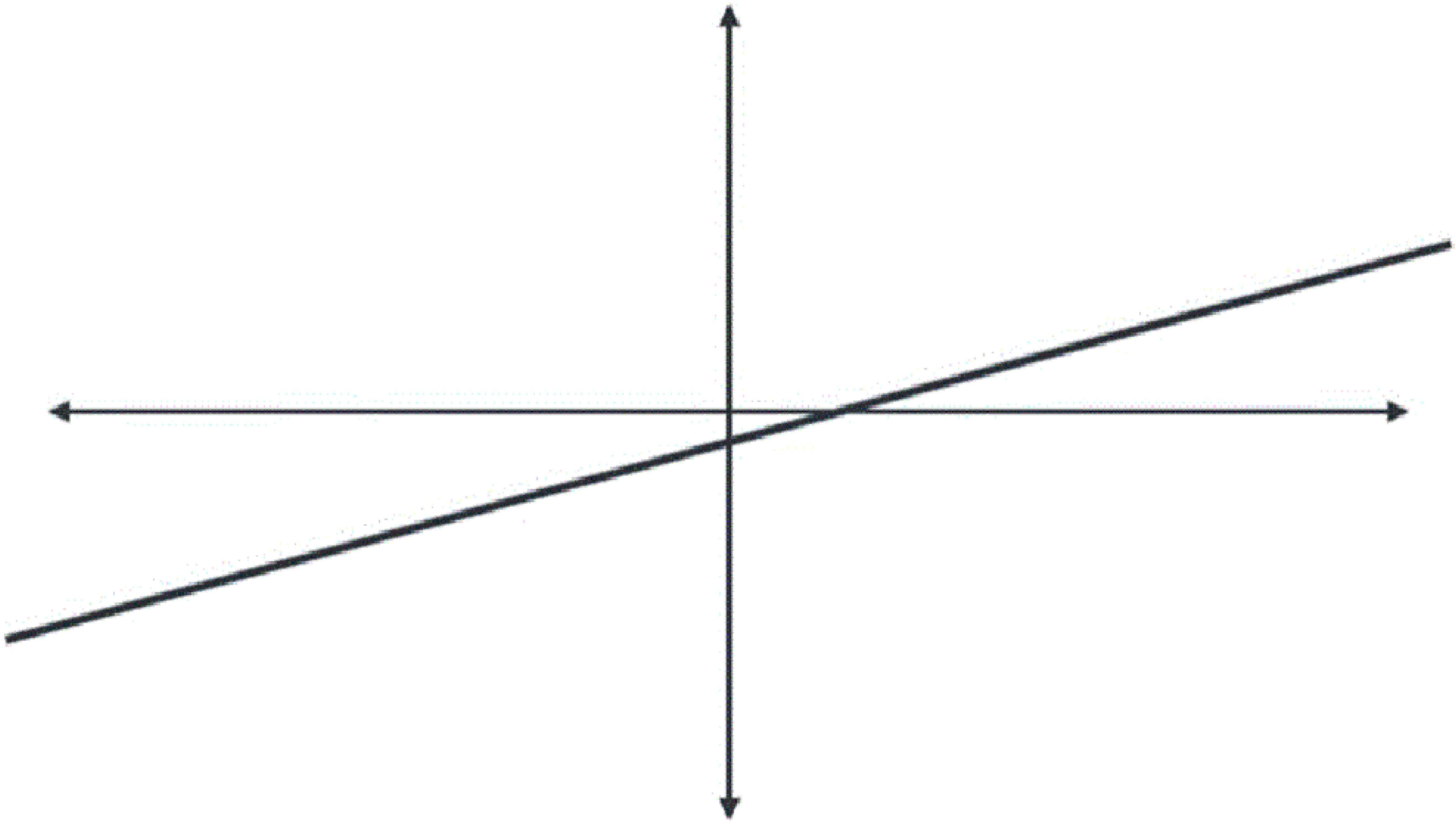
diff

How to move a line

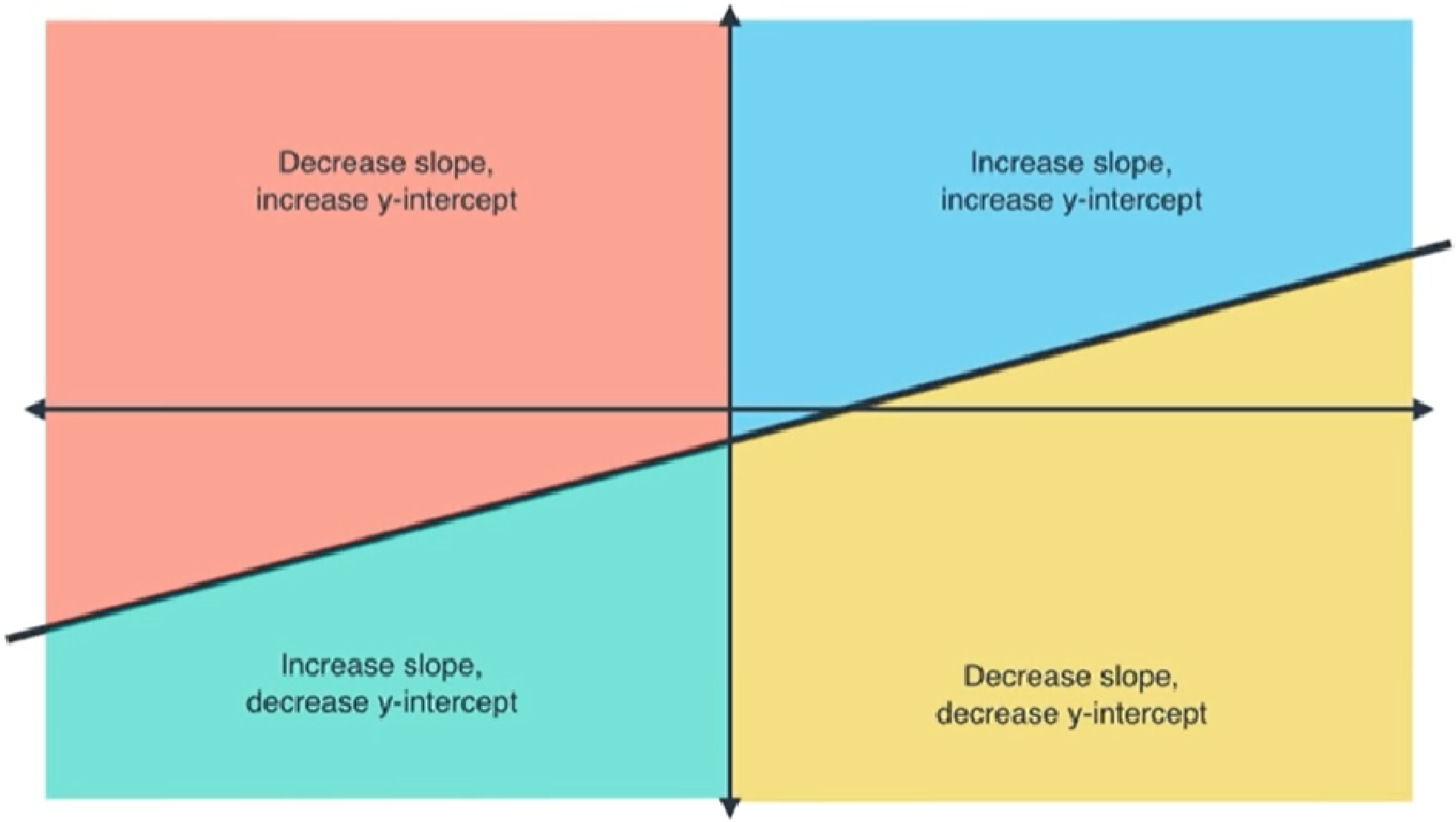


Rotate line counter-clockwise	Increase slope
Rotate line clockwise	Decrease slope
Translate line up	Increase y-intercept
Translate line down	Decrease y-intercept

Four cases



Four cases


Decrease slope,
increase y-intercept
Increase slope,
increase y-intercept
Increase slope,
decrease y-intercept
Decrease slope,
decrease y-intercept

How it happens

Video

Step 1: Start with a random line

Step 2: Pick a large number. 1000
(number of repetitions, or epochs)

Step 3: Pick a small number. 0.01
(learning rate)

Step 4: (repeat 1000 times)

-Pick random point

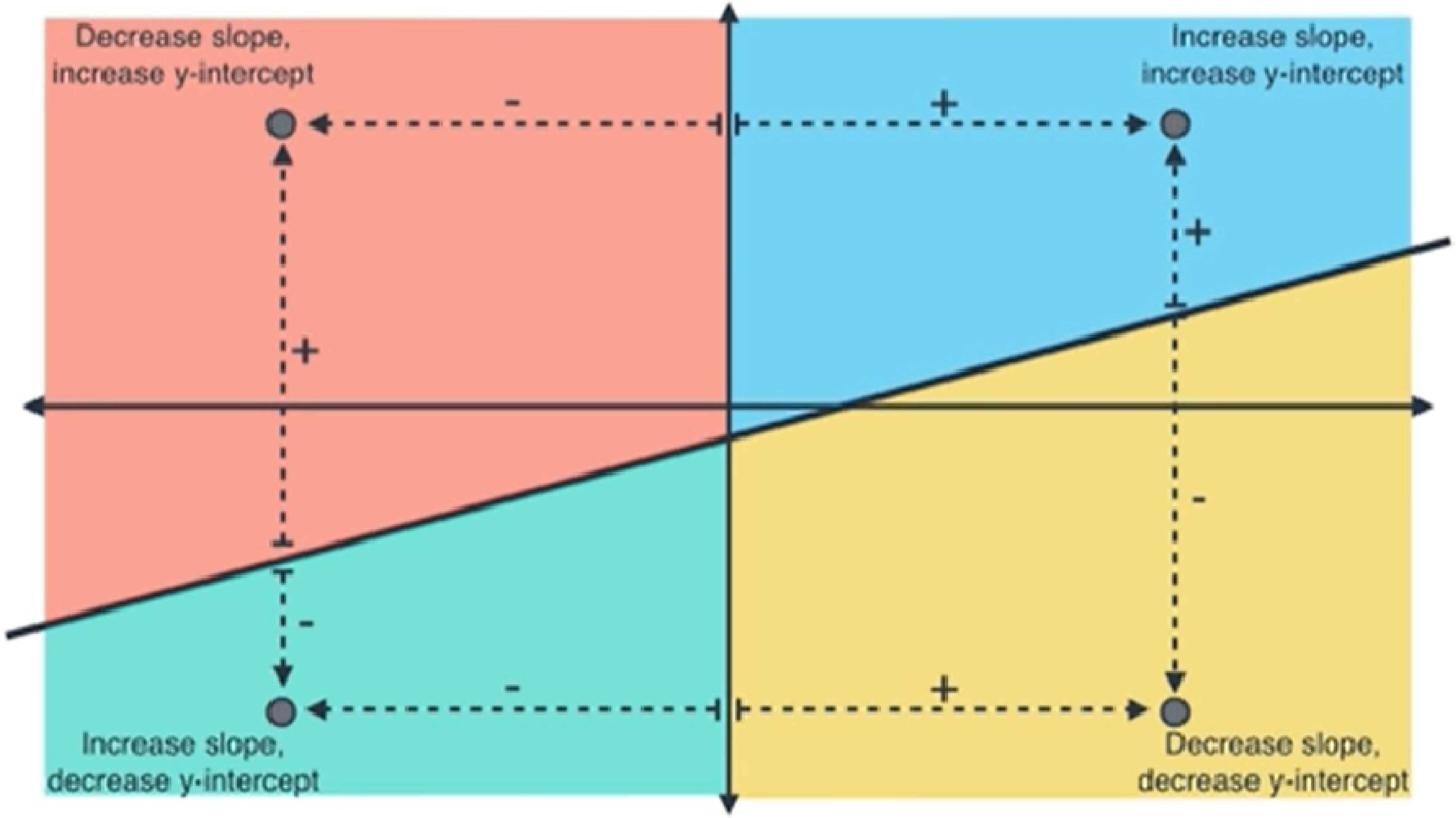
-If point **above** line, and to the **right** of the y-axis:
add 0.01 to slope
add 0.01 to y-intercept

-If point **above** line, and to the **left** of the y-axis:
subtract 0.01 to slope
add 0.01 to y-intercept

-If point **below** line, and to the **right** of the y-axis:
subtract 0.01 to slope
subtract 0.01 to y-intercept

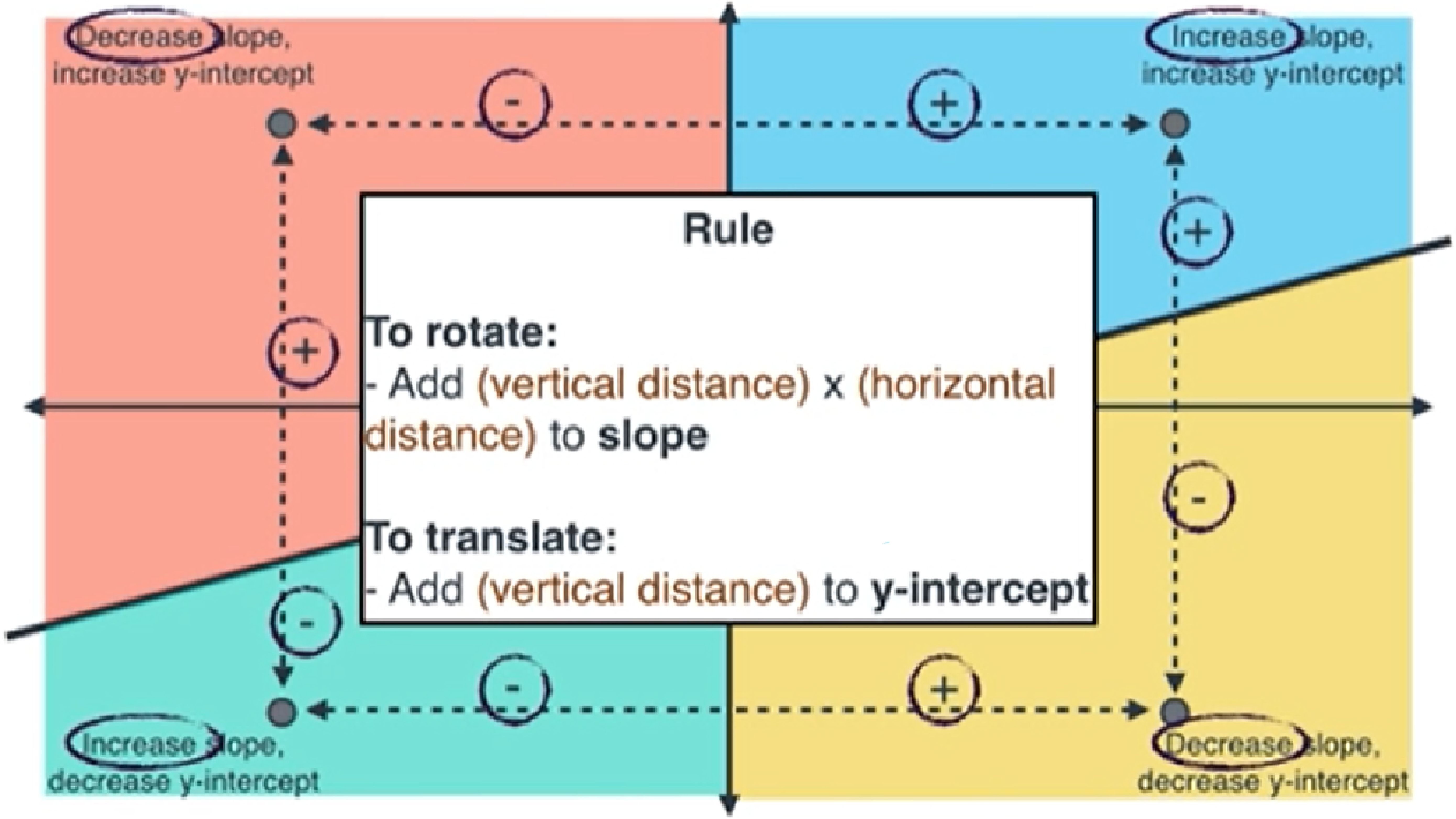
-If point **below** line, and to the **left** of the y-axis:
add 0.01 to slope
subtract 0.01 to y-intercept

Four cases



Decrease slope,
increase y-intercept
Increase slope,
increase y-intercept
-
+
+
+
-
-
-
+
Increase slope,
decrease y-intercept
Decrease slope,
decrease y-intercept

One rule to rule them all



ession
m

Step 1: Start with a random line

Step 2: Pick a large number. 1000
(number of repetitions, or epochs)

Step 3: Pick a small number. 0.01
(learning rate)

Step 4: (repeat 1000 times)

-Pick random point

-If point **above** line, and to the **right** of the y-axis:
add 0.01 to slope
add 0.01 to y-intercept

-If point **above** line, and to the **left** of the y-axis:
subtract 0.01 to slope
add 0.01 to y-intercept

4 cases!

-If point **below** line, and to the **right** of the y-axis:
subtract 0.01 to slope
subtract 0.01 to y-intercept

-If point **below** line, and to the **left** of the y-axis:
add 0.01 to slope
subtract 0.01 to y-intercept

$$SSE = \Sigma(y - \hat{y})^2$$

$$\hat{y} = \beta_0 + \beta_1 x$$ "x"

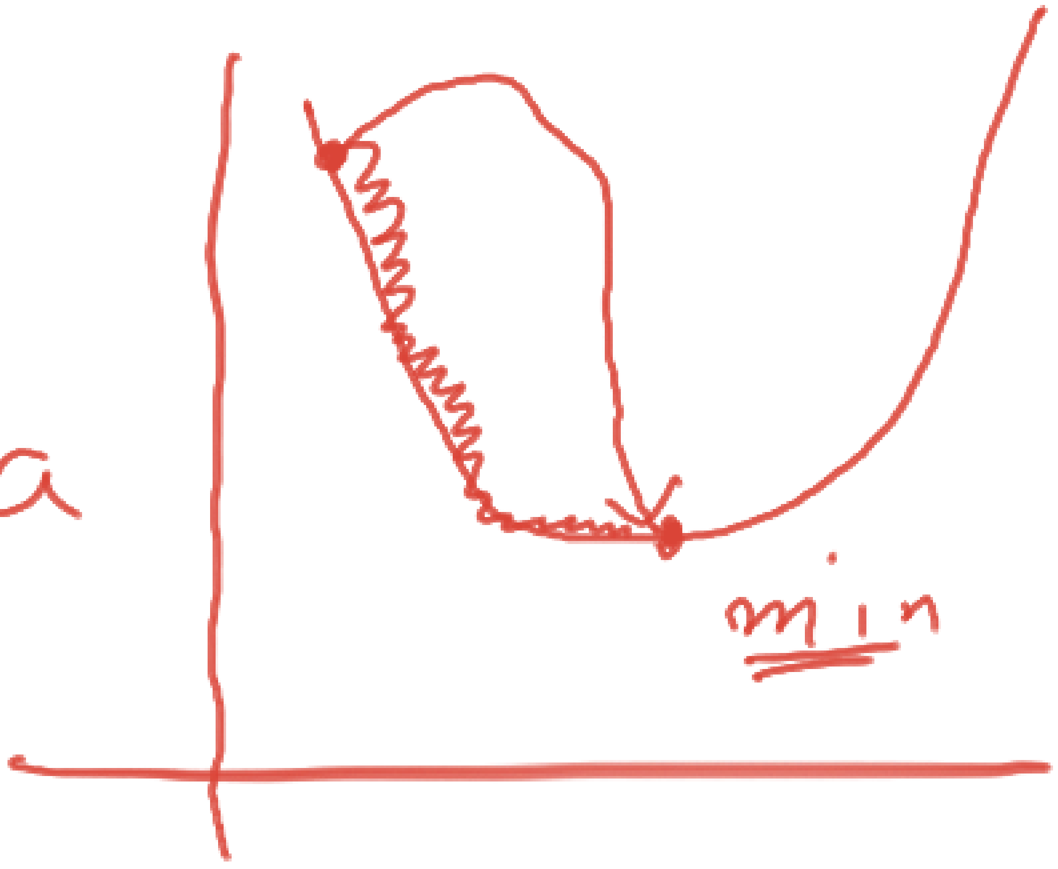
$$SSE = \Sigma(y - \beta_0 - \beta_1 x)^2$$

Gradient descent

β_0 / β_1

y x

data



$$\boxed{\frac{\partial SSE}{\partial \beta_0} = 0}$$

$$\frac{\partial SSE}{\partial \beta_1} = 0$$

$$\frac{dy}{dx} = 0 \qquad y^2 = 4ax$$

$$\frac{d^2y}{dx^2} > 0, \quad < 0$$

$$\beta_0 : \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_0} = \frac{-1}{n} \sum_{i=1}^{n} \left(y - (\beta_0 + \beta_1 X) \right)$$

$$\beta_1 : \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_1} = \frac{-1}{n} \sum_{i=1}^{n} \left(y - (\beta_0 + \beta_1 X) \right) . X$$

Step 1:

Pick a small number (learning rate) 0.01

Step 2:

- Add (learning rate) x (vertical distance) x (horizontal distance) to **slope**
- Add (learning rate) x (vertical distance) to **y-intercept**