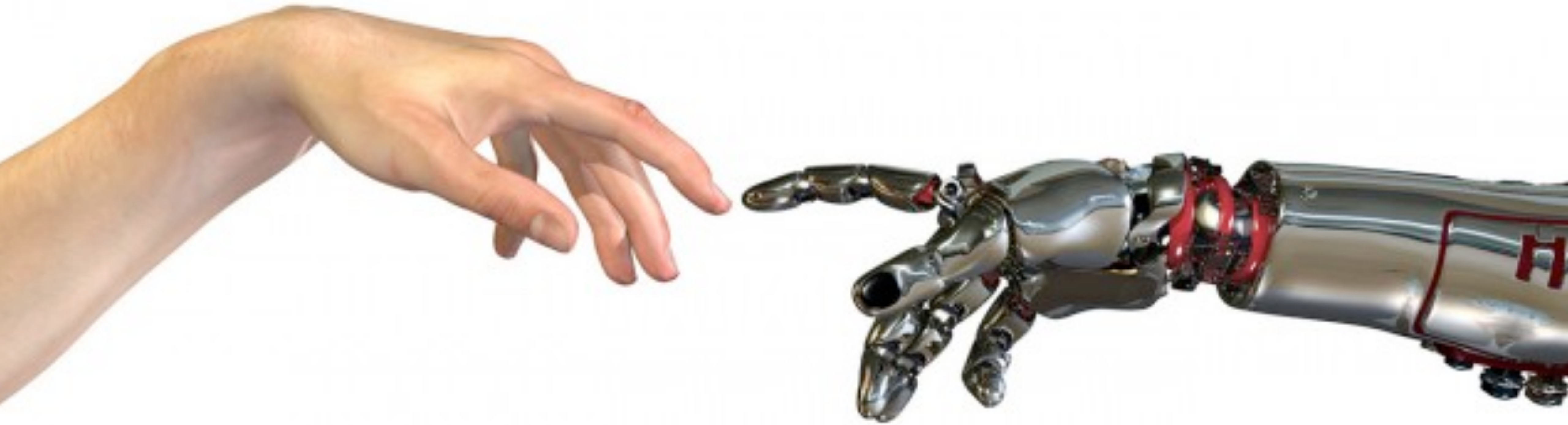


Machine Learning

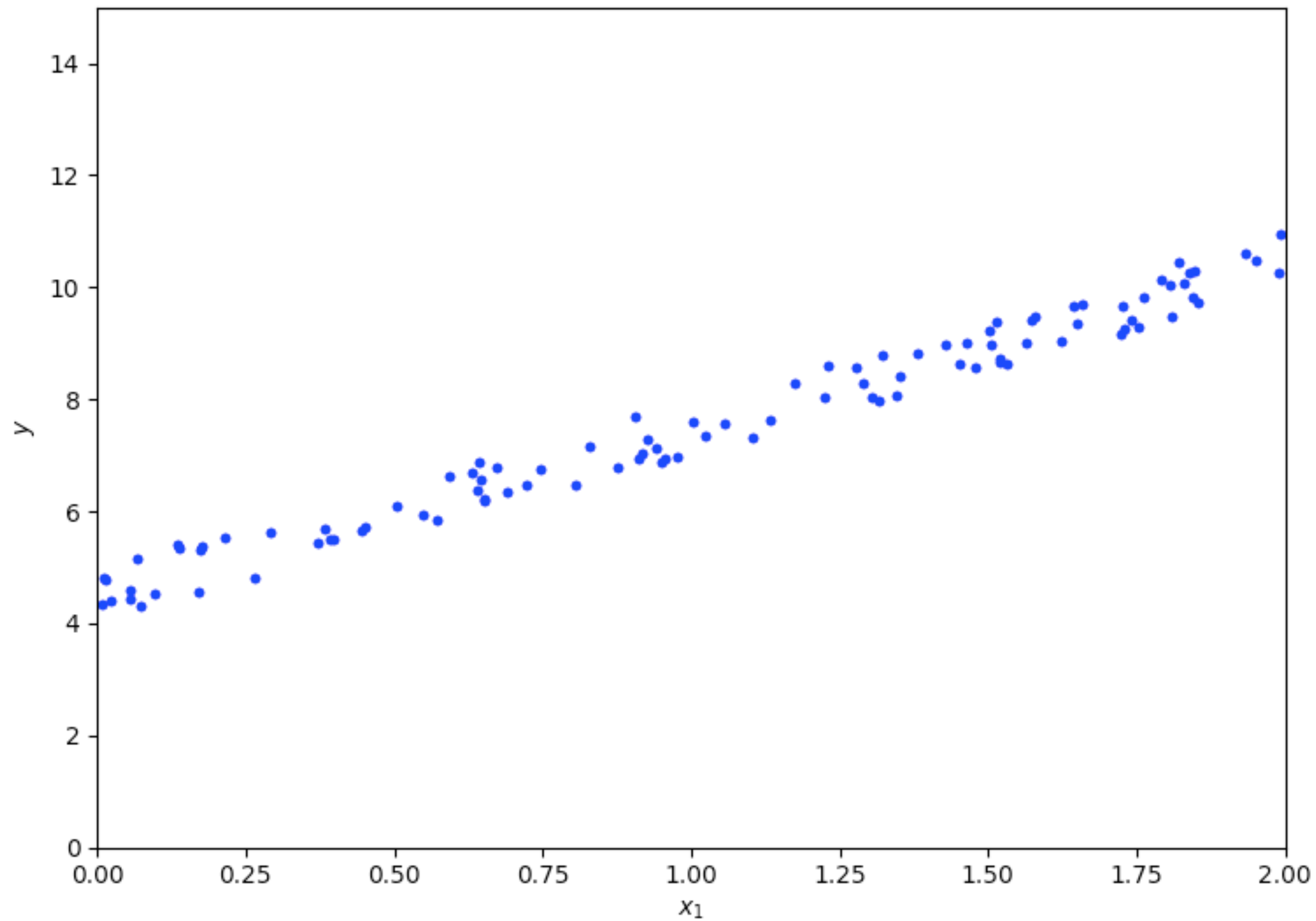
2. regressie en gradient descent

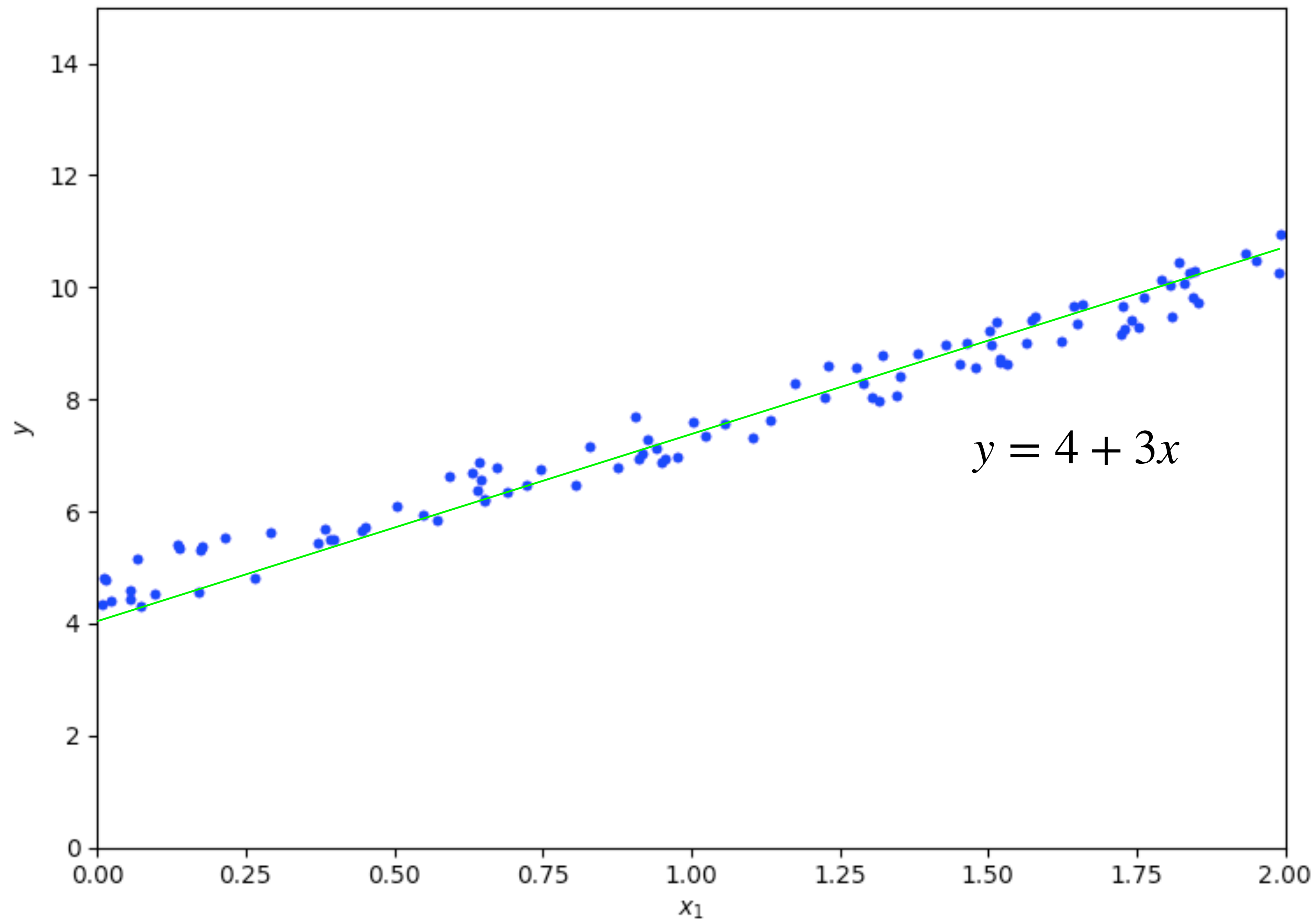


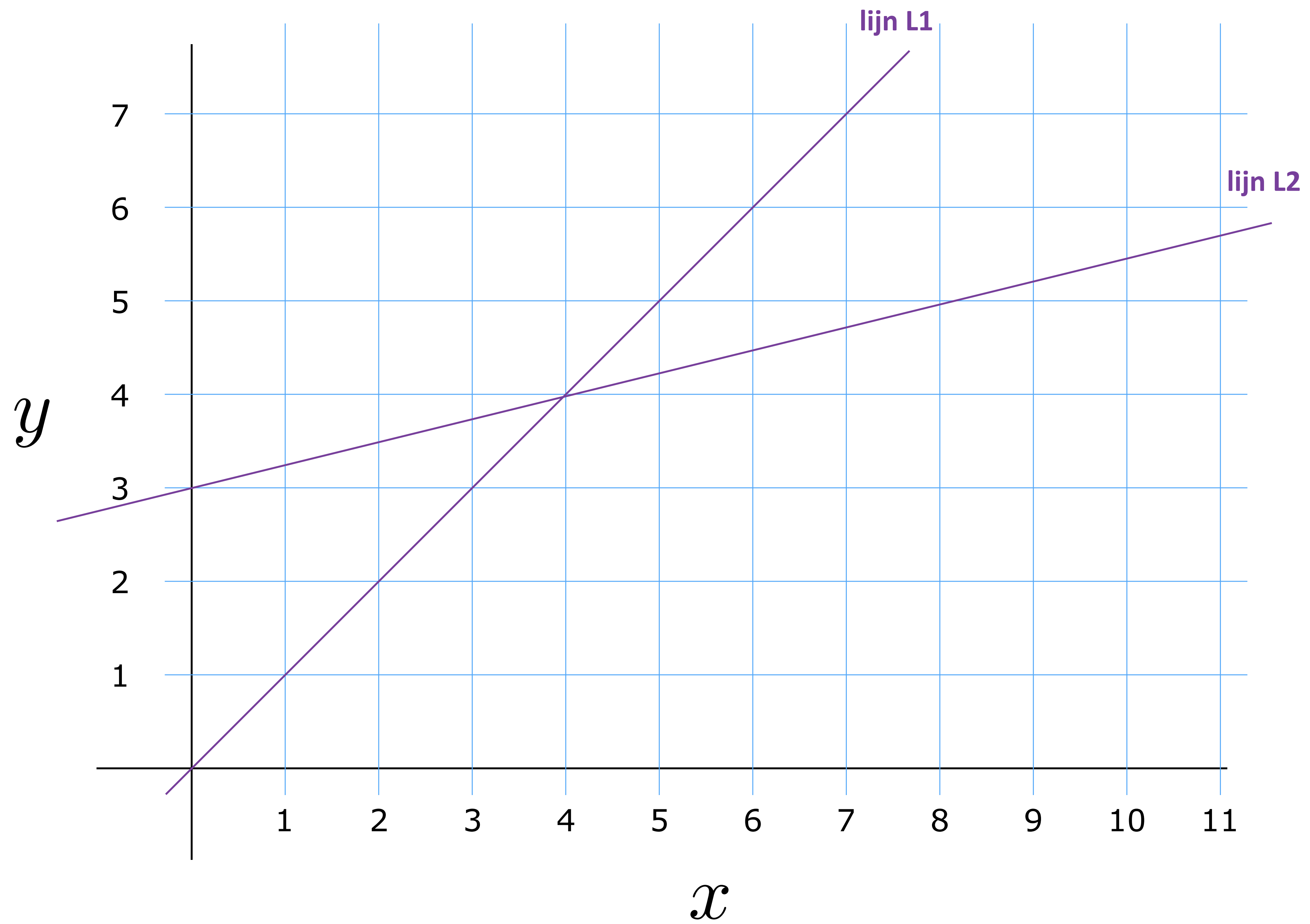
ml: regressie

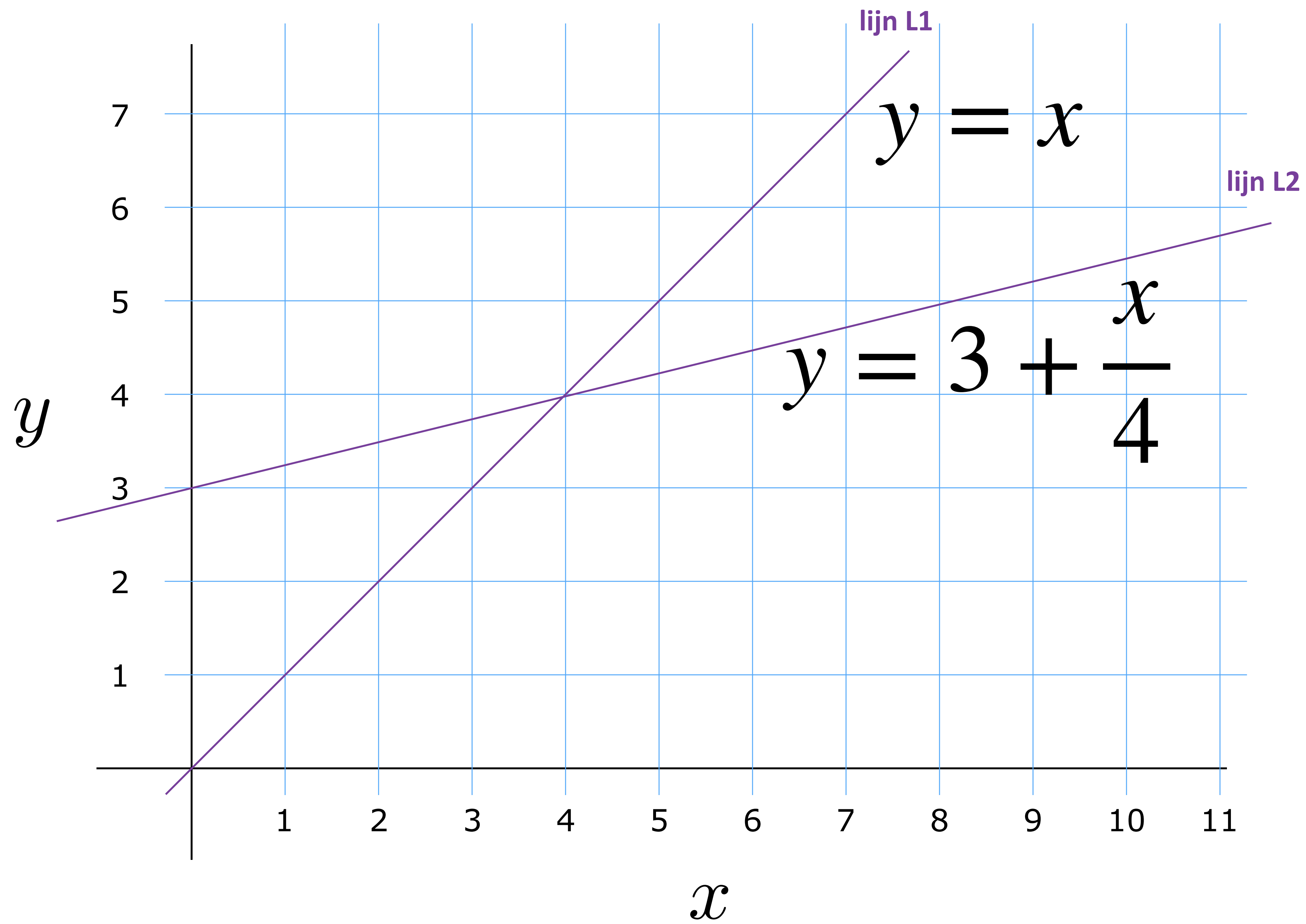
Lineaire regressie

- Met behulp van een model...
- Op basis van 1 (simple) of meer (multiple) features...
- ...proberen een uitkomstwaarde te voorspellen

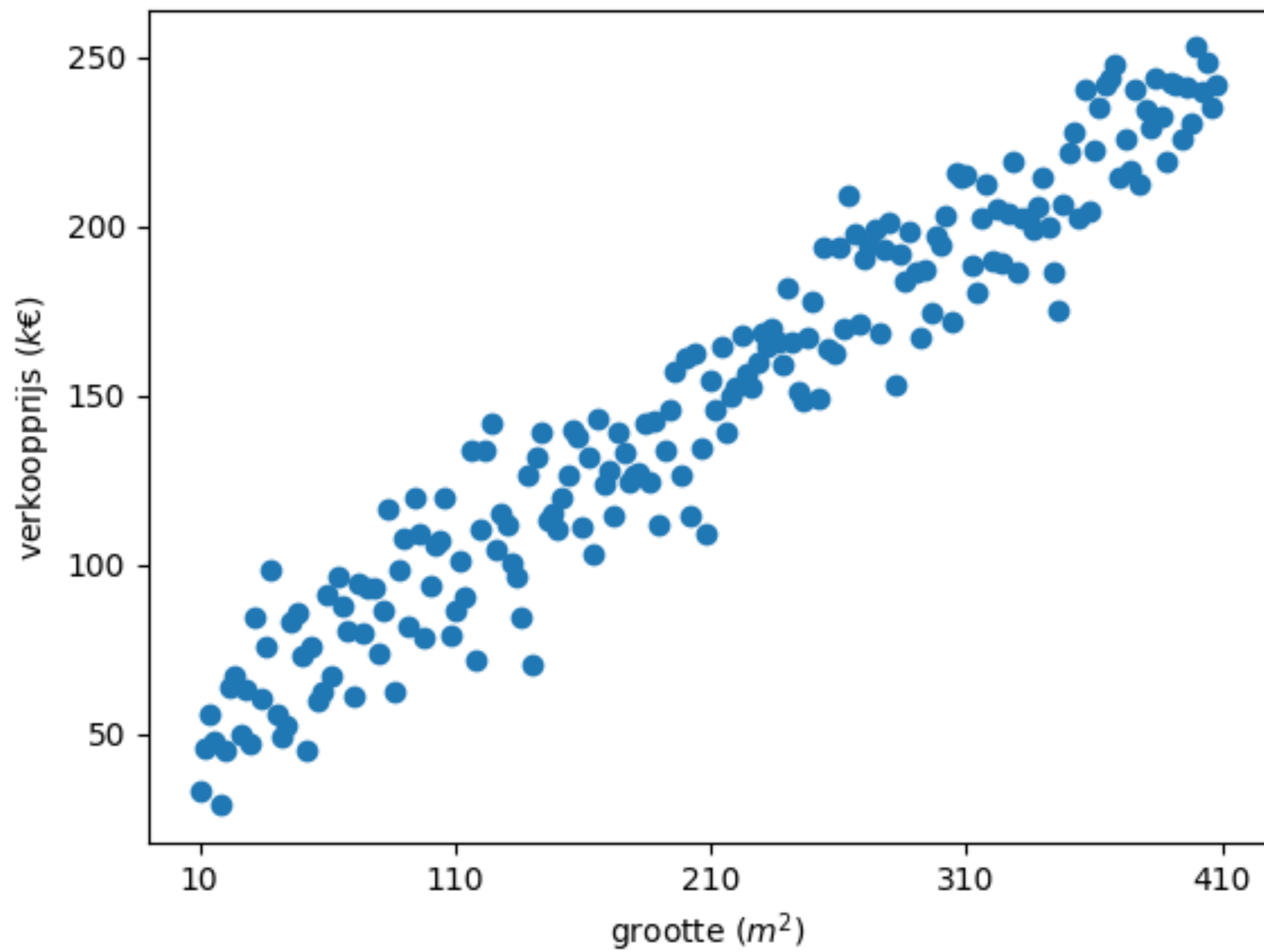




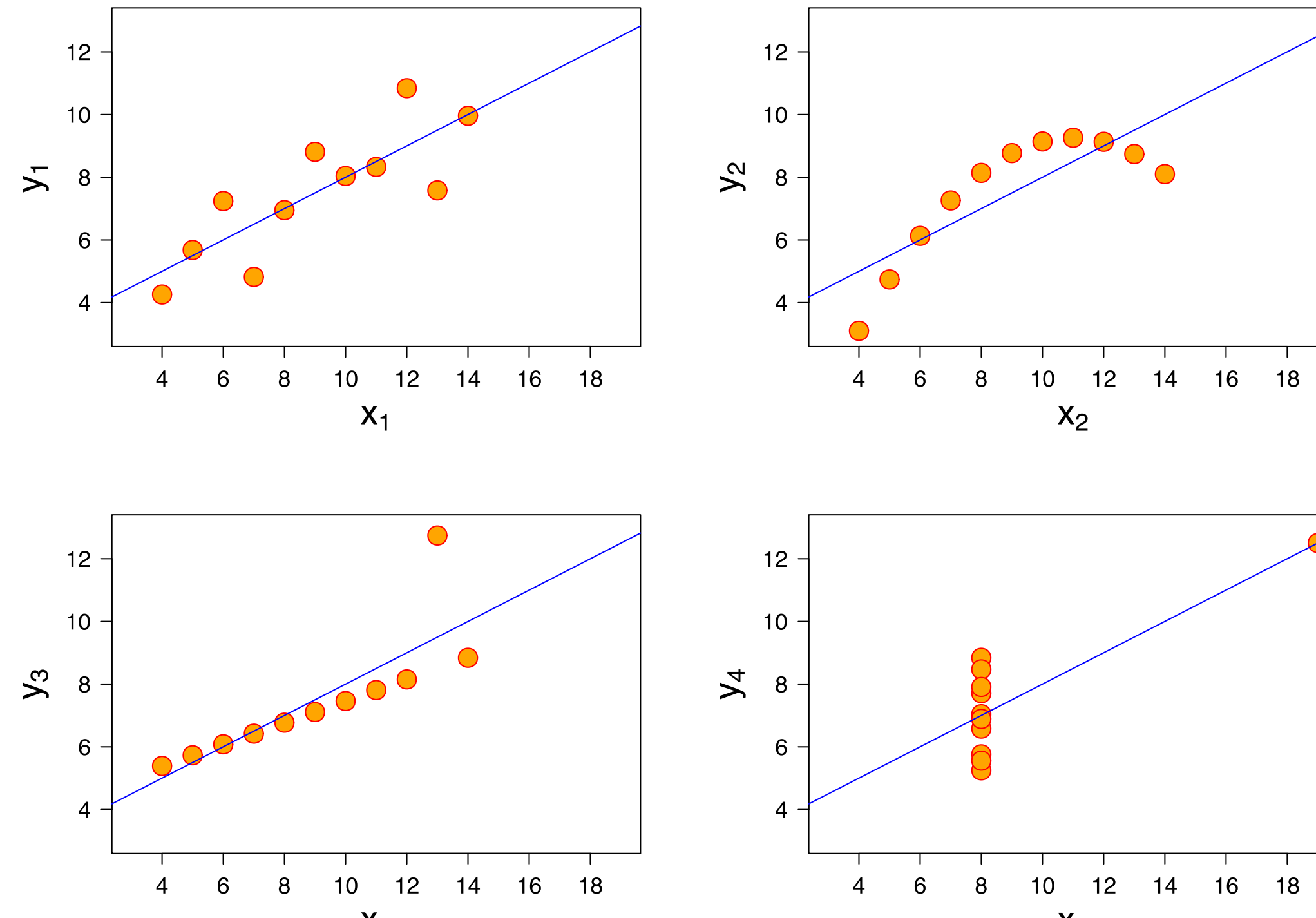




Verkoopprijs huizen Groningen

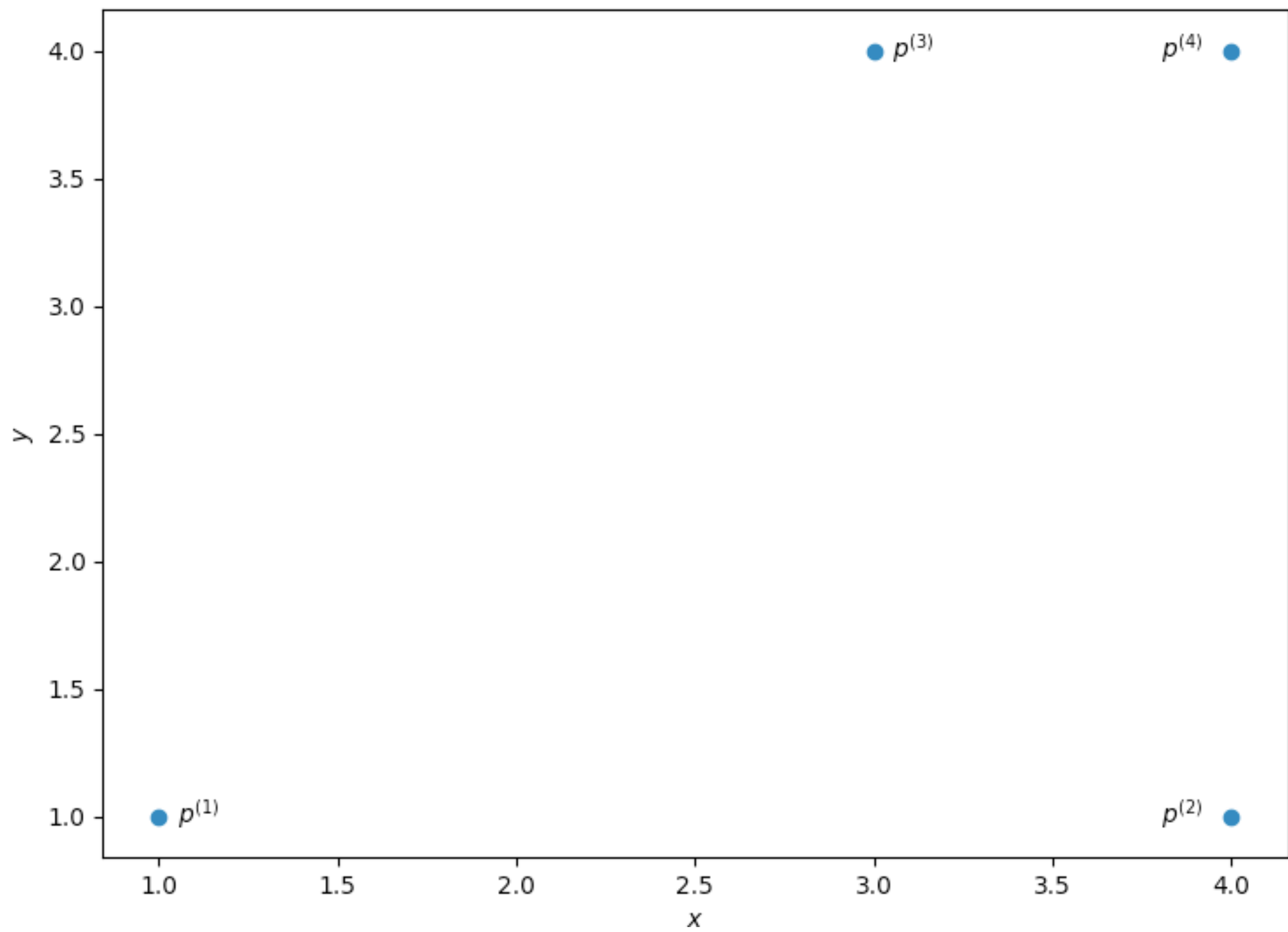


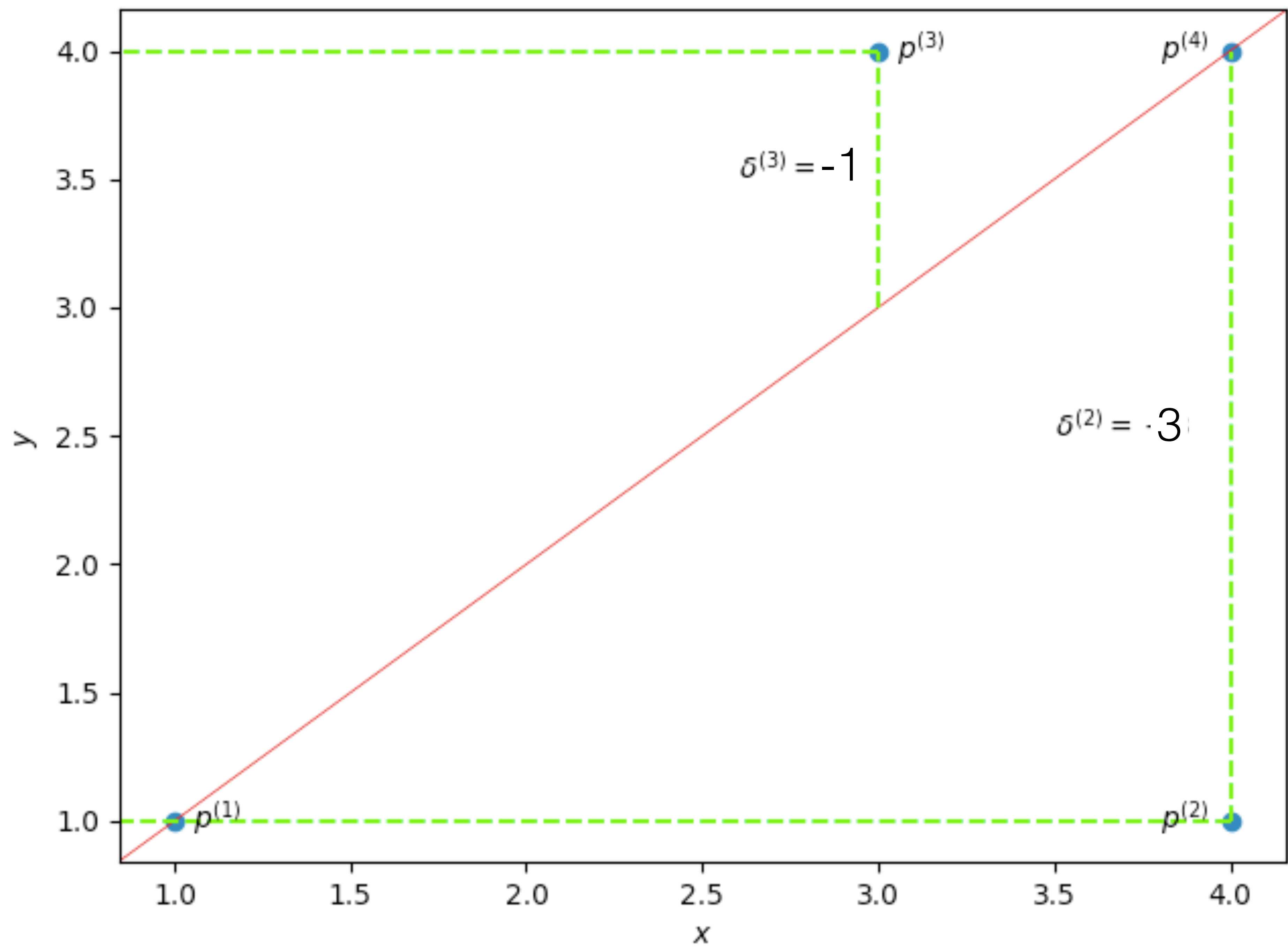
Let op: Anscombe's Quartet

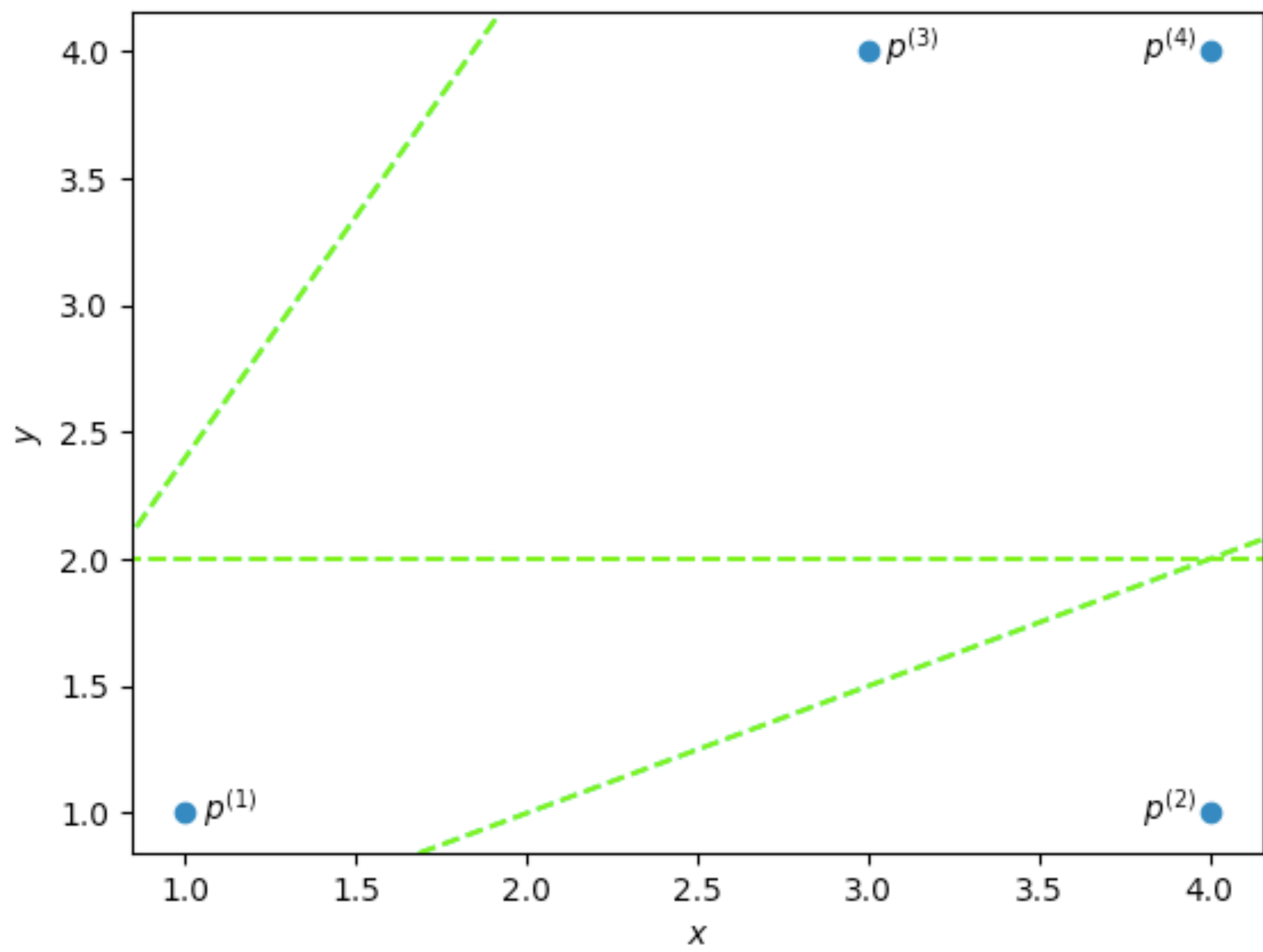


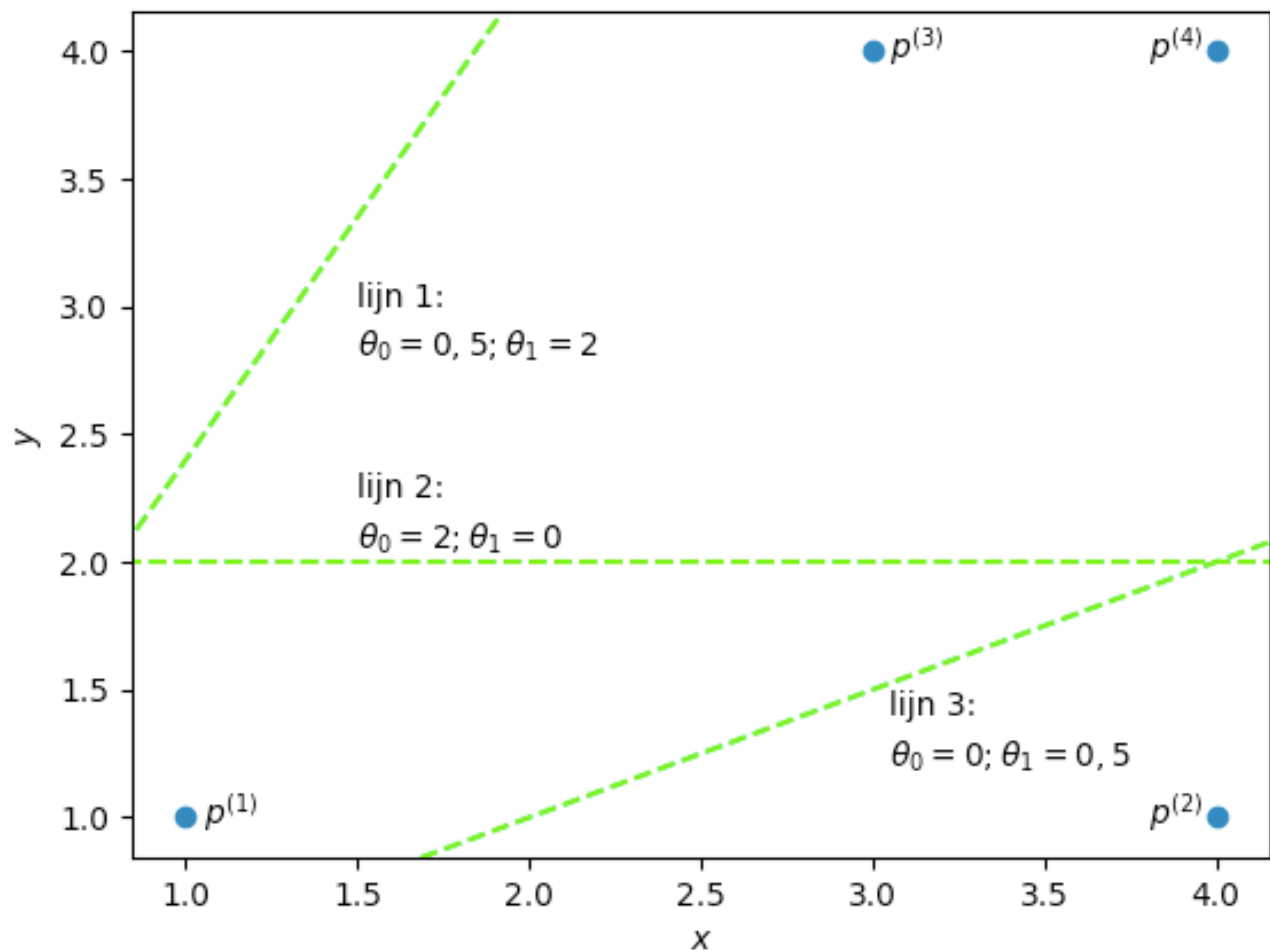
4 zeer verschillende verdelingen met dezelfde optimale regressielijn

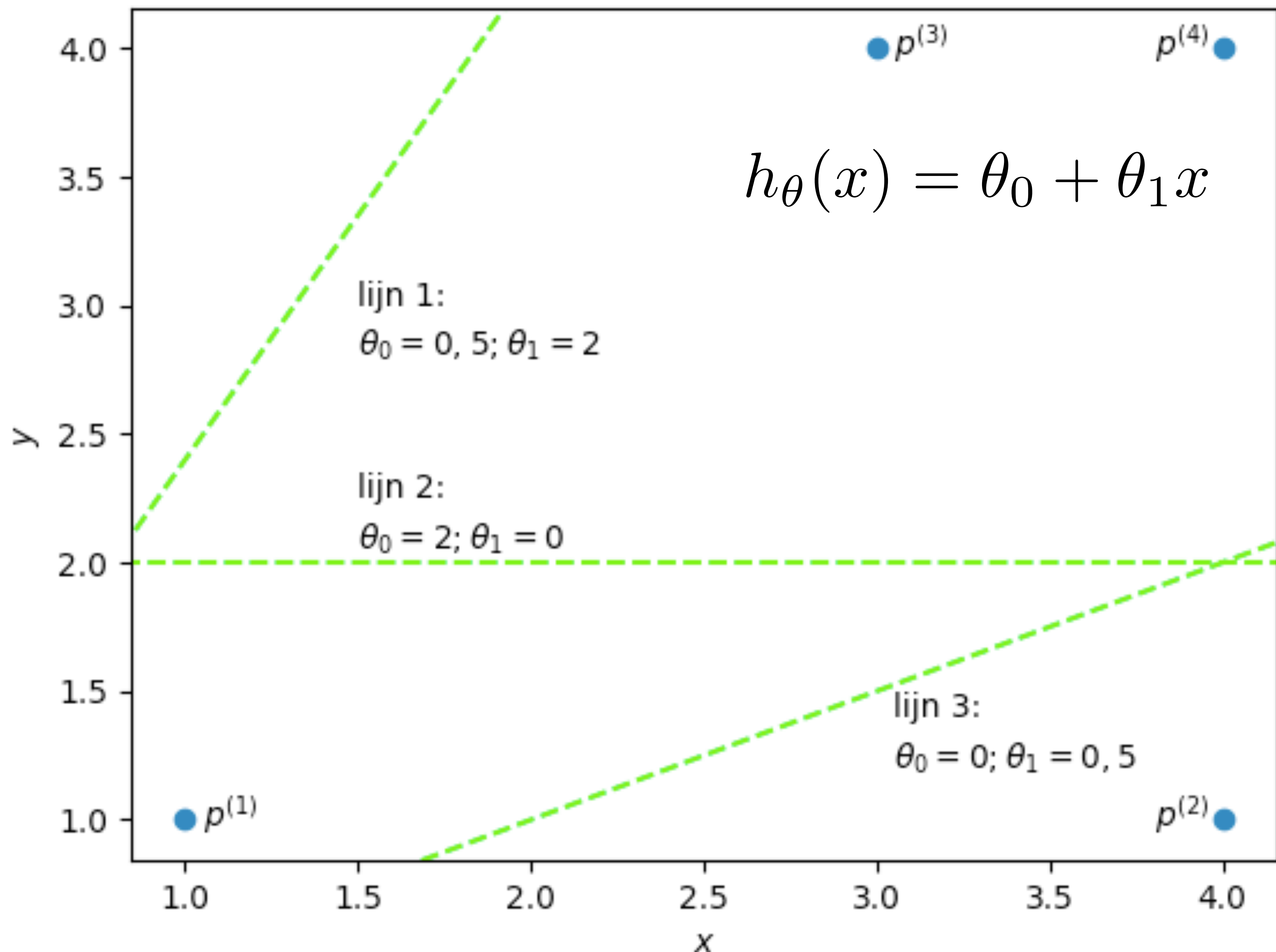
ml: fouten en gewichten

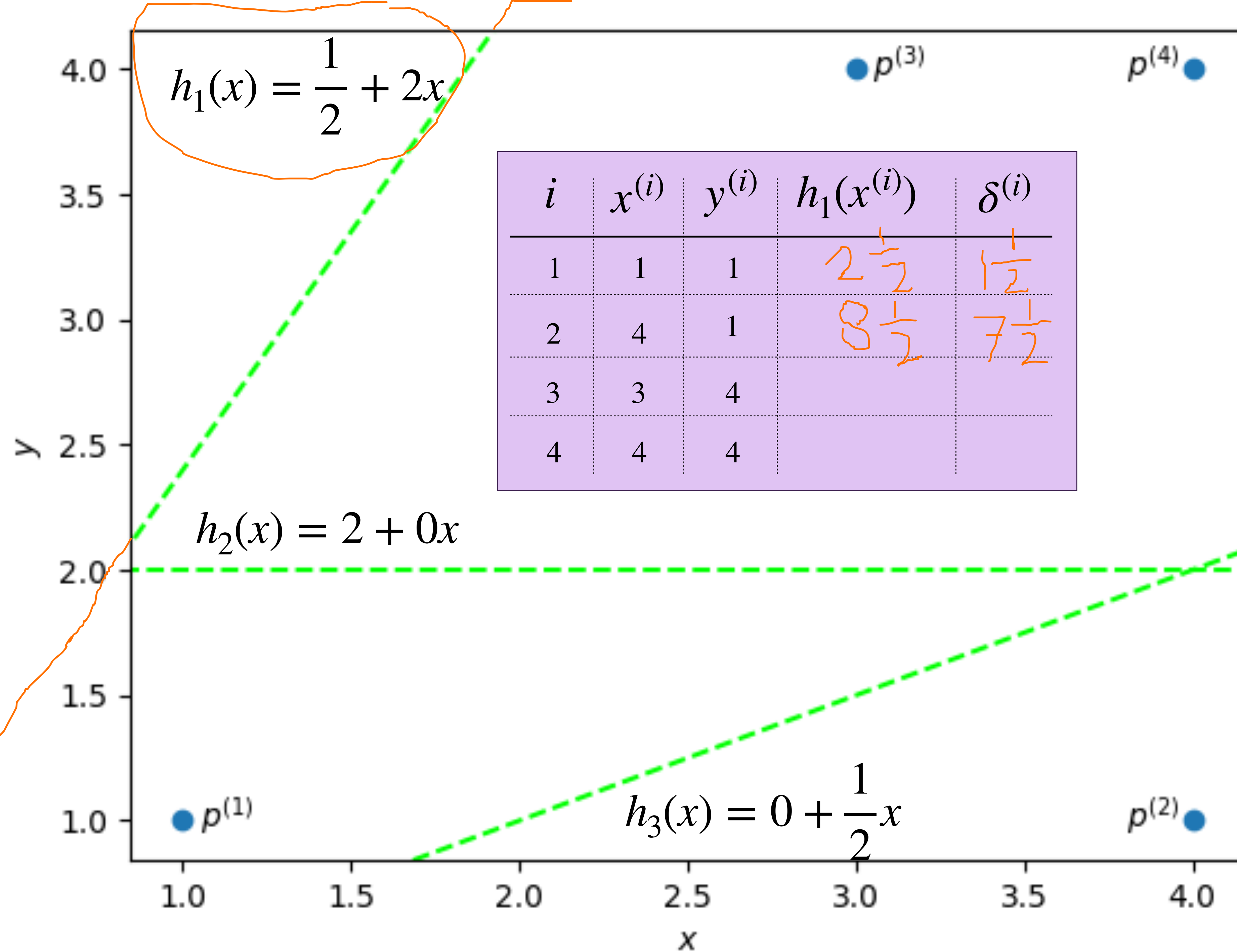












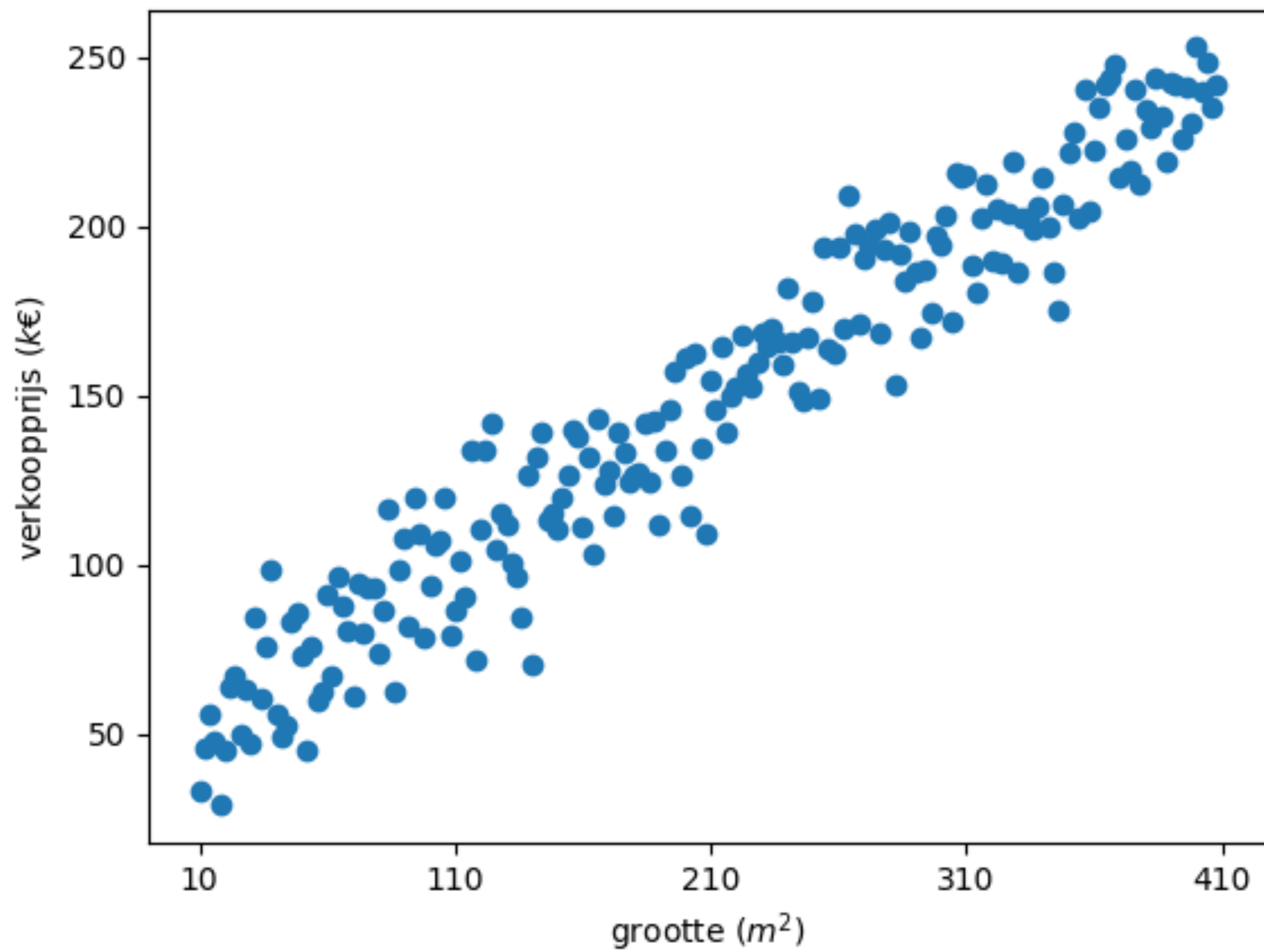
Kostenfunctie (J) voor 1 feature

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

Waarom kwadraat?

m_l : meerdere eigenschappen

Verkoopprijs huizen Groningen



Eén feature

grootte (m ²)	verkoopprijs (€)
127	279.500
101	195.000
120	167.500
135	290.000
183	534.500
180	315.000
96	189.000
70	115.000
160	449.000
...	...

1 grootte (m ²)	2 aantal kamers	3 tuin	4 energielabel	5 verkoopprijs (€)
127	3	j	A	279.500
101	2	n	C	195.000
120	2	j	B	167.500
135	4	j	C	290.000
183	3	n	D	534.500
...

notatie-afspraken

m	aantal observaties
n	aantal eigenschappen (per observatie)
$x^{(i)}$	observatie nummer i
$x_j^{(i)}$	eigenschap j van observatie nummer i
$y^{(i)}$	actuele waarde van observatie i
θ_j	factor waarmee waarde van feature j moet worden vermenigvuldigd

Hypotheses o.b.v. 1 of meer features

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

X

127	3	j	A
101	2	n	C
120	2	j	B
135	4	j	C
183	3	n	D

θ

θ_0
θ_1
θ_2
θ_3
θ_4



Intermezzo: vermenigvuldigen van matrices met vectoren

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix}$$

Dimensies (rijen x kolommen):

$$A = \{ 2 \times \mathbf{3} \}$$

$$B = \{ \mathbf{3} \times 1 \}$$

=>

$$A.B = \{ 2 \times 1 \}$$

X

$$\begin{bmatrix} 1 & 127 & 3 & j & A \\ 1 & 101 & 2 & n & C \\ 1 & 120 & 2 & j & B \\ 1 & 135 & 4 & j & C \\ 1 & 183 & 3 & n & D \end{bmatrix}$$

θ

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

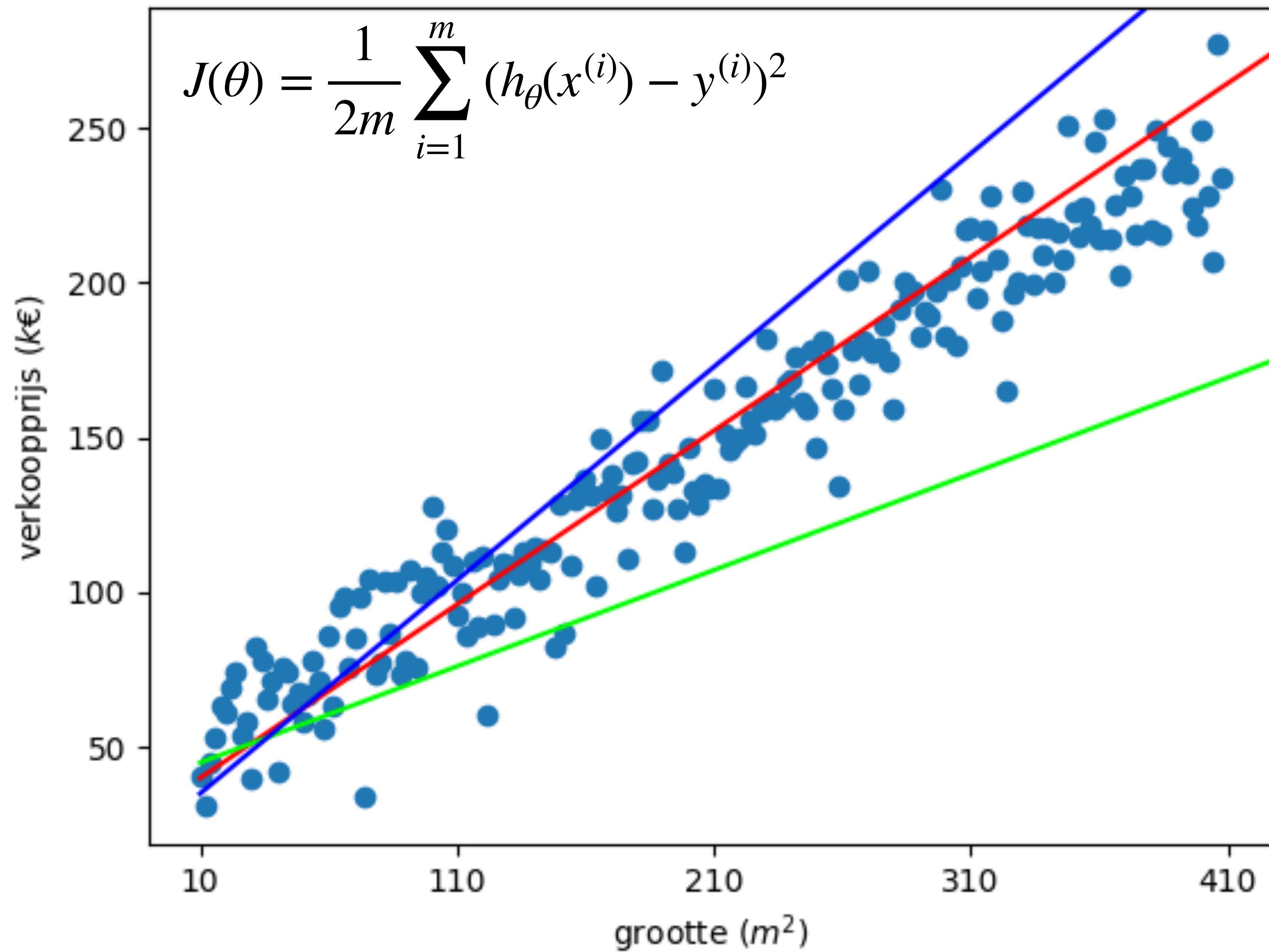
Algemene kostenfunctie (J)

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

$$h_{\theta}(x^{(i)}) = \theta^T x^{(i)}$$

ml: ordinary least square

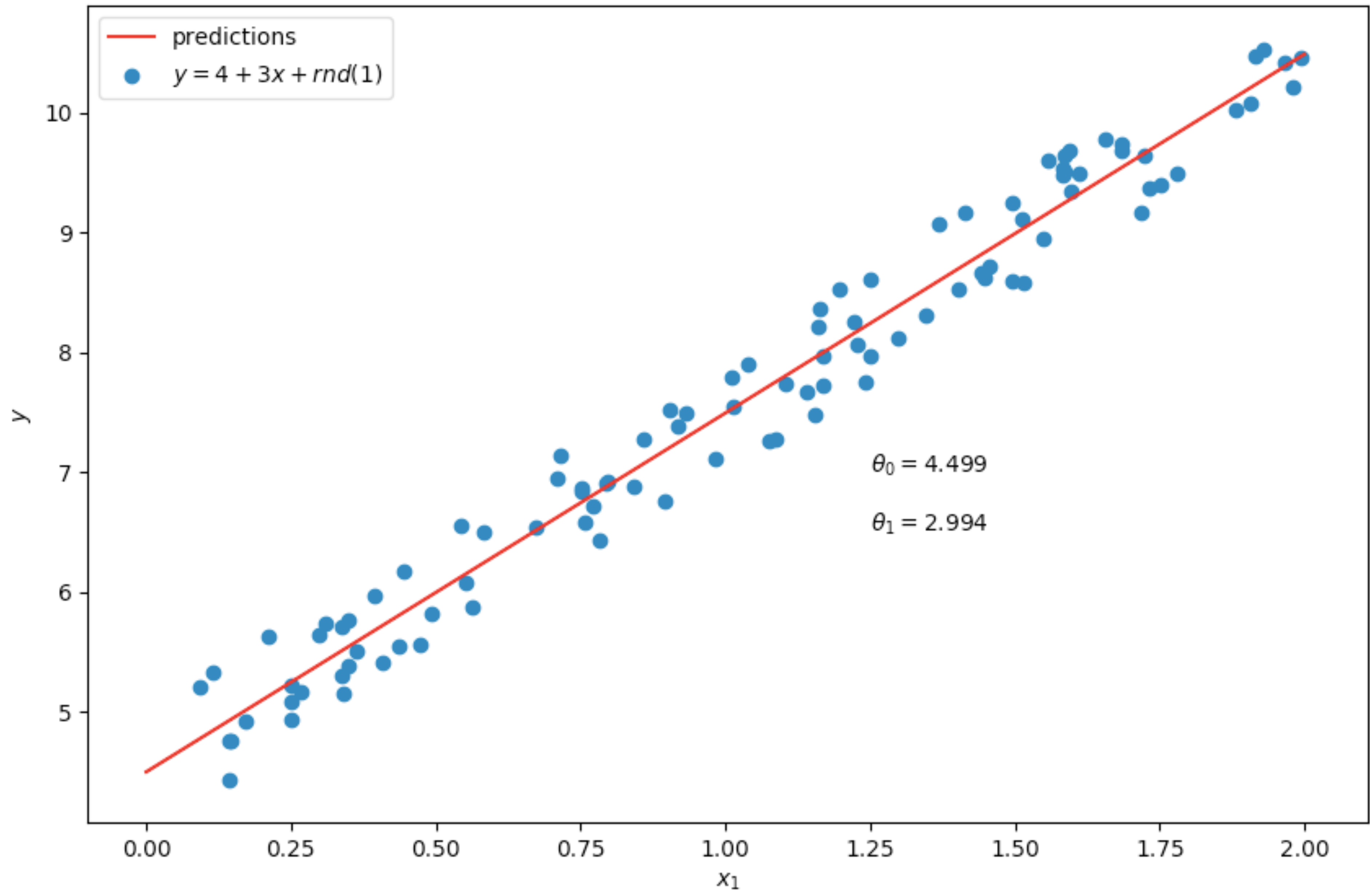
Verkoopprijs huizen Groningen



Gesloten oplossing: Ordinary Least Squares (OLS)

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

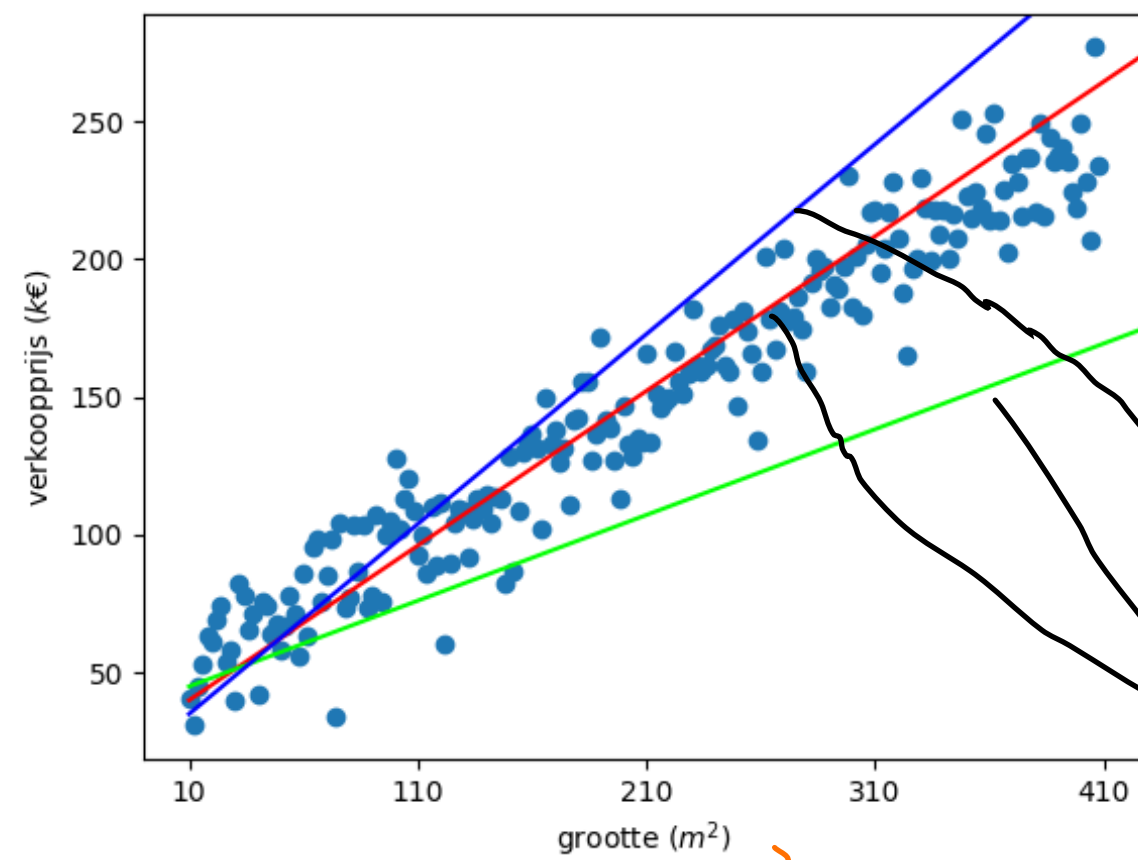
```
theta = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```



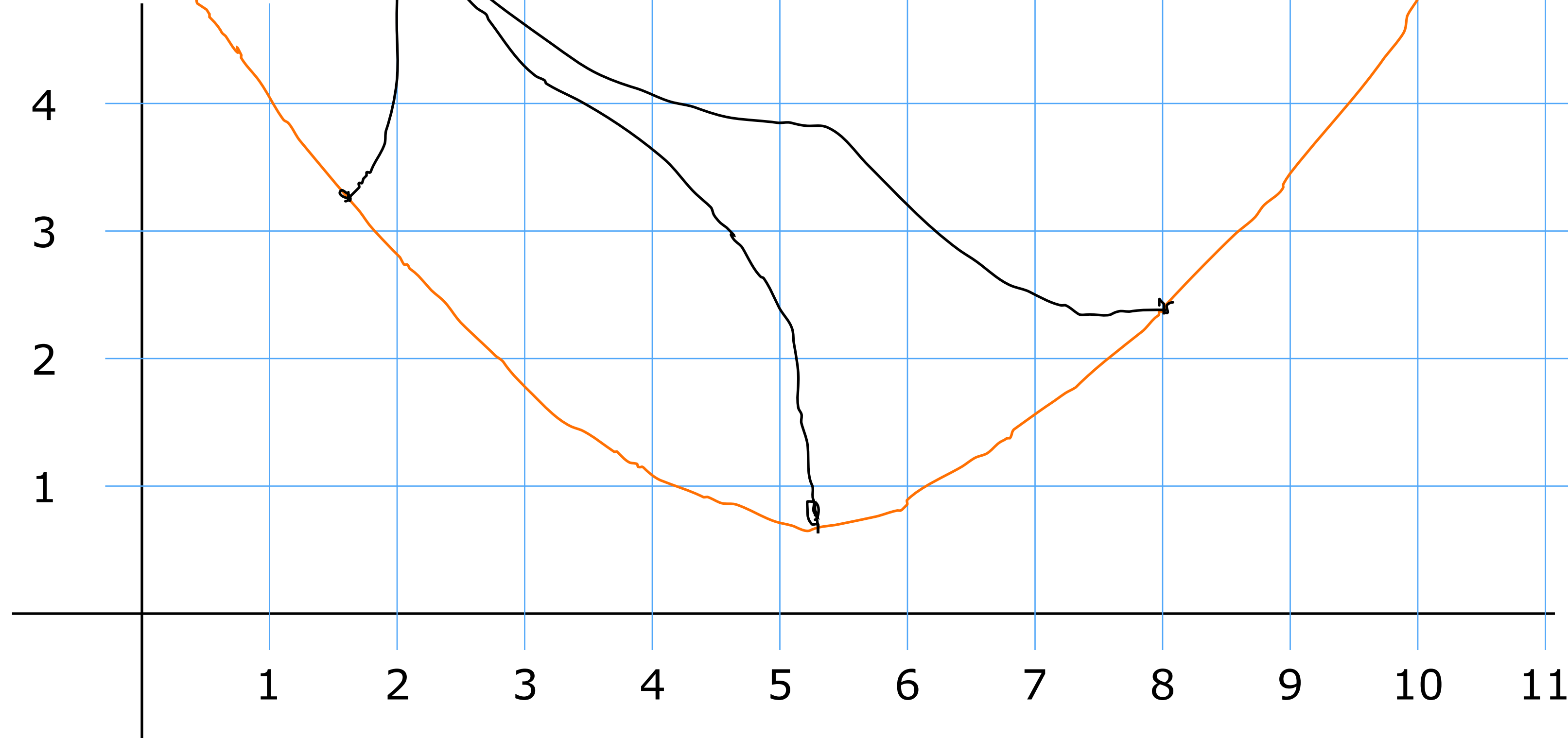


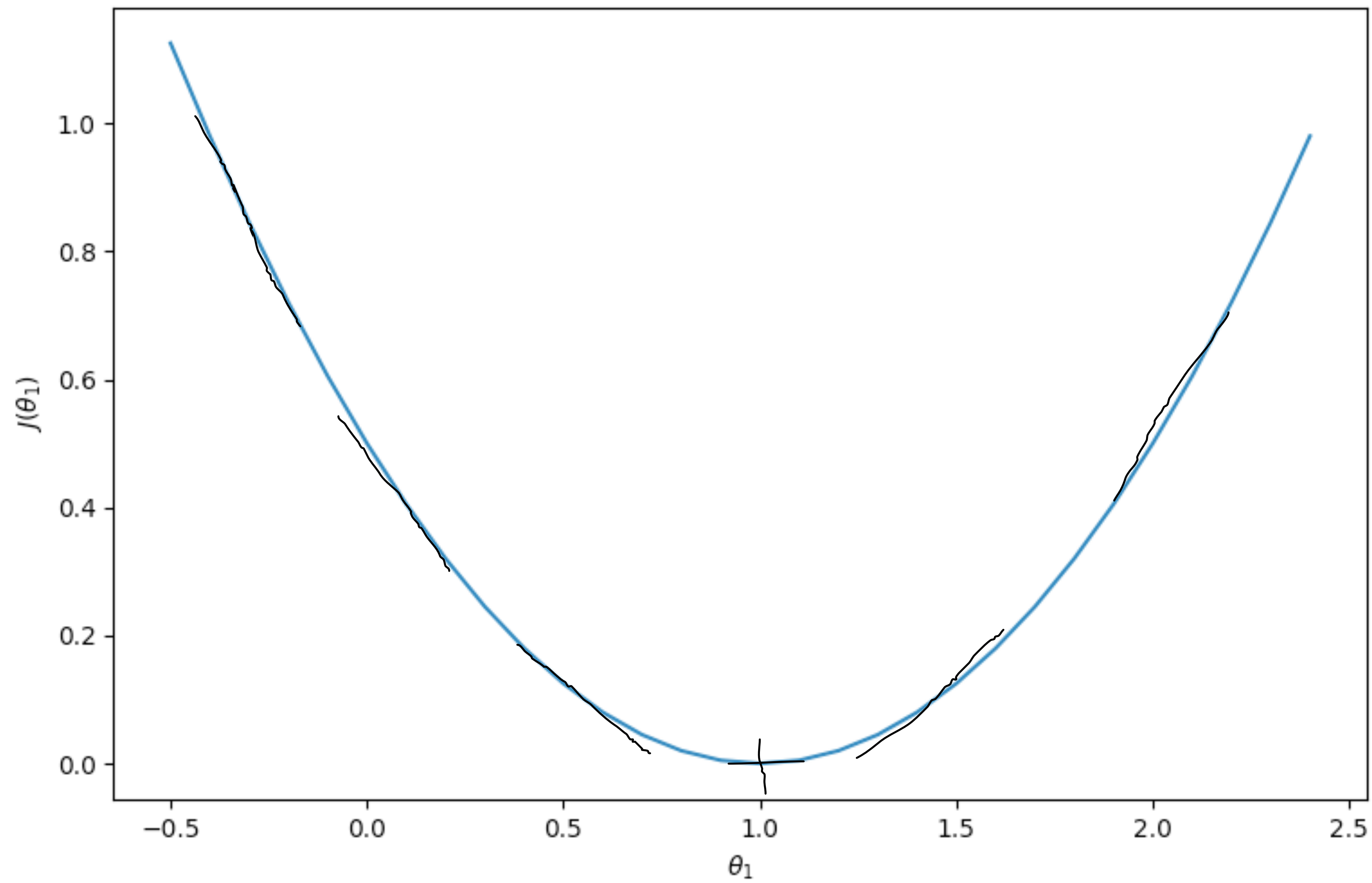
ml: gradient descent

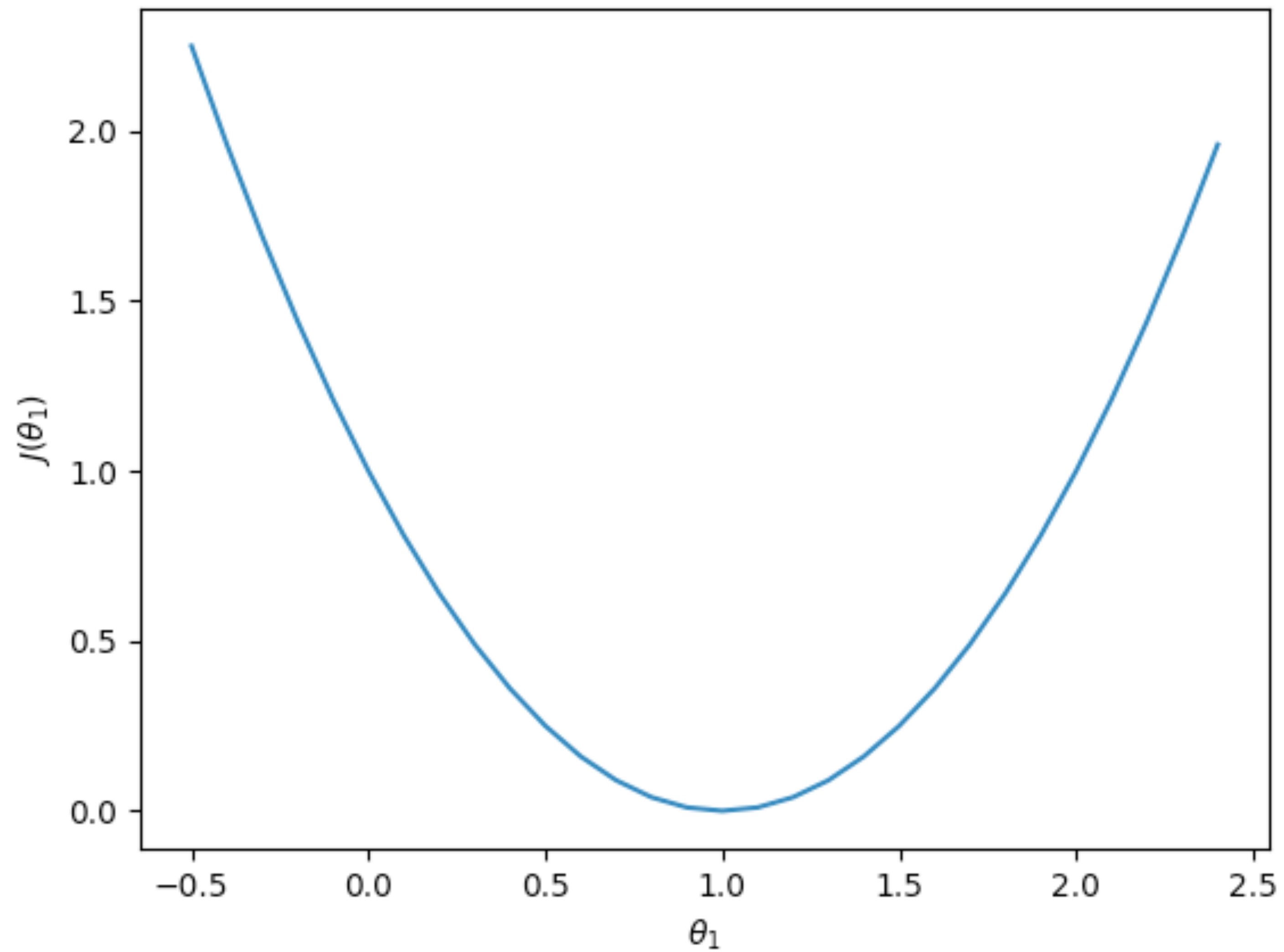
Verkoopprijs huizen Groningen



$J(\theta_1) \uparrow$





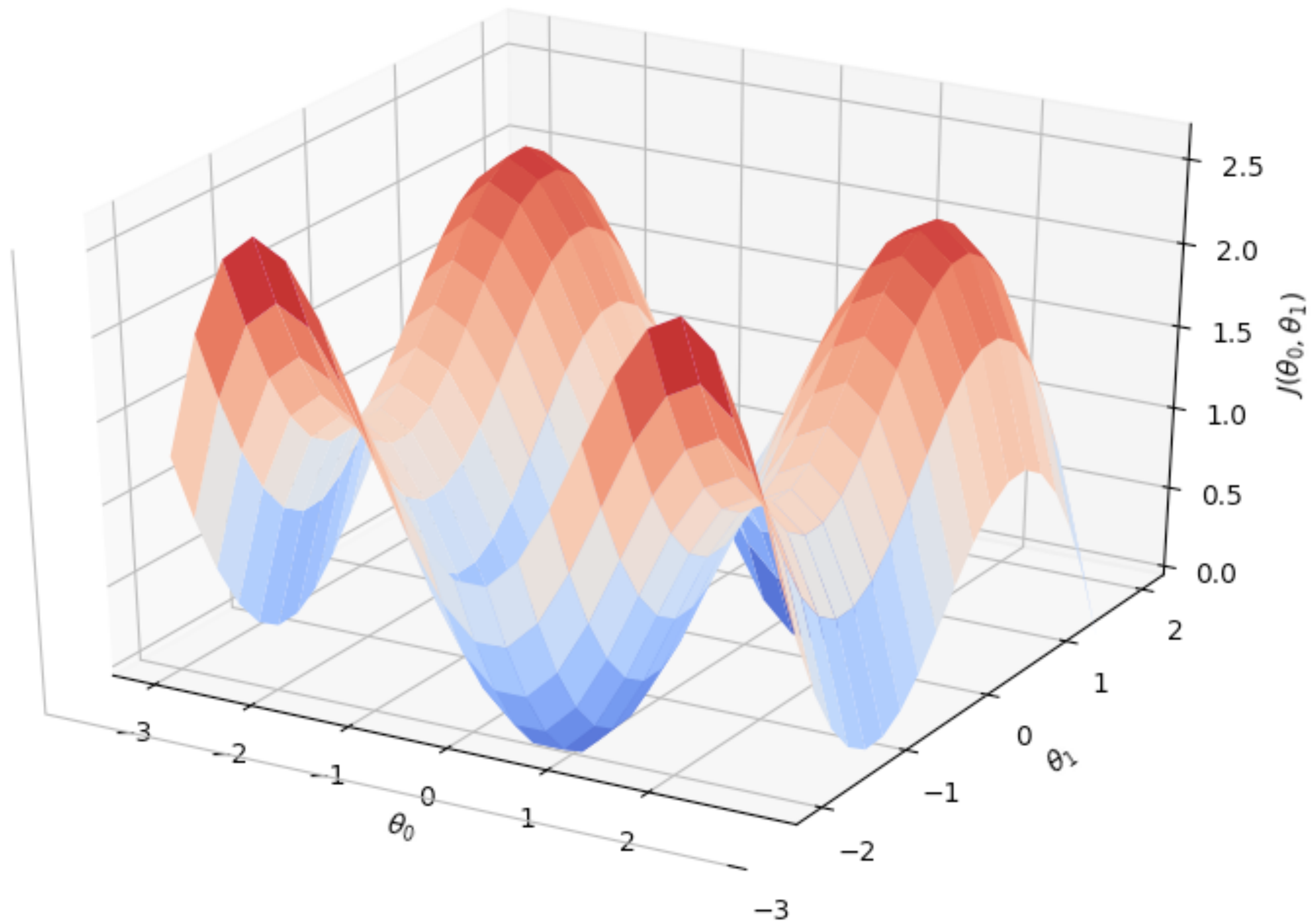


$$y = \frac{1}{2}(1 - x)^2$$

$$y = \frac{1}{2}x^2 - x + \frac{1}{2}$$

$$y' = x - 1$$





Herhaling: kostenfunctie

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

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

Aanpassing van de gewichten

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

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

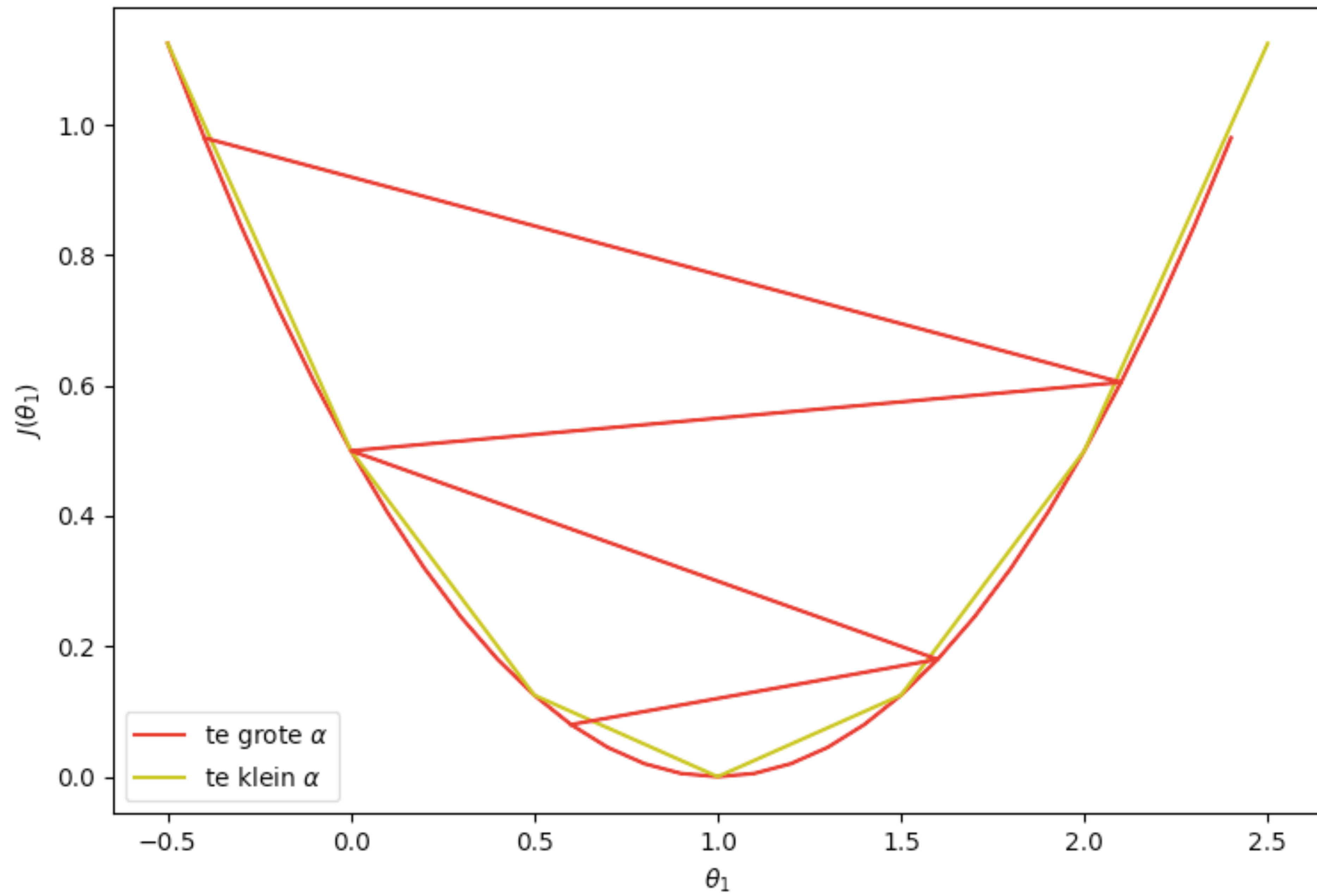
Stappenplan

update alle $\theta_j, j = 1, j = 2, \dots, j = n$

herhaal totdat een minimum bereikt is:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

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





Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.