SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

# 10.020 Data Driven World

## *Linear Regression: Training*

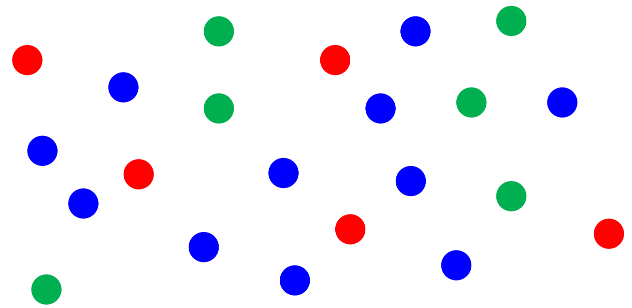
Peng Song, ISTD

Week 9, Lesson 1, 2021

some slides are from Jia-Bin Huang

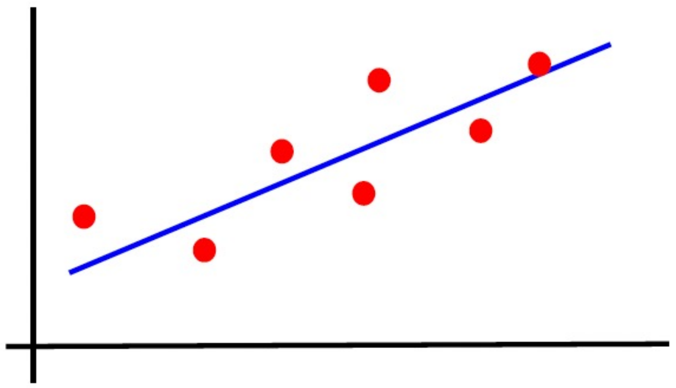
# Revision: Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict $y$ given $x$
  - y is **categorical**

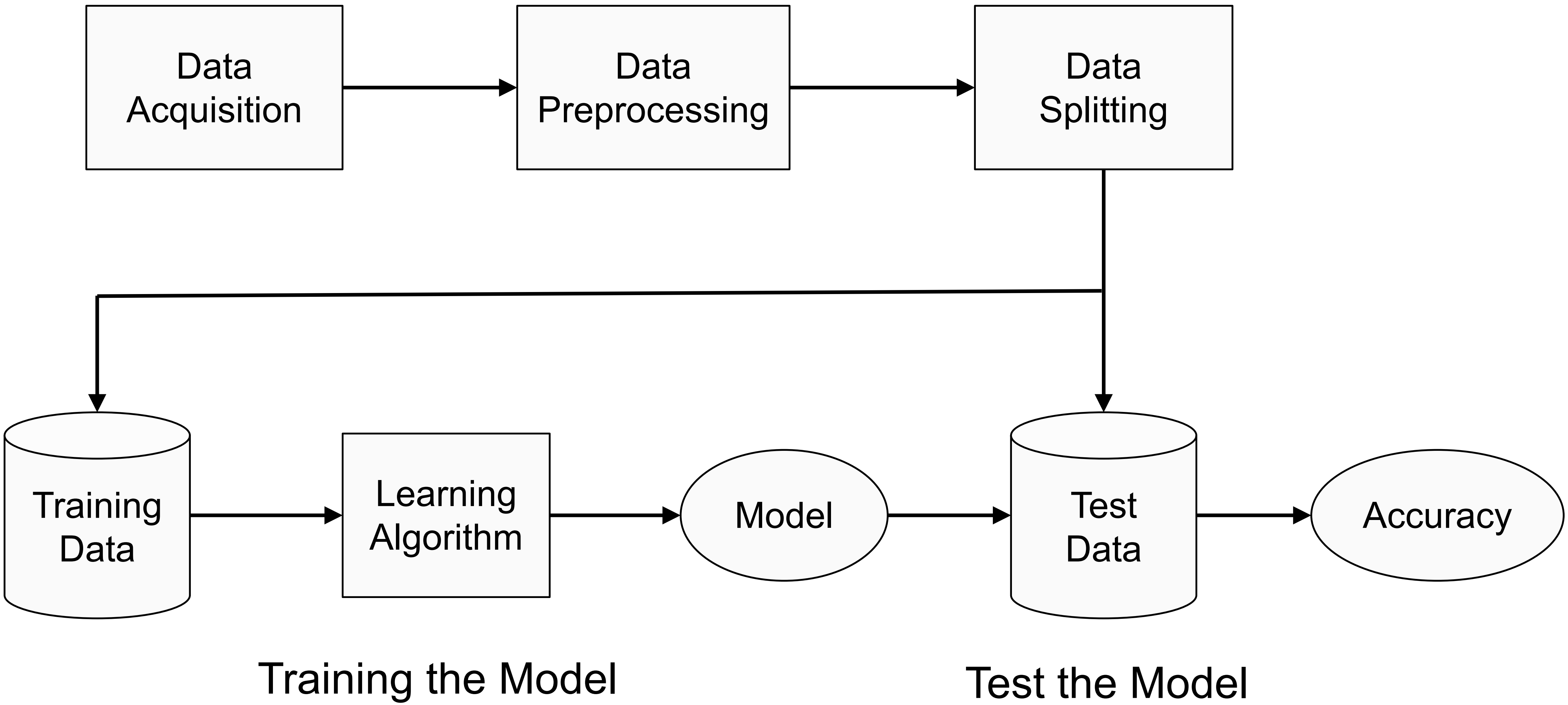


We will focus on Regression this week

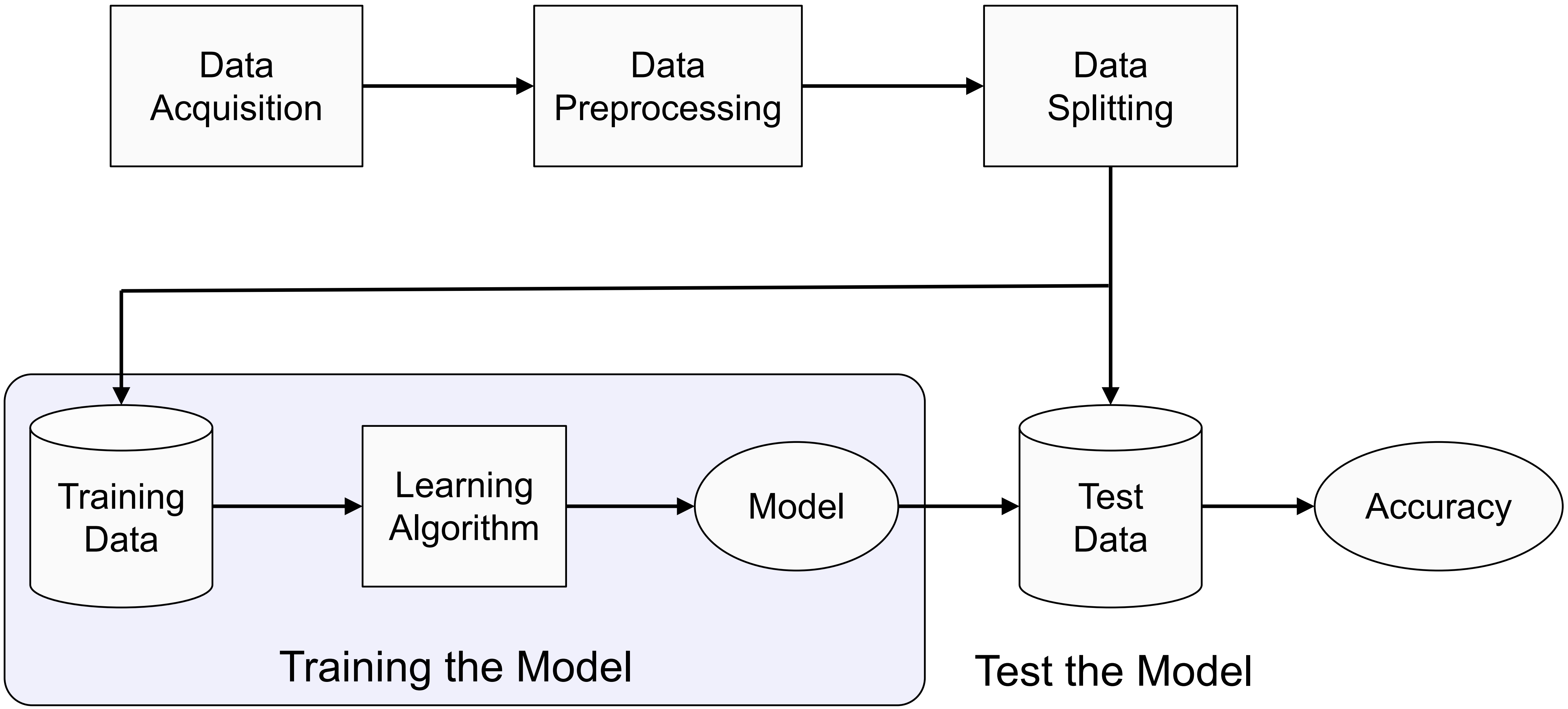
- **Regression**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict $y$ given $x$
  - y is **numeric**



# Revision: Supervised Learning Process



# Today: Linear Regression

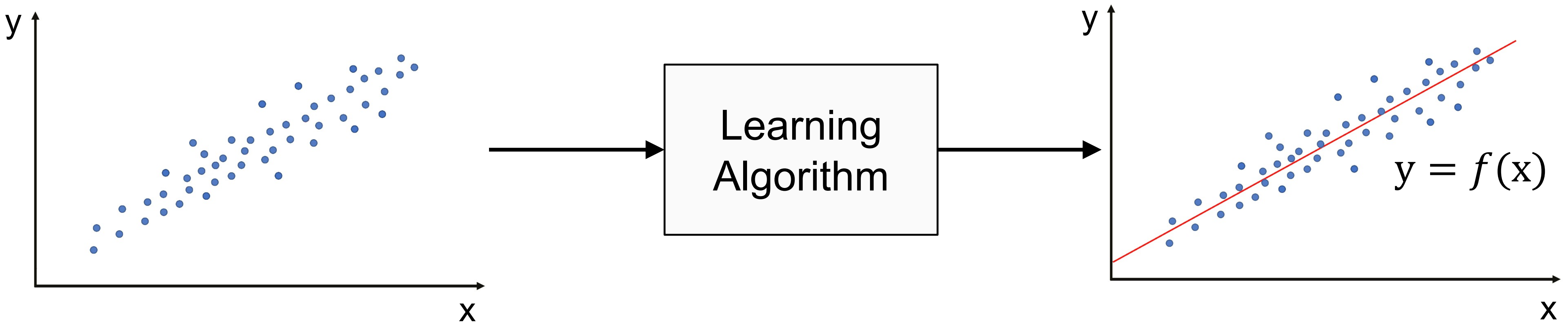


# Regression Problem: House Pricing Prediction

- Given samples $(x, y)$, where
  - $\mathrm{x}$ is house size
  - $\mathrm{y}$ is house price
- Learn a function $\mathrm{y} = f(\mathrm{x})$
  - Assume $f$ is a linear function

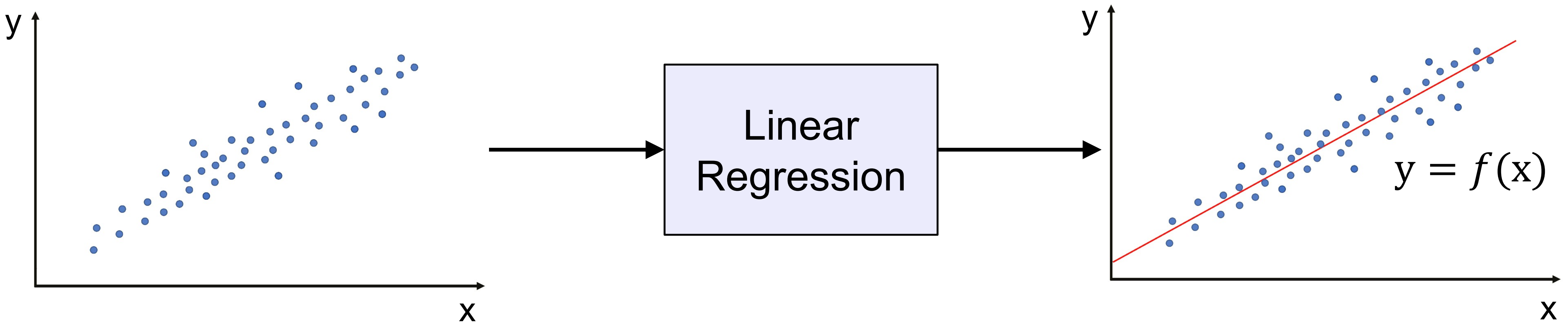
# Regression Problem: House Pricing Prediction

- Given samples $(x, y)$, where
  - x is house size
  - y is house price
- Learn a function $\mathrm{y} = f(\mathrm{x})$
  - Assume $f$ is a linear function



# Linear Regression

Linear regression is a machine learning algorithm that assumes a **linear** relationship between the input variable $x$ and the single output variable $y$.



# Training Set

| Size in feet^2 ($x$) | Price ($) in 1000's ($y$) |
|---|---|
| 2104 | 460 |
| 1416 | 232 |
| 1534 | 315 |
| 852 | 178 |
| … | … |

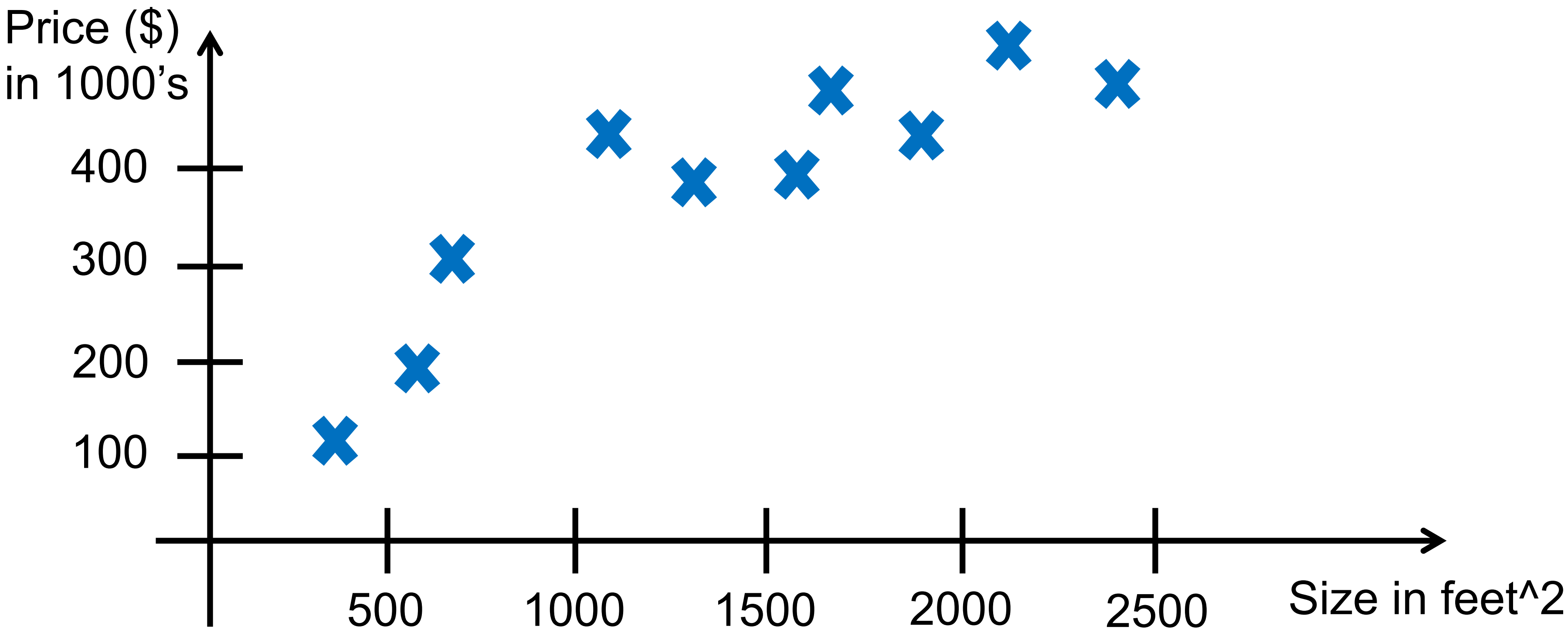
$m$ = 47

- $m$ = Number of training examples
- $x$ = Input variable / feature
- $y$ = Output variable / target variable
- ($x^i$, $y^i$) = $i^{th}$ training example

Examples:
$x^1 = 2104$
$x^2 = 1416$
$y^1 = \ 460$

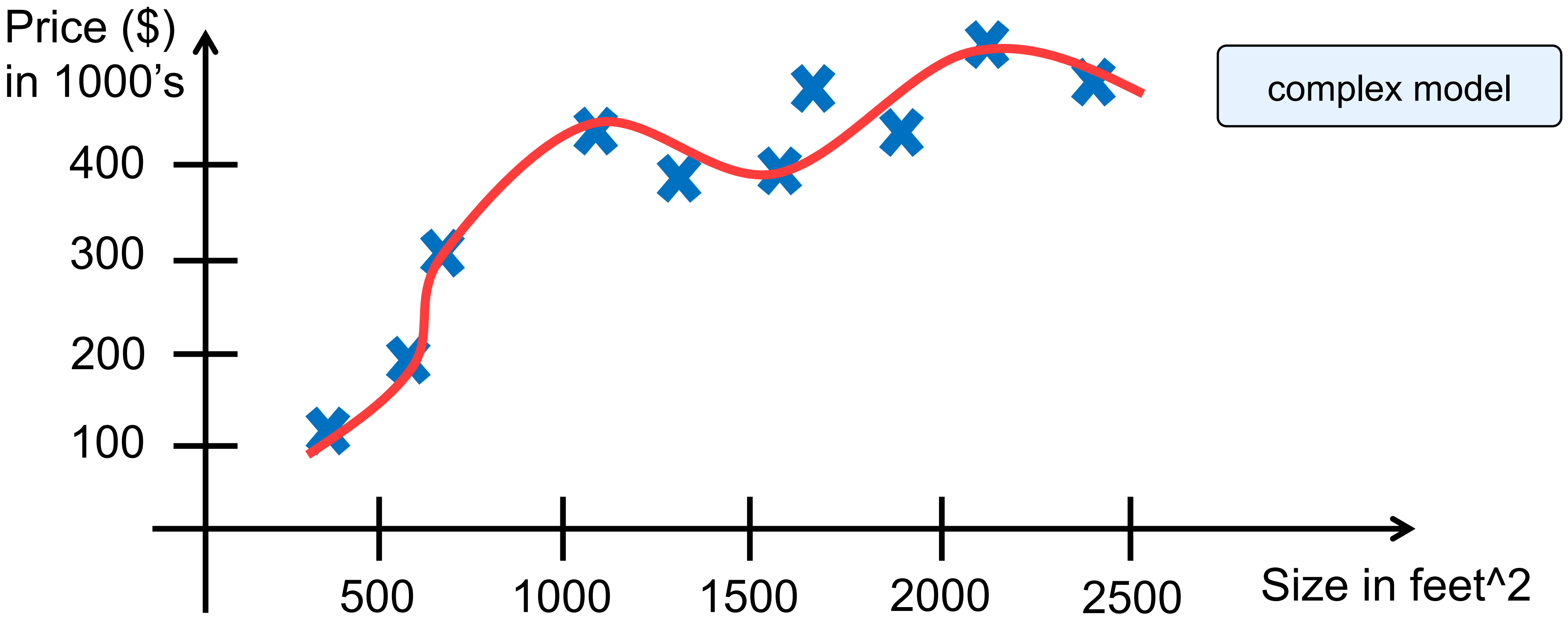
# Training Set

Visualize the training set as a scatter plot.



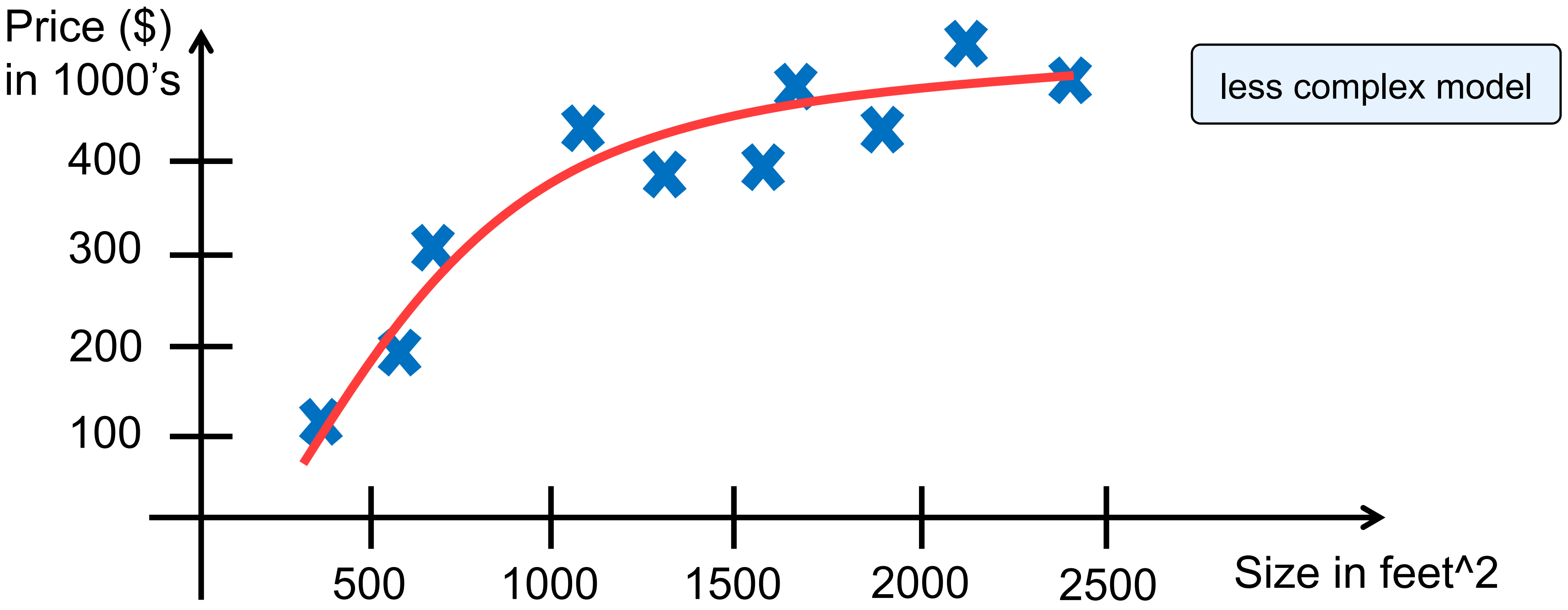
# Hypothesis

Hypothesis is a **candidate model** that approximates a target function for mapping examples of inputs to outputs.



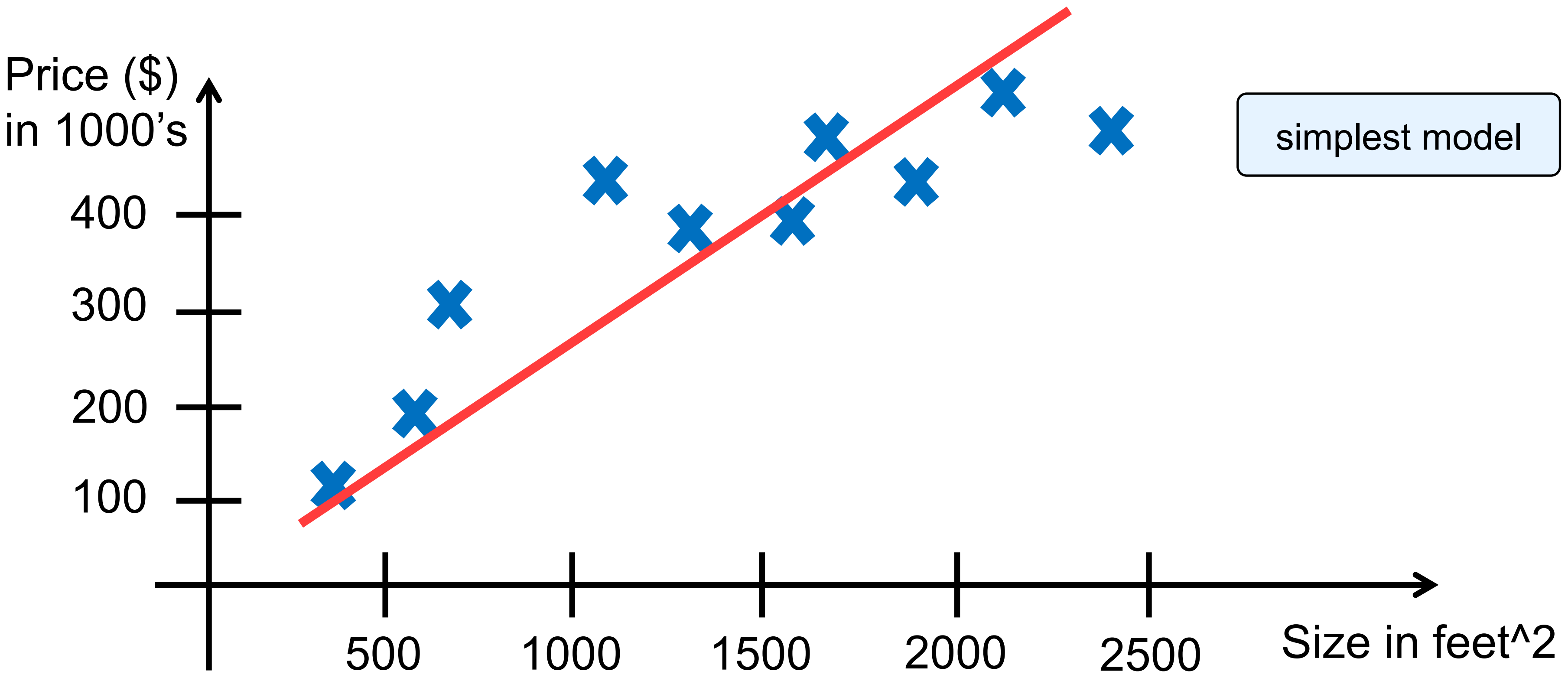
# Hypothesis

Hypothesis is a **candidate model** that approximates a target function for mapping examples of inputs to outputs.



# Hypothesis

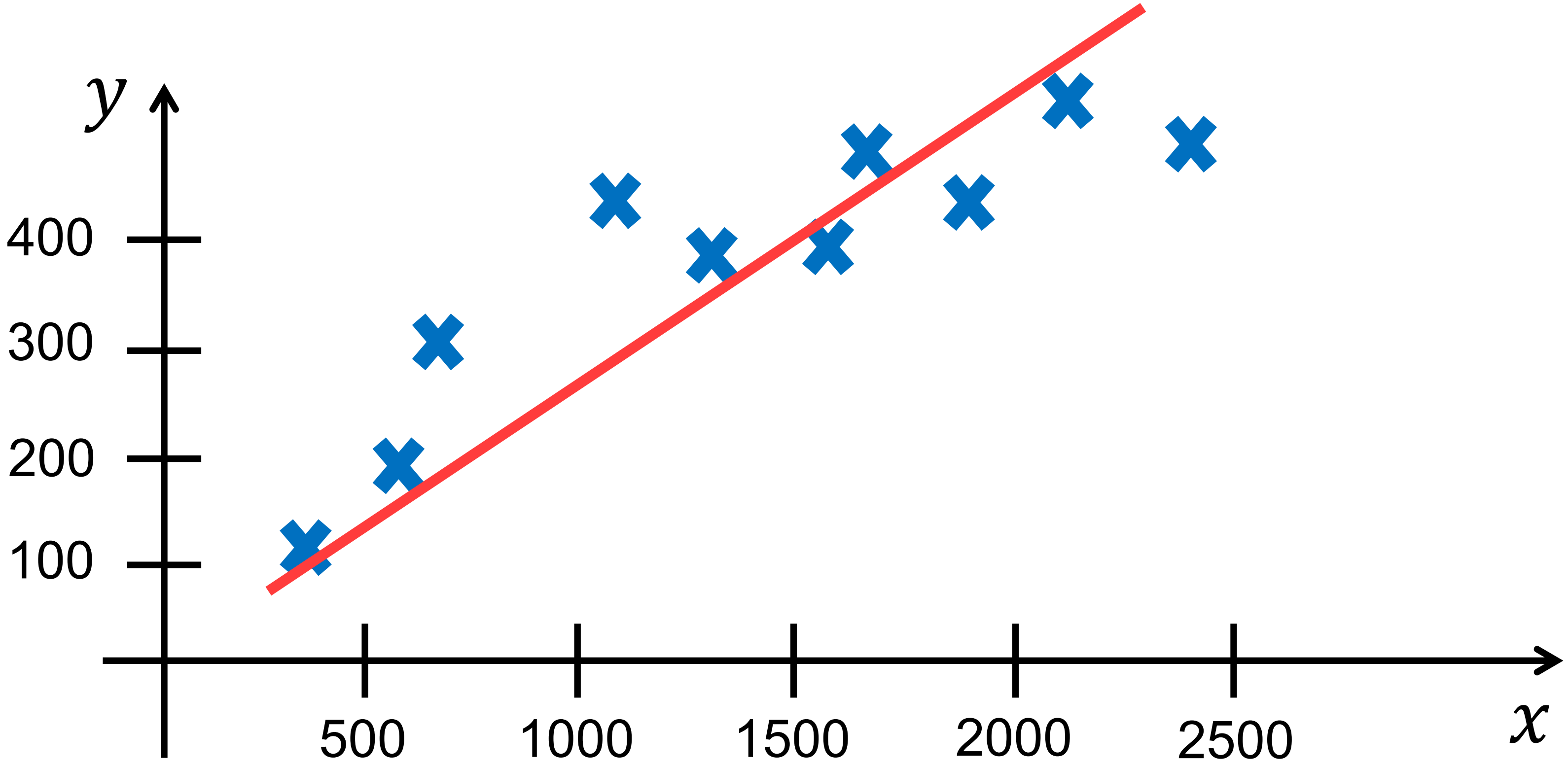
Hypothesis is a **candidate model** that approximates a target function for mapping examples of inputs to outputs.



# Hypothesis in Linear Regression

$$y = h_\beta(x) = \beta_0 + \beta_1 x$$

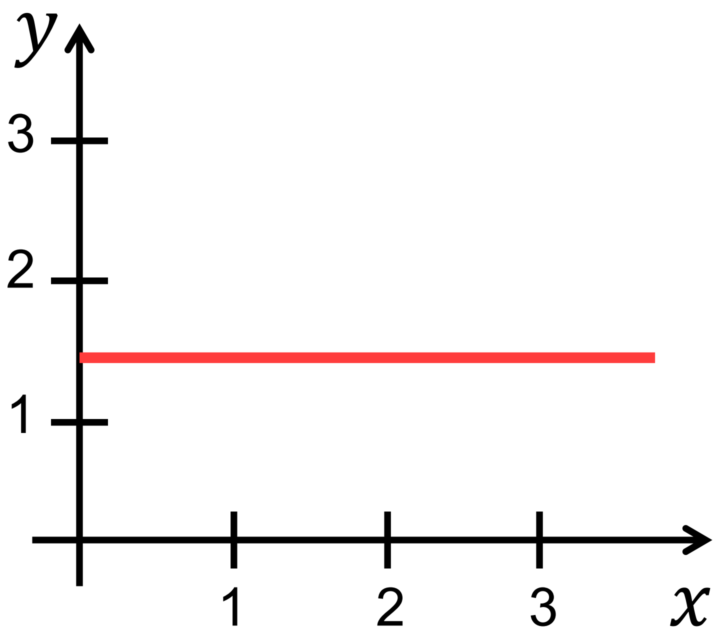
$\beta_0$ and $\beta_1$ are unknown parameters of the hypothesis



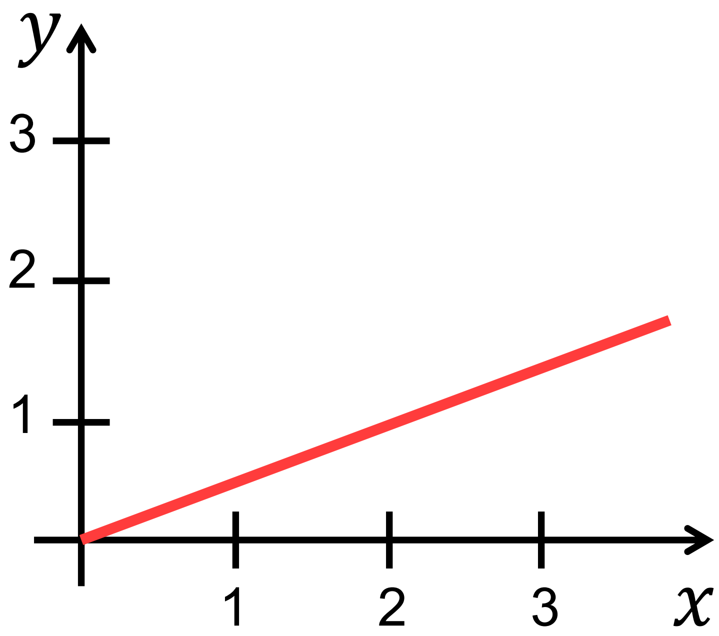
# Hypothesis in Linear Regression

$$y = h_\beta(x) = \beta_0 + \beta_1 x$$

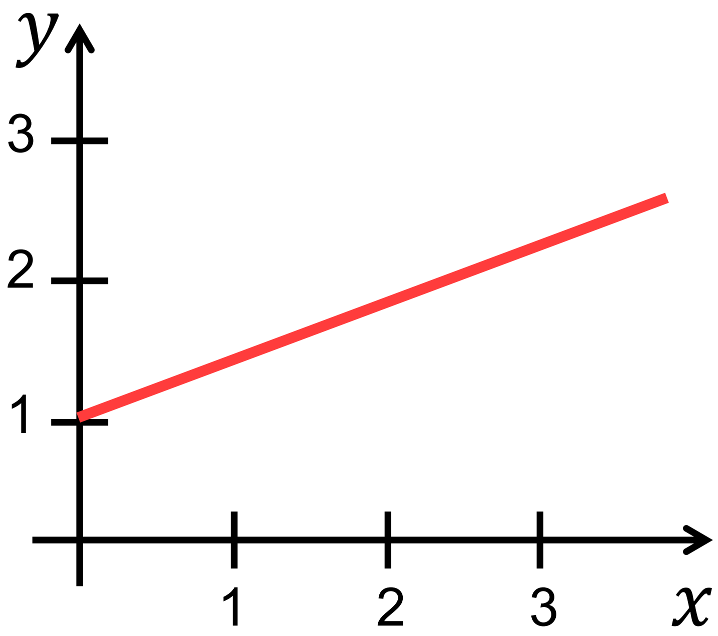
How to choose $\beta_0$ and $\beta_1$?




$\beta_0 = 1.5$
$\beta_1 = 0$



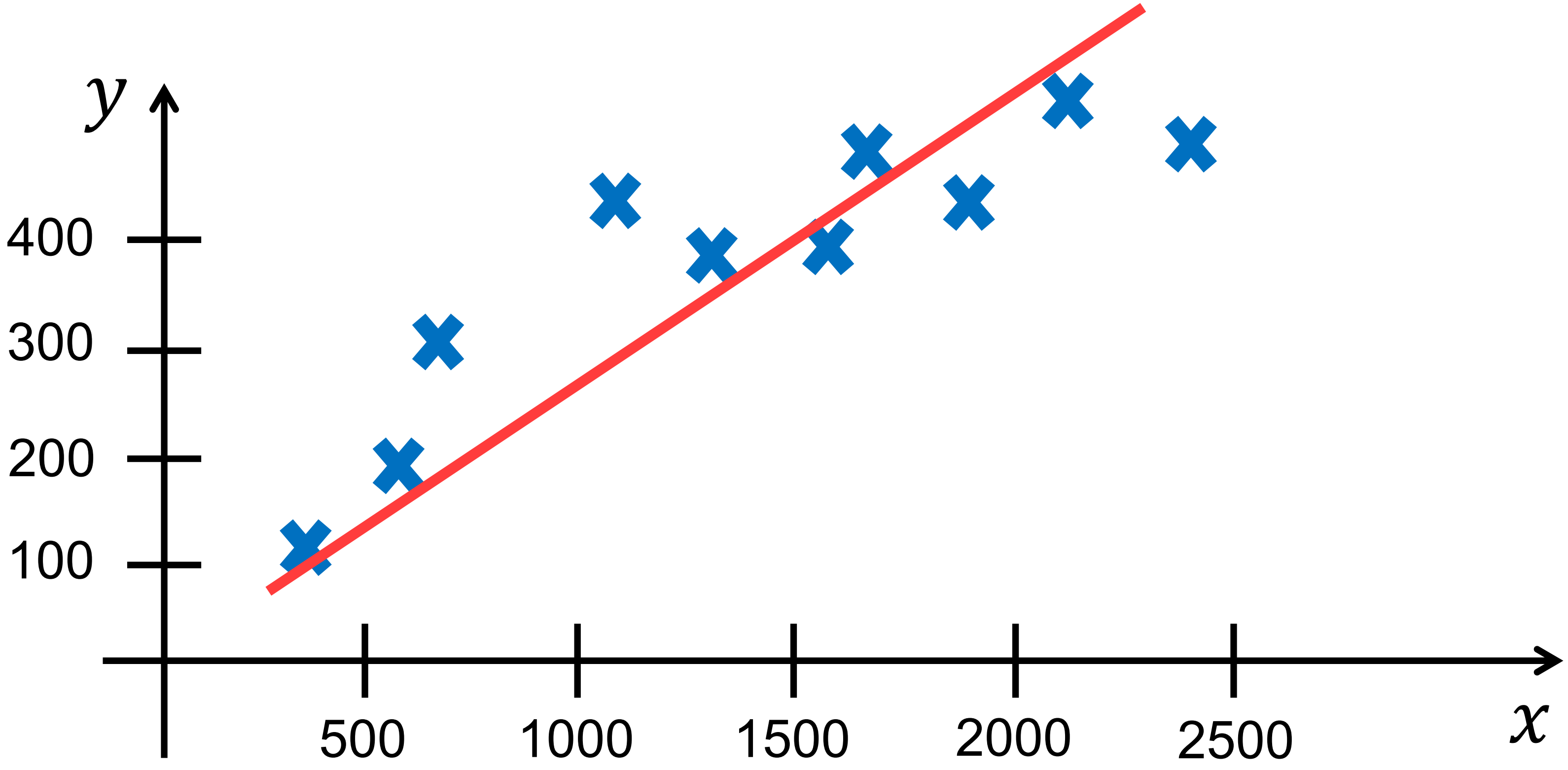

$\beta_0 = 0$
$\beta_1 = 0.5$




$\beta_0 = 1$
$\beta_1 = 0.5$

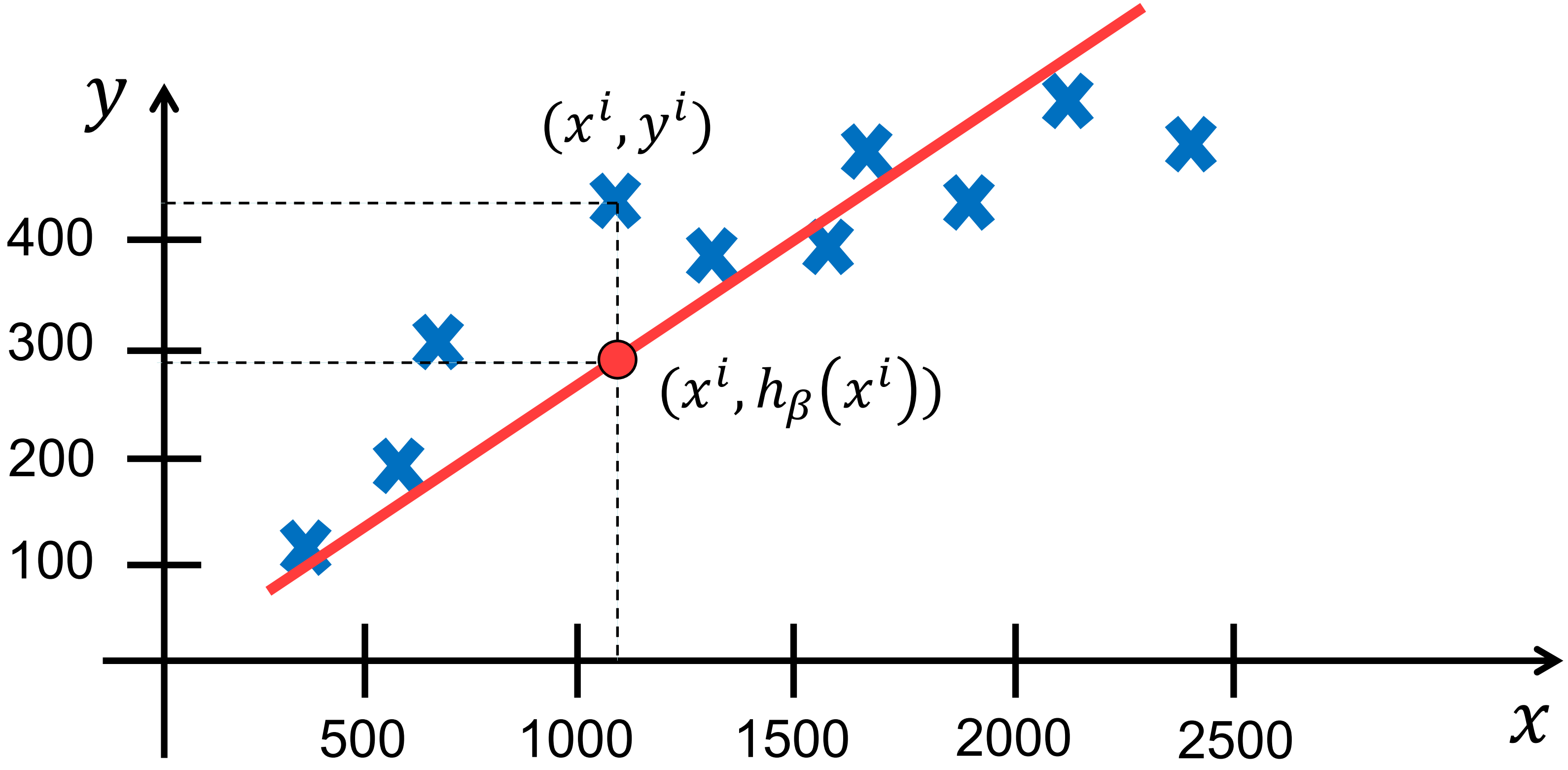
# Cost function

**Idea**: Choose $\beta_0, \beta_1$ so that our predicted value $h_\beta(x^i)$ is close to the observed value $y^i$ for our training examples $\{(x^i, y^i)\}$



# Cost function

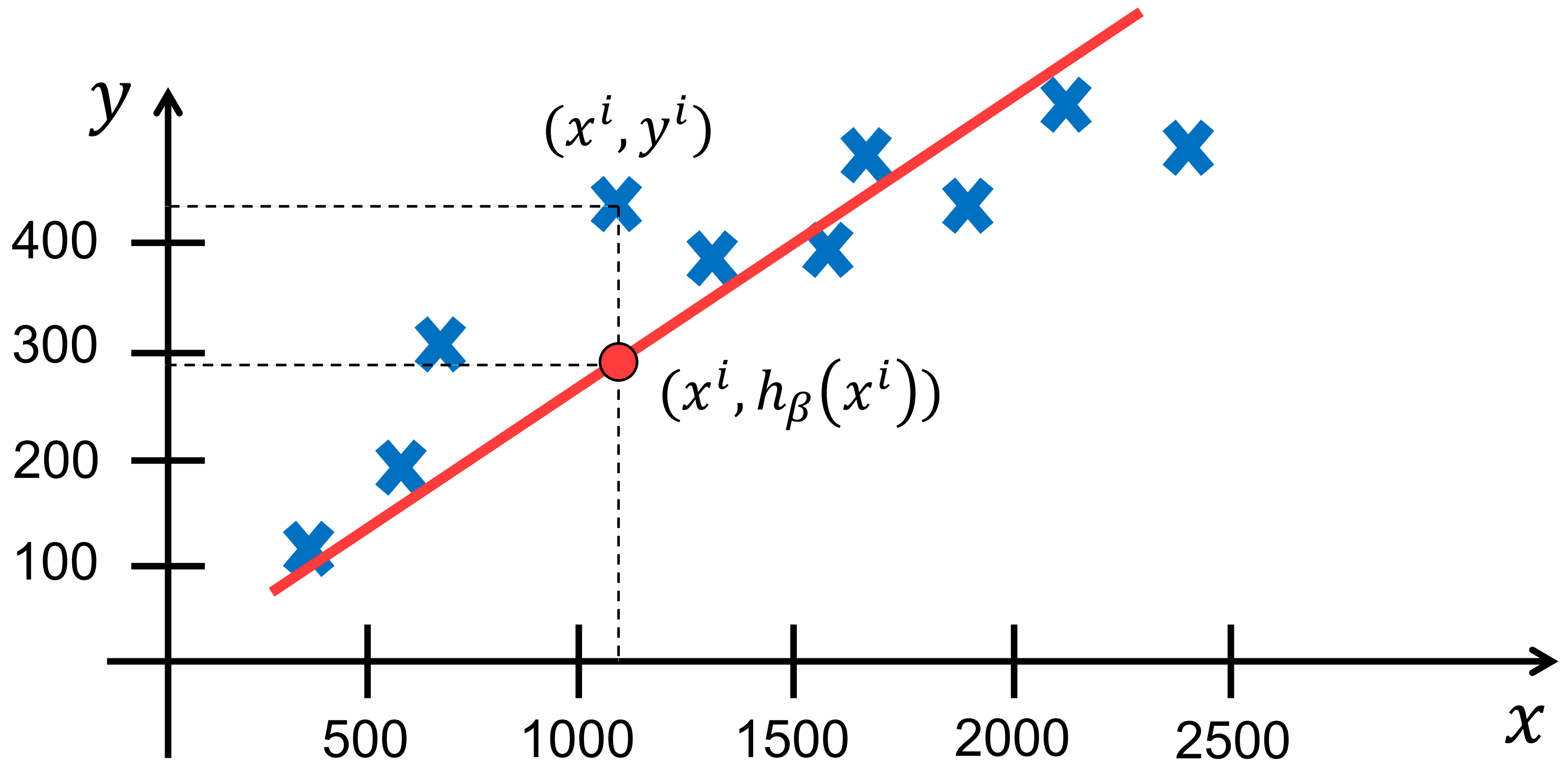
**Idea**: Choose $\beta_0, \beta_1$ so that our predicted value $h_\beta(x^i)$ is close to the observed value $y^i$ for our training examples $\{(x^i, y^i)\}$



# Cost function

$$\underset{\beta_0, \beta_1}{\text{minimize}} \; \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta\left(x^i\right) - y^i\right)^2$$

where $h_\beta\left(x^i\right) = \beta_0 + \beta_1 x^i$



# Cost function

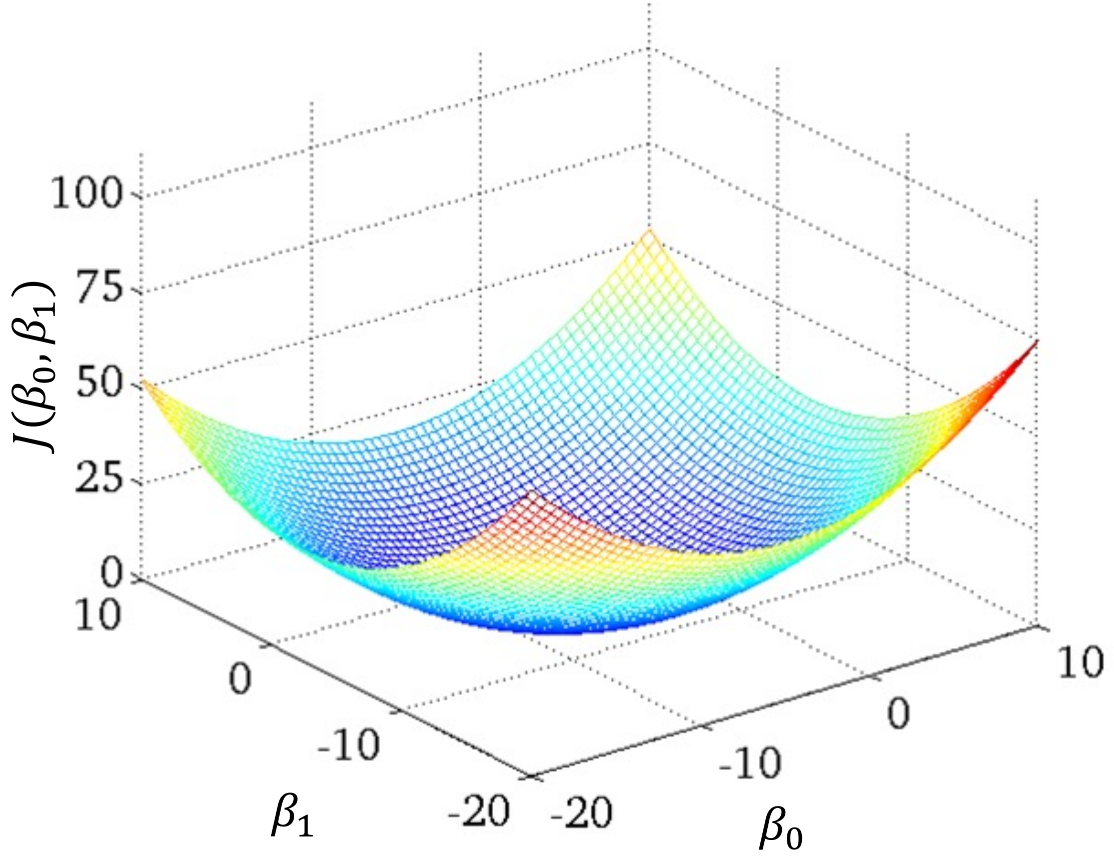
$$\underset{\beta_0, \beta_1}{\text{minimize}} \boxed{\frac{1}{2m}\sum_{i=1}^{m}\left(h_\beta\left(x^i\right) - y^i\right)^2}$$

where $h_\beta\left(x^i\right) = \beta_0 + \beta_1 x^i$

$$J(\beta_0, \beta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\beta\left(x^i\right) - y^i\right)^2$$

**Cost function**

$$\underset{\beta_0, \beta_1}{\text{minimize}} \boxed{J(\beta_0, \beta_1)}$$



# Cost Function

- **Hypothesis:** $h_\beta(x) = \beta_0 + \beta_1 x$

- **Parameters:** $\beta_0, \beta_1$

- **Cost function:** $J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\beta\left(x^i\right) - y^i \right)^2$

- **Goal**: $\underset{\beta_0, \beta_1}{\text{minimize}} \; J(\beta_0, \beta_1)$

# Simplified Hypothesis

- **Hypothesis:**

$$h_\beta(x) = \beta_0 + \beta_1 x$$

- **Parameters:**

$\beta_0, \beta_1$

- **Cost function:**

$$J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)^2$$

- **Goal**:

$$\underset{\beta_0, \beta_1}{\text{minimize}} \; J(\beta_0, \beta_1)$$

- **Hypothesis:**

$$h_\beta(x) = \beta_1 x \qquad \beta_0 = 0$$

- **Parameters:**

$\beta_1$

- **Cost function:**

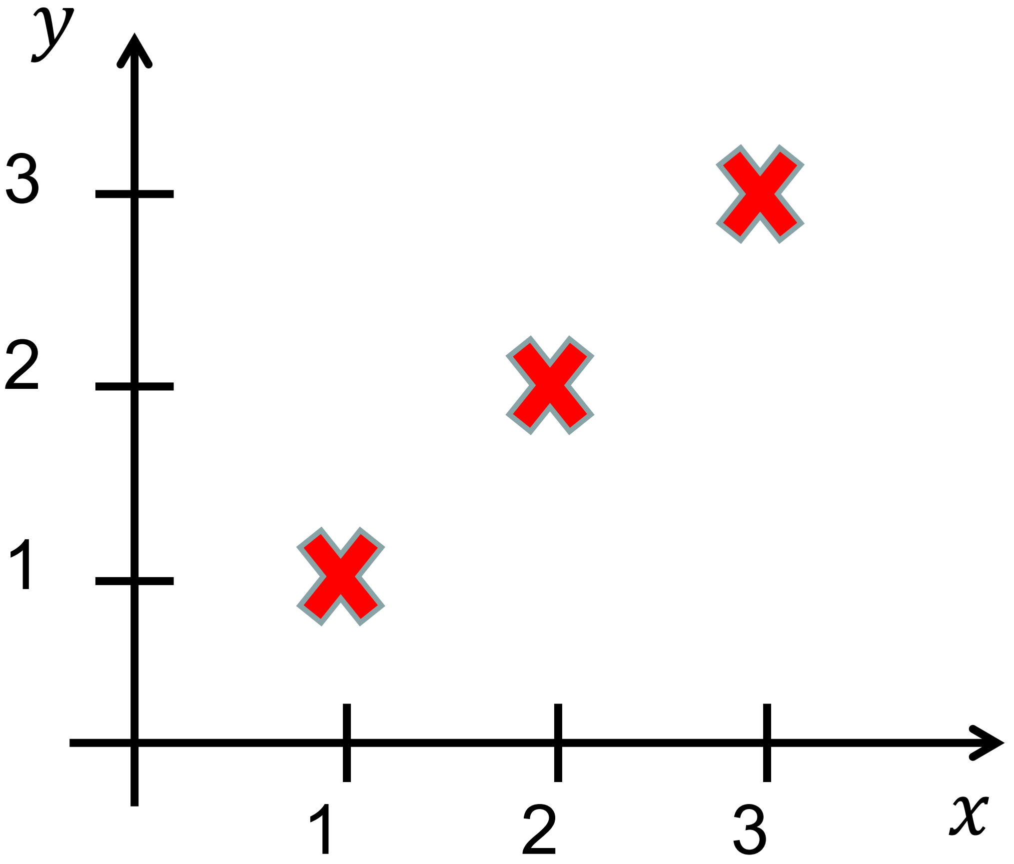
$$J(\beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)^2$$

- **Goal**:

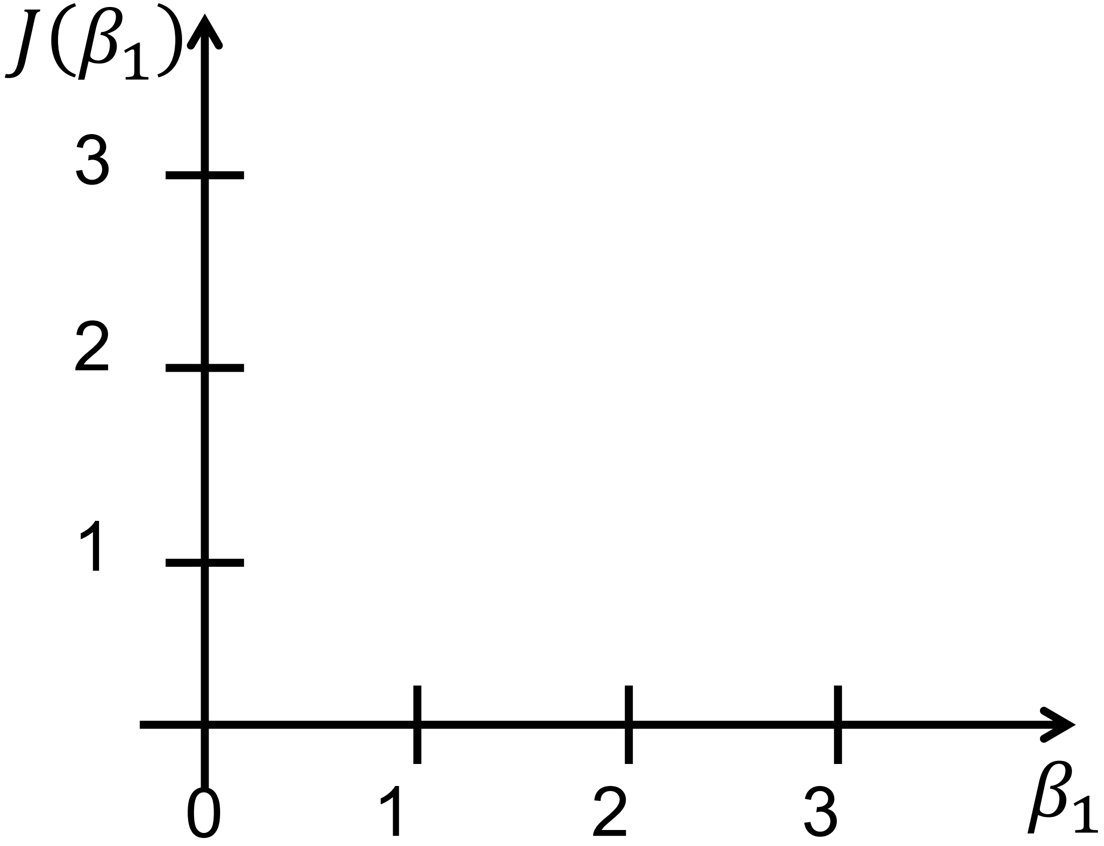
$$\underset{\beta_1}{\text{minimize}} \; J(\beta_1)$$

# Simplified Hypothesis

$h_\beta(x)$, function of $x$



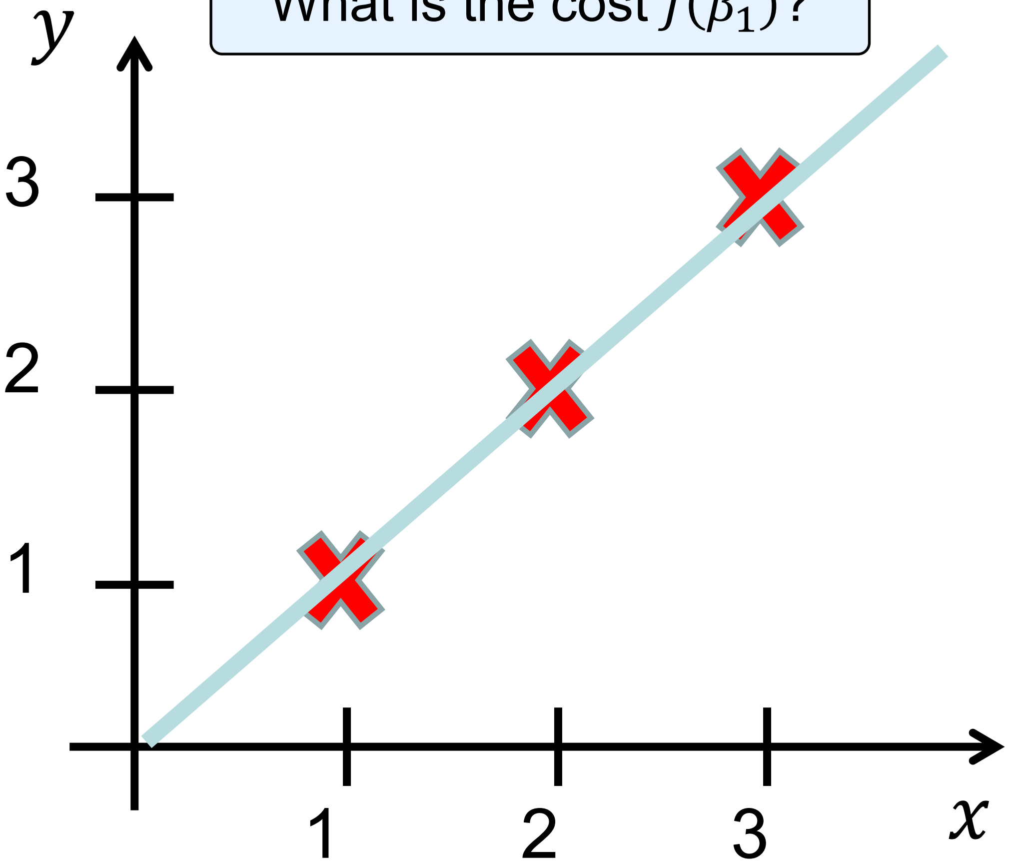

$J(\beta_1)$, function of $\beta_1$



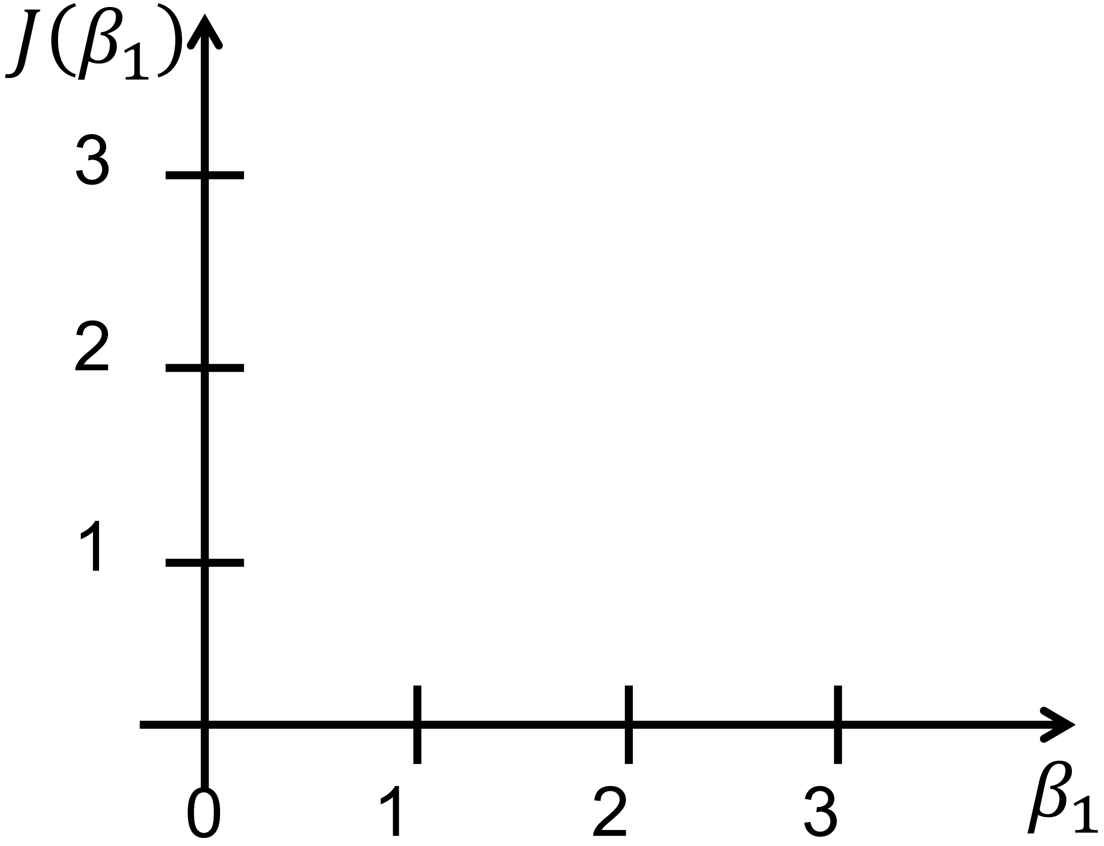
# Simplified Hypothesis

## $h_\beta(x)$, function of $x$

What is the cost $J(\beta_1)$?

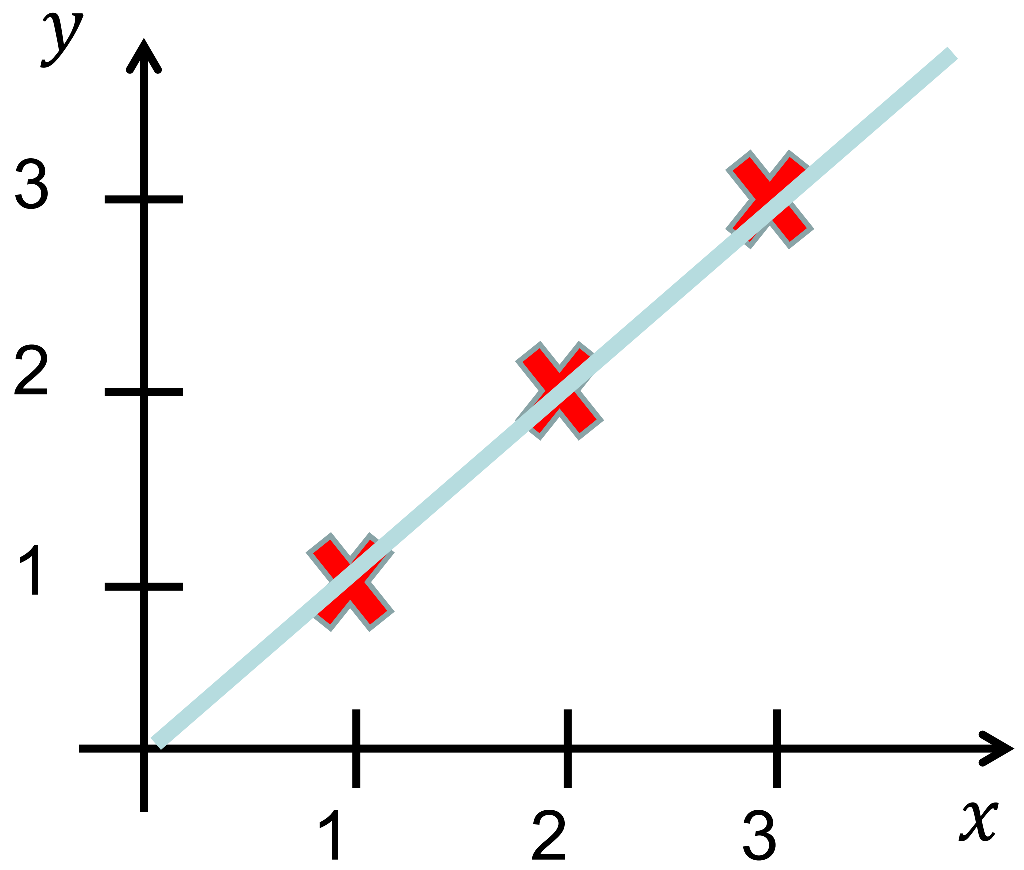



## $J(\beta_1)$, function of $\beta_1$

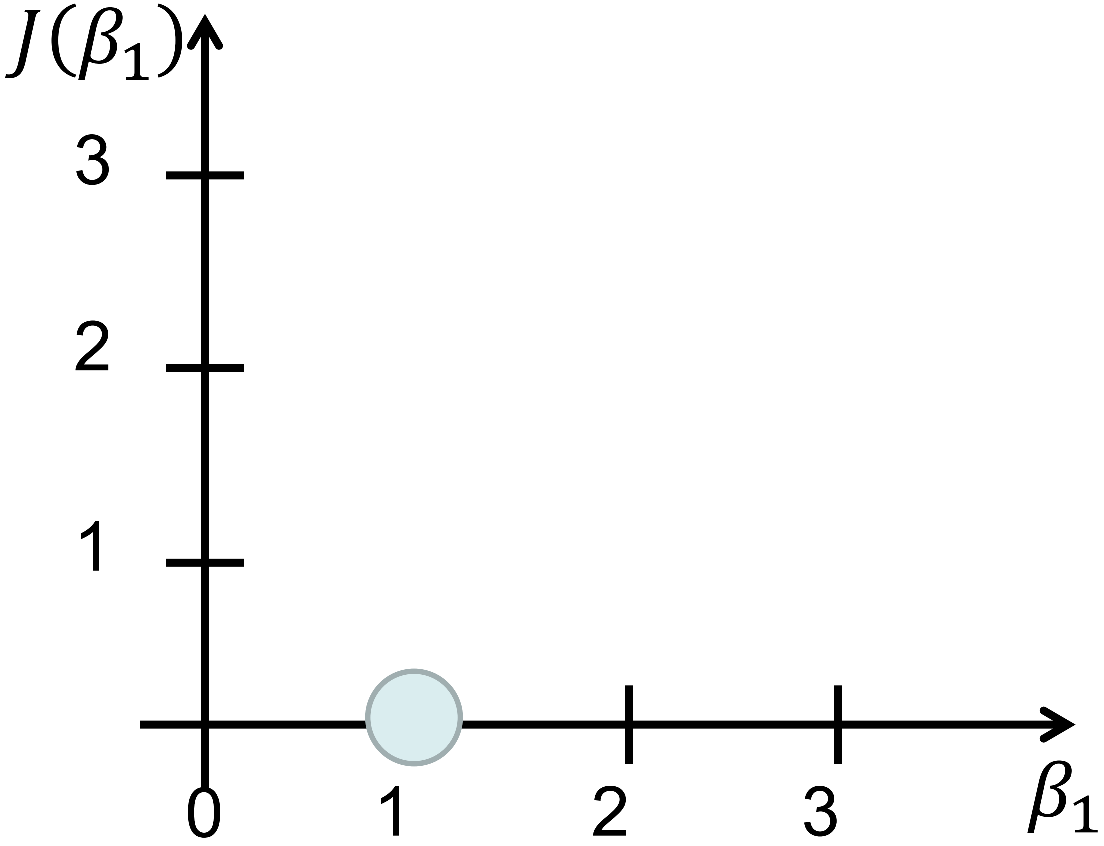


# Simplified Hypothesis

$h_\beta(x)$, function of $x$



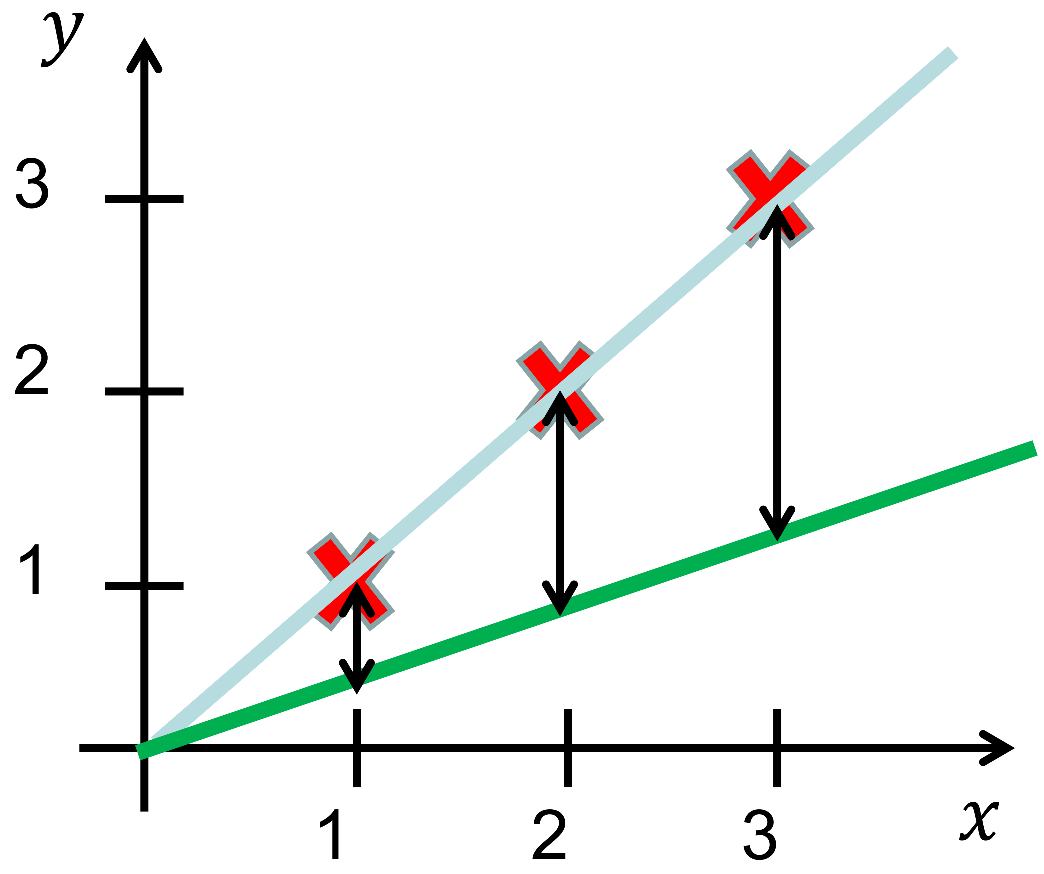

$J(\beta_1)$, function of $\beta_1$

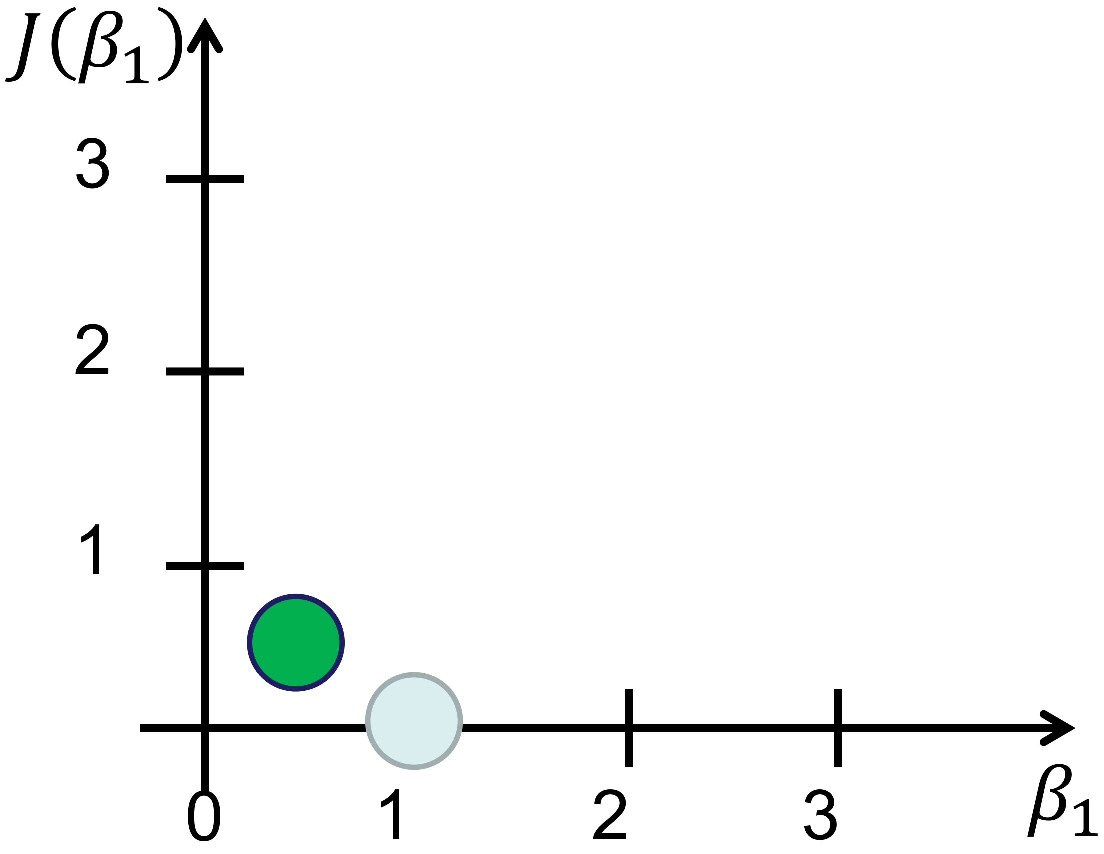


# Simplified Hypothesis

$h_\beta(x)$, function of $x$

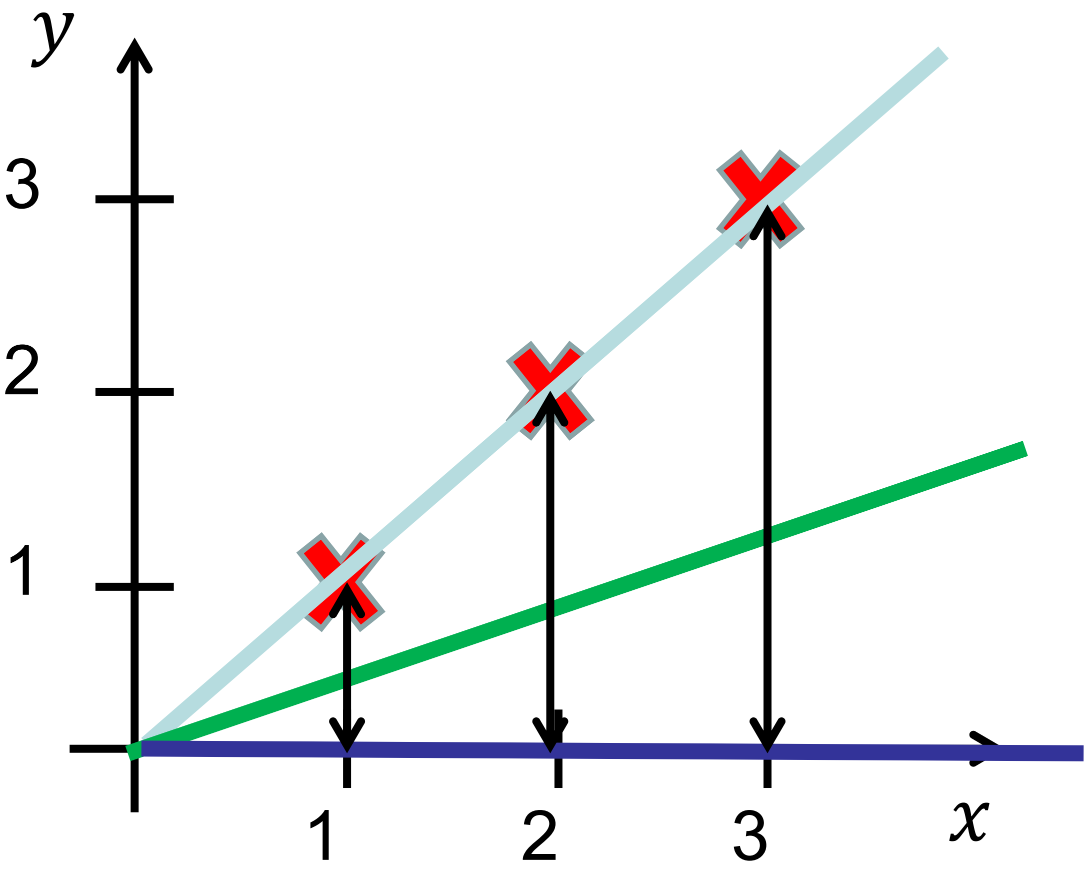
$J(\beta_1)$, function of $\beta_1$



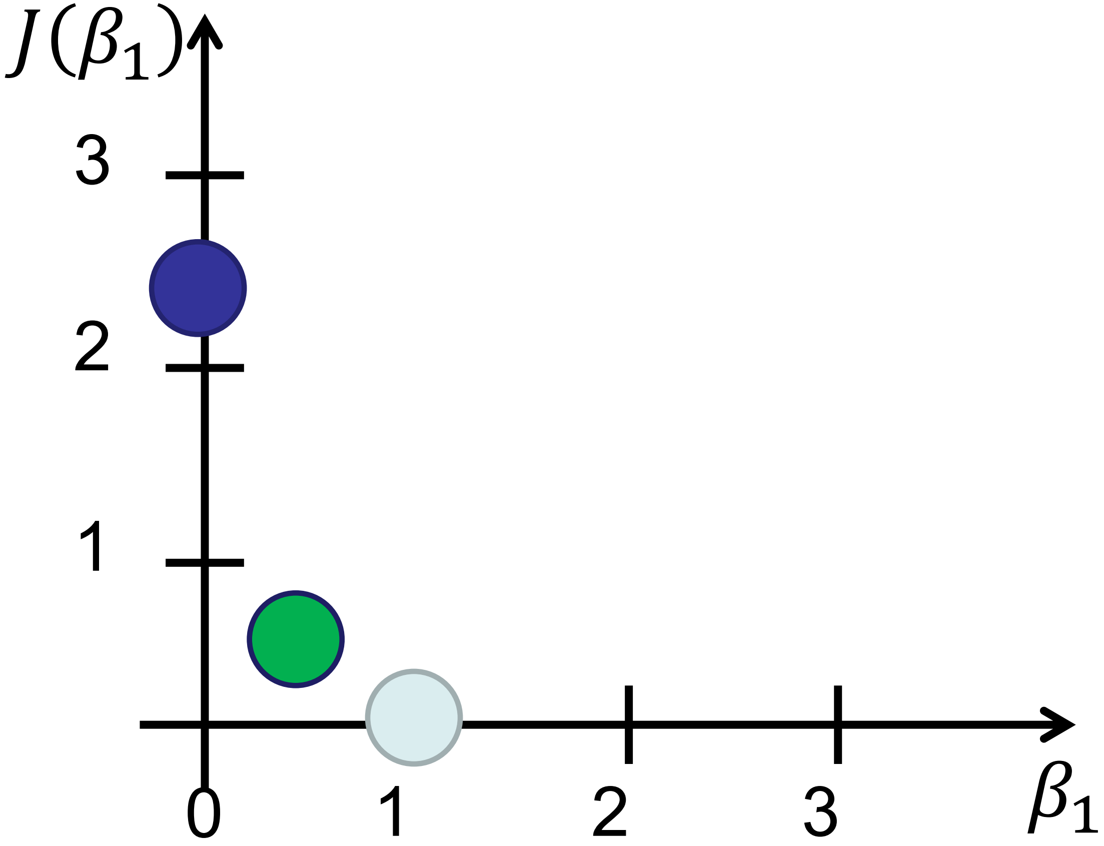



# Simplified Hypothesis

$h_\beta(x)$, function of $x$

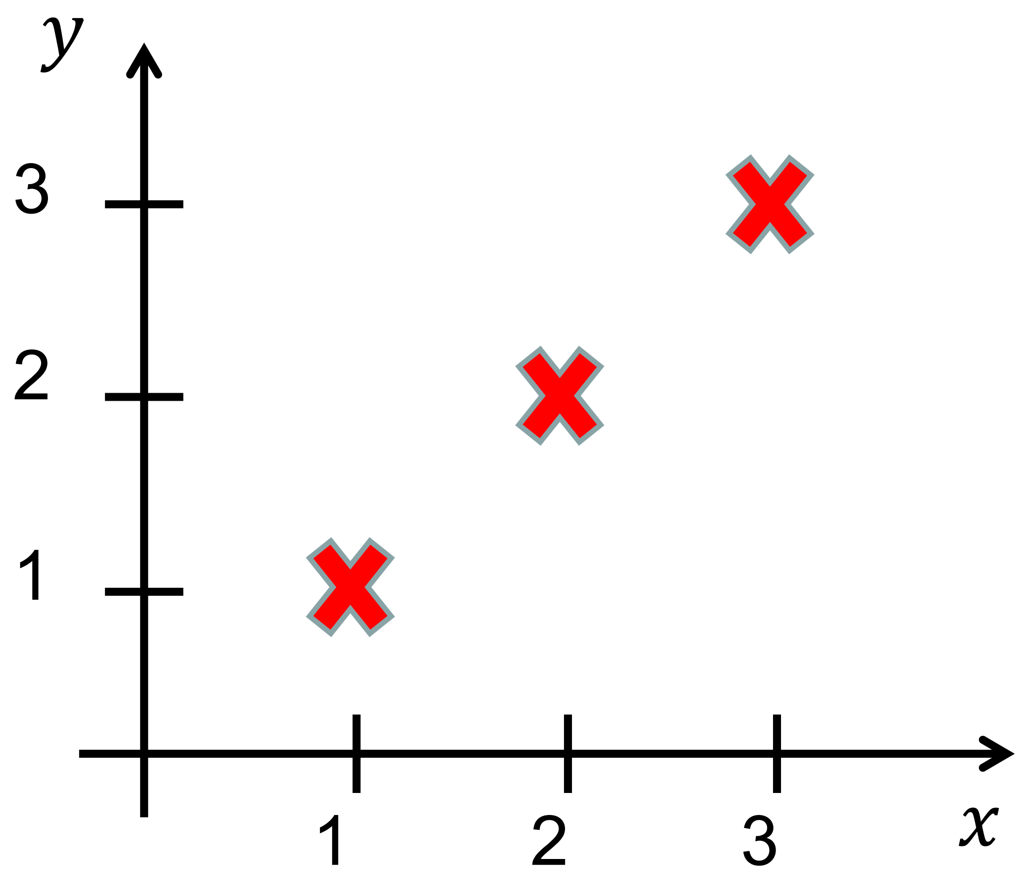



$J(\beta_1)$, function of $\beta_1$

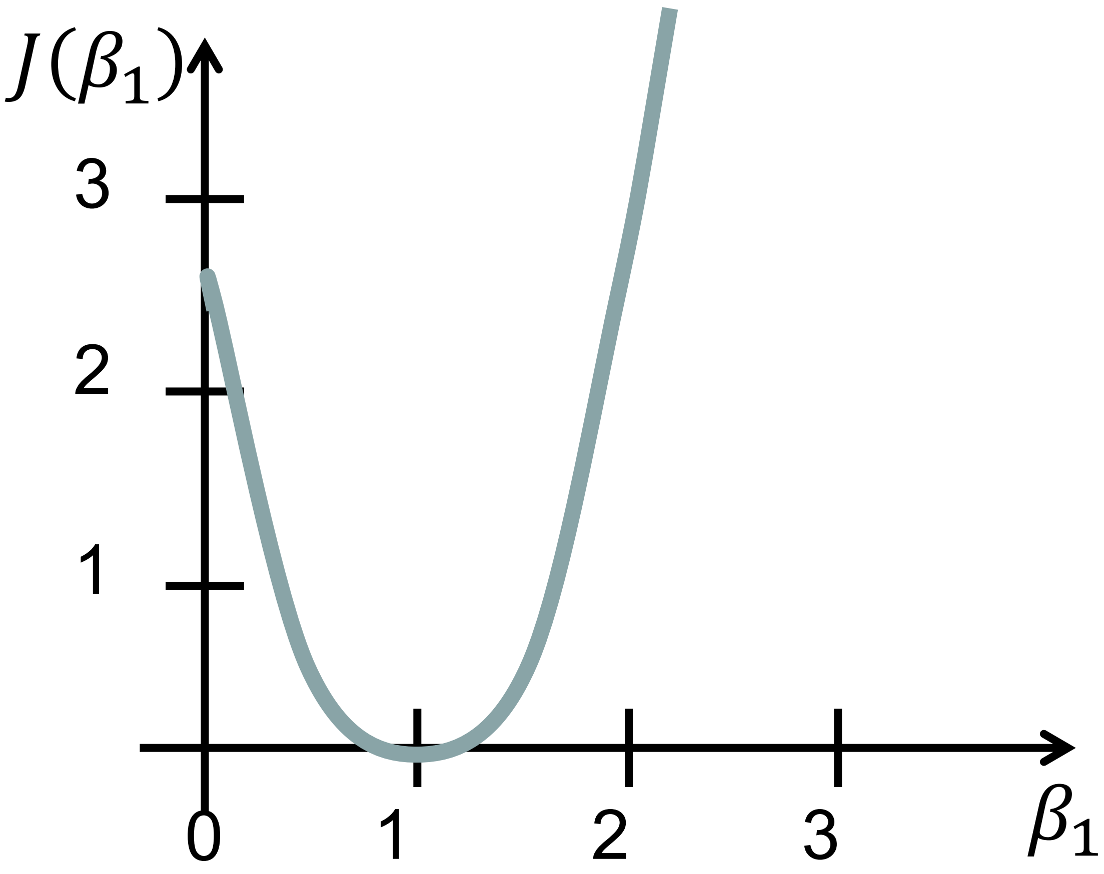


# Simplified Hypothesis

$h_\beta(x)$, function of $x$



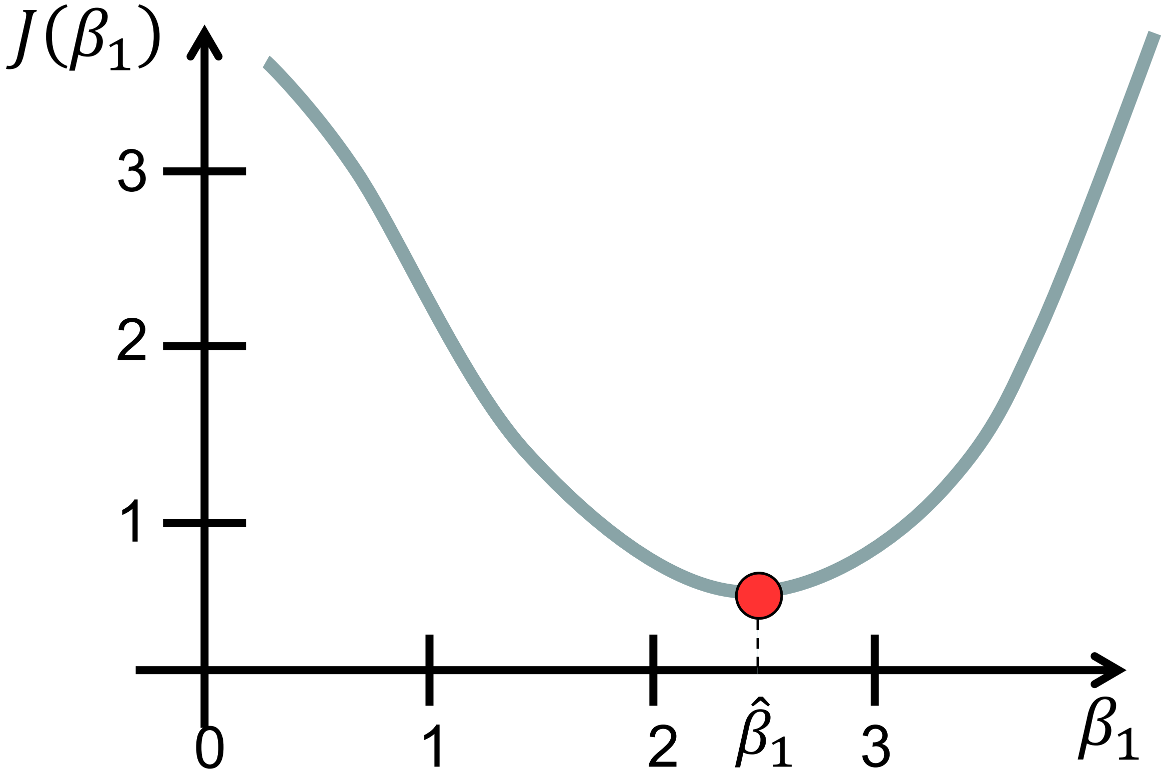

$J(\beta_1)$, function of $\beta_1$



# Simplified Hypothesis

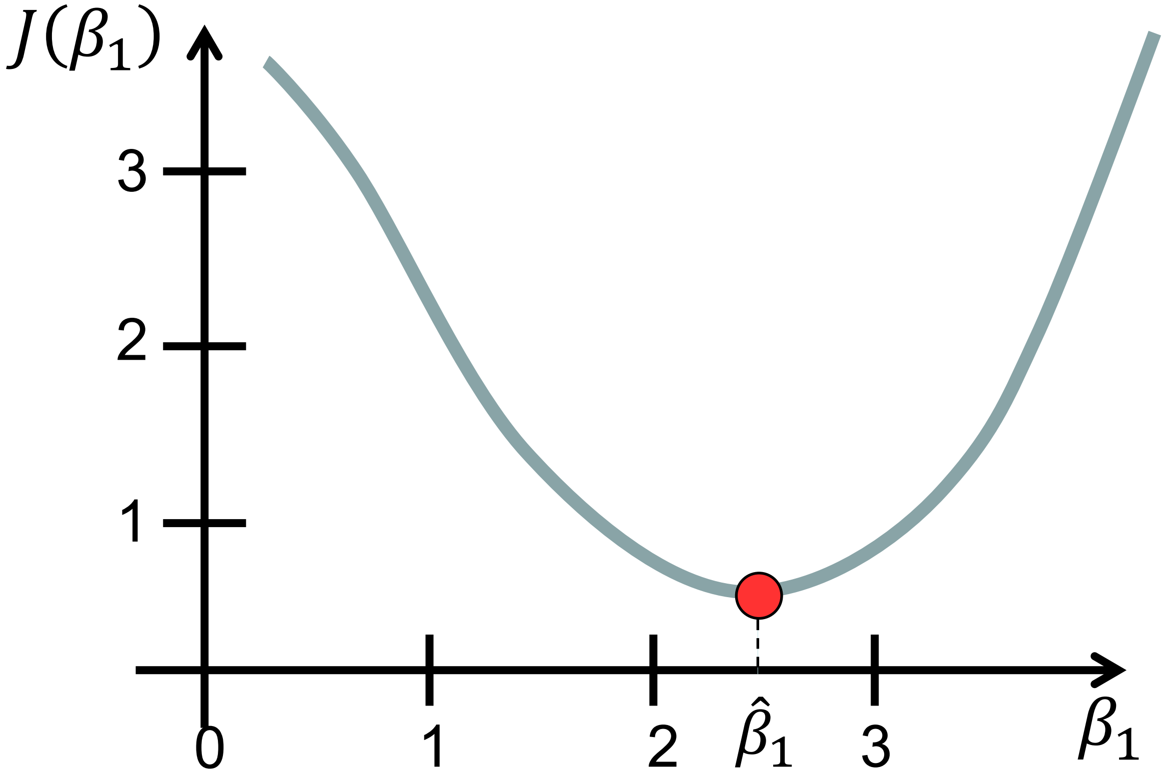
$$\underset{\beta_1}{\text{Minimize}}\ J(\beta_1)$$

How to solve the problem?



# Gradient Descent

Gradient descent is an iterative optimization algorithm for finding a **local minimum** of a **differentiable** function.



# Gradient Descent

Start with some $\beta_1$

Repeat until convergence

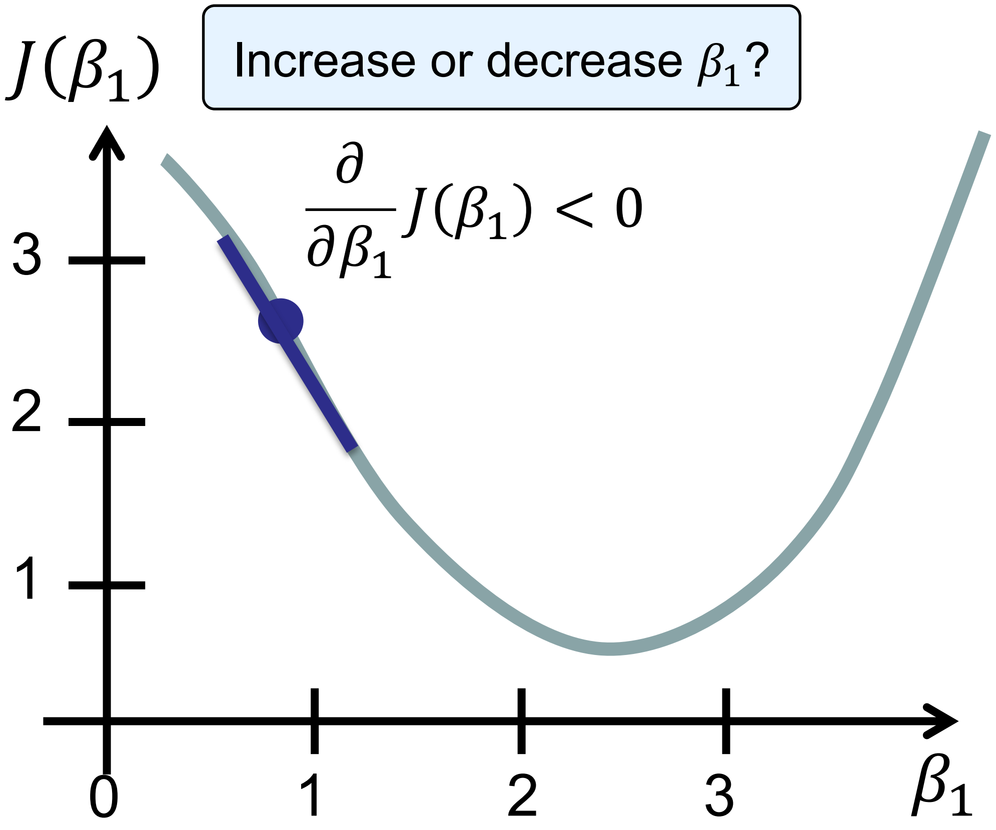
{

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_1)$$

}

$\alpha$: Learning rate (step size)

$\frac{\partial}{\partial \beta_1} J(\beta_1)$: derivative



# Gradient Descent

Start with some $\beta_1$

Repeat until convergence

{

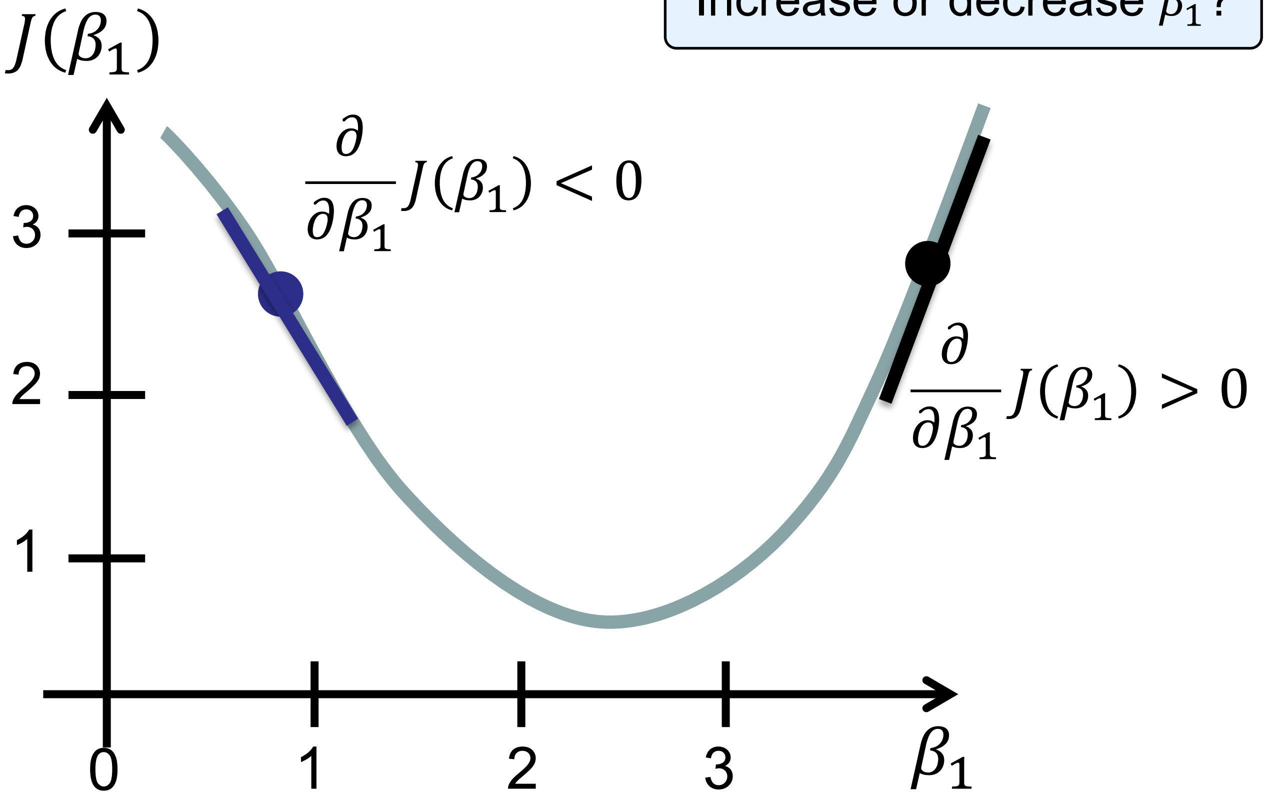
$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_1)$$

}

$\alpha$: Learning rate (step size)

$\frac{\partial}{\partial \beta_1} J(\beta_1)$: derivative

Increase or decrease $\beta_1$?

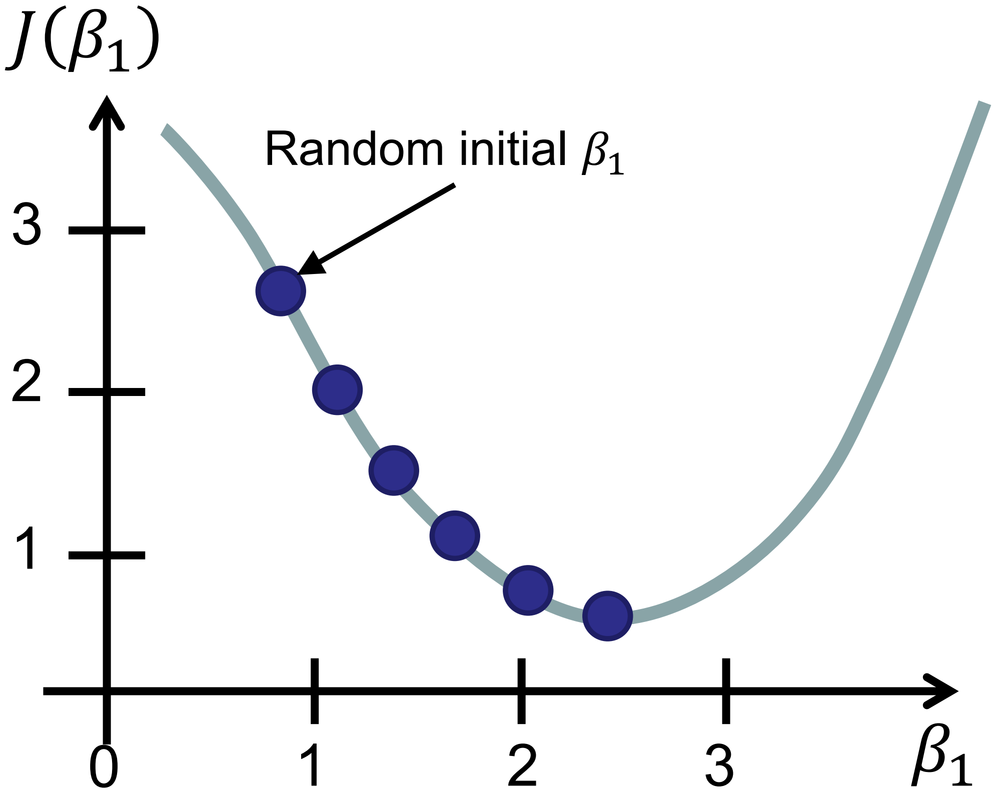


# Gradient Descent

Start with some $\beta_1$

Repeat until convergence

$$\left\{ \quad \beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_1) \quad \right\}$$




$\alpha$: Learning rate (step size)

$\frac{\partial}{\partial \beta_1} J(\beta_1)$: derivative

# Gradient Descent

Start with some $\beta_0, \ \beta_1$

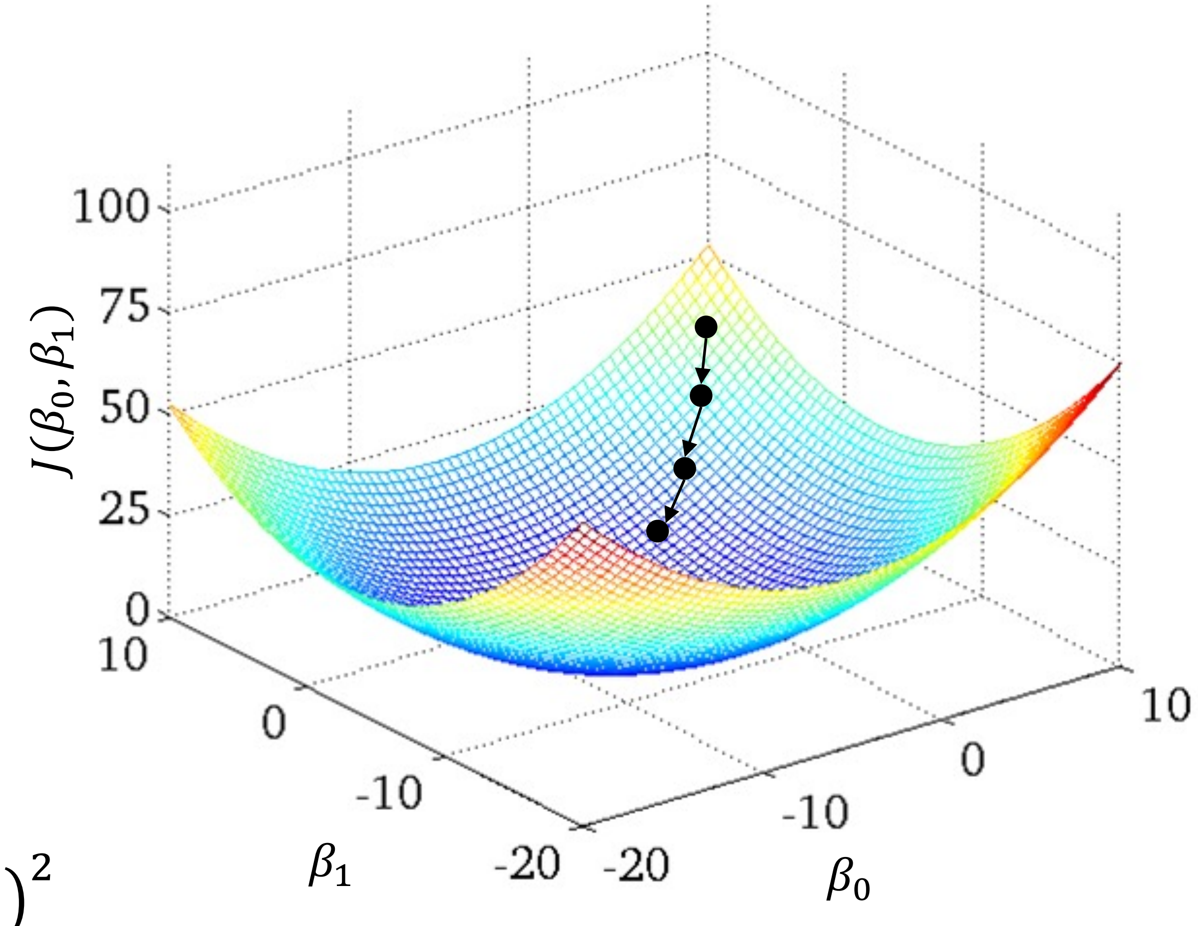
Repeat until convergence

{

$$\beta_0 := \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$$

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$$

}




Linear regression model

Cost function: $J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)^2$

Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x$

# Gradient Descent

Start with some $\beta_0, \ \beta_1$

Repeat until convergence

{

$$\beta_0 := \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$$

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$$

}

How to compute the gradient $\frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$, $\frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$ (i.e., partial derivative)?

Linear regression model

Cost function: $J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta\left(x^i\right) - y^i\right)^2$

Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x$

# Computing Partial Derivative

- Differentiate equation $J(\beta_0, \beta_1)$ with respect to $\beta_0$

$$\frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_0} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)^2$$

$$= \frac{\partial}{\partial \beta_0} \frac{1}{2m} \sum_{i=1}^{m} \left(\beta_0 + \beta_1 x^i - y^i\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(\beta_0 + \beta_1 x^i - y^i\right)$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)$$

# Computing Partial Derivative

- Differentiate equation $J(\beta_0, \beta_1)$ with respect to $\beta_1$

$$\frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right)^2$$

$$= \frac{\partial}{\partial \beta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(\beta_0 + \beta_1 x^i - y^i\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(\beta_0 + \beta_1 x^i - y^i\right) x^i$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta(x^i) - y^i\right) x^i$$

# Gradient Descent for Linear Regression

Start with some $\beta_0, \ \beta_1$

Repeat until convergence

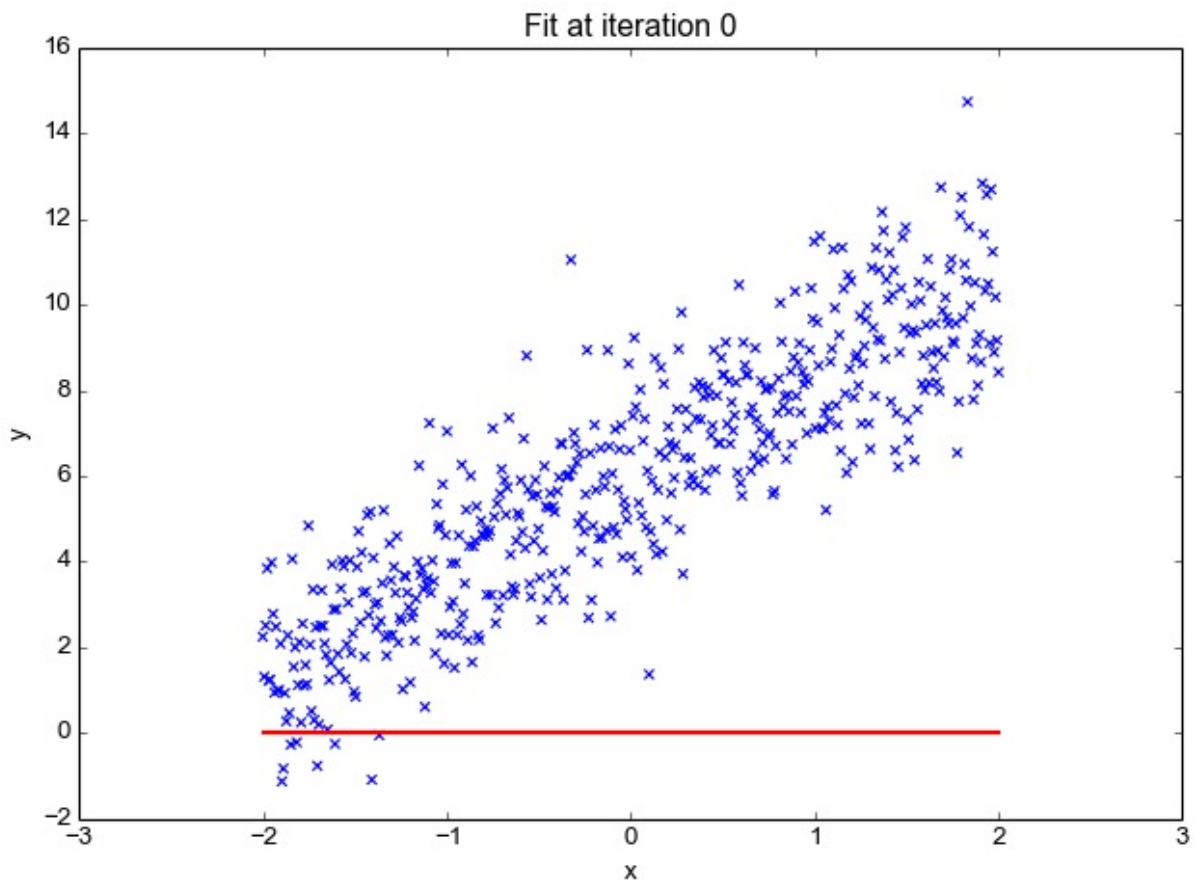
{

$$\beta_0 := \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$$

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$$

}

Note: update $\beta_0$ and $\beta_1$ simultaneously



# Linear Regression

- Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x$
- Cost function: $J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta\left(x^i\right) - y^i\right)^2$
- Gradient Descent:

$$\beta_0 := \beta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta\left(x^i\right) - y^i\right)$$

$$\beta_1 := \beta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta\left(x^i\right) - y^i\right) x^i$$

Can we write these equations in a more compact form?

# Hypothesis

Hypothesis: $h_\beta(x) = \beta_0 + \beta_1 x$

$$= \begin{bmatrix} 1 & x \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$= \boldsymbol{x} \times \boldsymbol{b}$$

where $\boldsymbol{x} = \begin{bmatrix} 1 & x \end{bmatrix}$ $\quad \boldsymbol{b} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$

# Hypothesis

For $m$ data points, $\hat{y}^i = h_\beta(x^i) = \beta_0 + \beta_1 x^i = \boldsymbol{x}^i \times \boldsymbol{b}$

$$\hat{y}^1 = \boldsymbol{x}^1 \boldsymbol{b}$$

$$\hat{y}^2 = \boldsymbol{x}^2 \boldsymbol{b}$$

$$\cdots$$

$$\hat{y}^m = \boldsymbol{x}^m \boldsymbol{b}$$

$$\Rightarrow \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \vdots \\ \hat{y}^m \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^1 \boldsymbol{b} \\ \boldsymbol{x}^2 \boldsymbol{b} \\ \vdots \\ \boldsymbol{x}^m \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \\ \vdots \\ \boldsymbol{x}^m \end{bmatrix} \boldsymbol{b} = \boldsymbol{X} \times \boldsymbol{b}$$

where $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \\ \vdots \\ \boldsymbol{x}^m \end{bmatrix} = \begin{bmatrix} 1 & x^1 \\ 1 & x^2 \\ \vdots & \vdots \\ 1 & x^m \end{bmatrix}$

# Cost Function

Cost function $J(\beta_0, \beta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}^i - y^i\right)^2$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}^i - y^i\right)\times\left(\hat{y}^i - y^i\right)$$

$$= \frac{1}{2m}\left[(\hat{y}^1 - y^1) \quad .. \quad (\hat{y}^m - y^m)\right]\times\begin{bmatrix}(\hat{y}^1 - y^1)\\ ..\\ (\hat{y}^m - y^m)\end{bmatrix}$$

$$= \frac{1}{2m}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}\times(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

where $\hat{\boldsymbol{y}} = \begin{bmatrix}\hat{y}^1\\ \hat{y}^2\\ ..\\ \hat{y}^m\end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix}y^1\\ y^2\\ ..\\ y^m\end{bmatrix}$

# Gradient Descent

$$\beta_0 := \beta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left( \hat{y}^i - y^i \right)$$

$$\beta_1 := \beta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left( \hat{y}^i - y^i \right) x^i$$

$$\Rightarrow \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} - \alpha \frac{1}{m} \begin{bmatrix} \sum_{i=1}^{m} \left( \hat{y}^i - y^i \right) \\ \sum_{i=1}^{m} \left( \hat{y}^i - y^i \right) x^i \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} - \alpha \frac{1}{m} \begin{bmatrix} 1 & .. & 1 \\ x^1 & .. & x^m \end{bmatrix} \begin{bmatrix} (\hat{y}^1 - y^1) \\ .. \\ (\hat{y}^m - y^m) \end{bmatrix}$$

$$\Leftarrow$$

$$\boldsymbol{b} = \boldsymbol{b} - \alpha \frac{1}{m} \boldsymbol{X}^{\mathrm{T}} \times (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$\Downarrow$$

$$\boldsymbol{b} = \boldsymbol{b} - \alpha \frac{1}{m} \boldsymbol{X}^{\mathrm{T}} \times (\boldsymbol{X} \times \boldsymbol{b} - \boldsymbol{y})$$

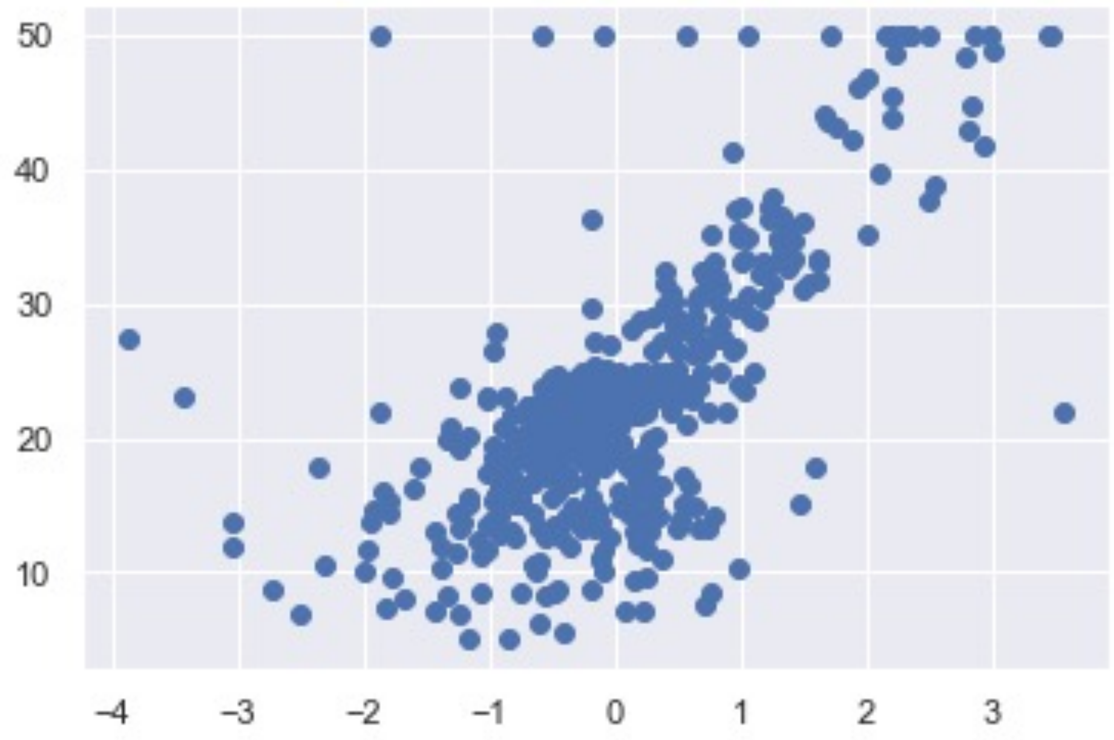
# Linear Regression

- Hypothesis: $\hat{\boldsymbol{y}} = \boldsymbol{X} \times \boldsymbol{b}$
- Cost Function: $J(\beta_0, \beta_1) = \frac{1}{2m} (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}} \times (\hat{\boldsymbol{y}} - \boldsymbol{y})$
- Gradient Descent: $\boldsymbol{b} = \boldsymbol{b} - \alpha \frac{1}{m} \boldsymbol{X}^{\mathrm{T}} \times (\boldsymbol{X} \times \boldsymbol{b} - \boldsymbol{y})$

where $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \hat{y}^3 \\ .. \\ \hat{y}^m \end{bmatrix}$ $\quad \boldsymbol{X} = \begin{bmatrix} 1 & x^1 \\ 1 & x^2 \\ .. & .. \\ 1 & x^m \end{bmatrix}$ $\quad \boldsymbol{b} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$ $\quad \boldsymbol{y} = \begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ .. \\ y^m \end{bmatrix}$

# Cohort Problem CS0

**CS0.** *Plot:* Read data for Boston Housing Prices and write a function get_features_targets() to get the columns for the features and the targets from the input argument data frame.



# Cohort Problem CS1

**CS1.** *Cost Function:* Write `def compute_cost(X, y, beta)` to compute the cost function of a linear regression model.

| | |
|---|---|
| Cost Function: | $J(\beta_0, \beta_1) = \frac{1}{2m}(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}} \times (\hat{\boldsymbol{y}} - \boldsymbol{y})$ |
| Hypothesis: | $\hat{\boldsymbol{y}} = \boldsymbol{X} \times \boldsymbol{b}$ |

Note: $m$ is the number of data points (i.e., number of rows in $\boldsymbol{X}$)

# Cohort Problem CS2

**CS2.** *Gradient Descent:* Write a function called

```
def gradient_descent(X, y, beta, alpha, num_iters):
```

- `X`: is a 2-D numpy array for the features
- `y`: is a vector array for the target
- `beta`: is a column vector for the initial guess of the parameters
- `alpha`: is the learning rate
- `num_iters`: is the number of iteration to perform

Gradient Descent: $\boldsymbol{b} = \boldsymbol{b} - \alpha \frac{1}{m} \boldsymbol{X}^{\mathrm{T}} \times (\boldsymbol{X} \times \boldsymbol{b} - \boldsymbol{y})$

Thank You!