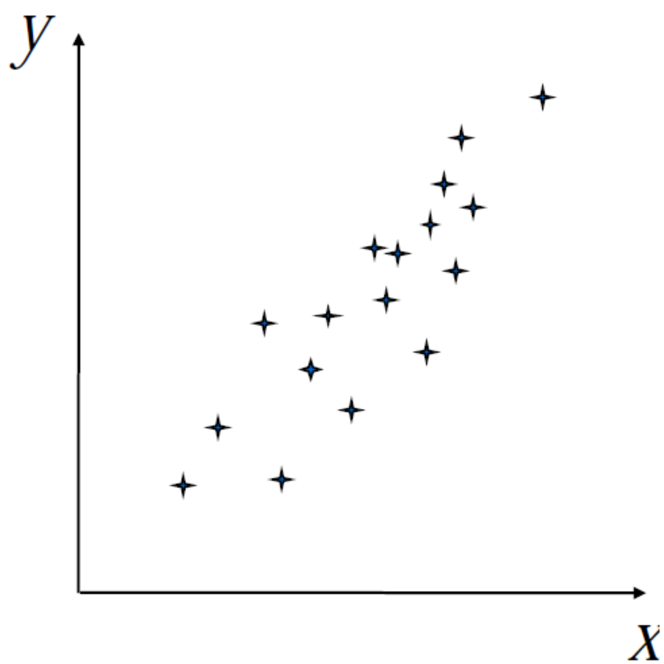
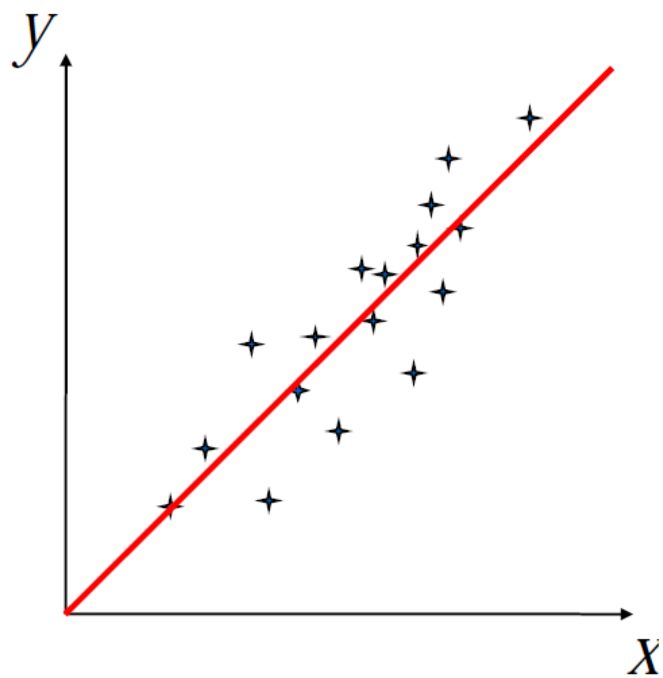
Linear Regression: One-Dimensional Case



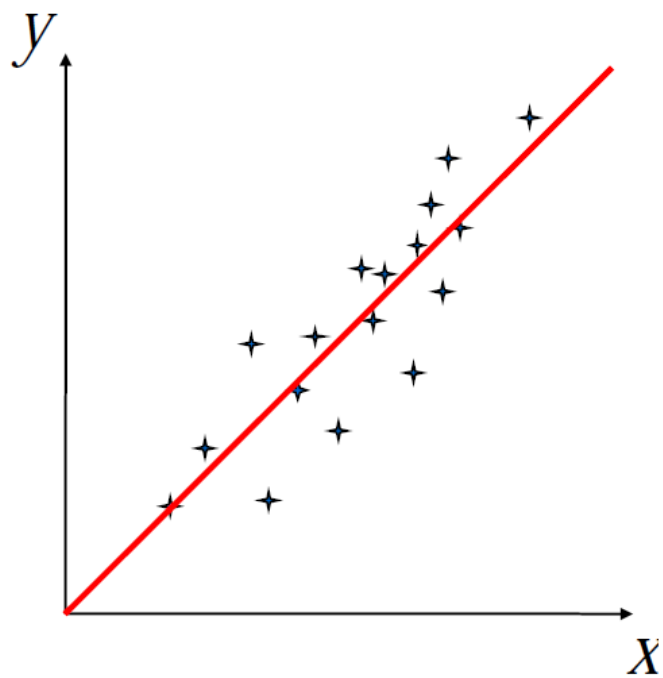
- **Given:** a set of N input-response pairs
- The inputs (x) and the responses (y) are one dimensional scalars
- **Goal:** Model the relationship between x and y

Linear Regression: One-Dimensional Case



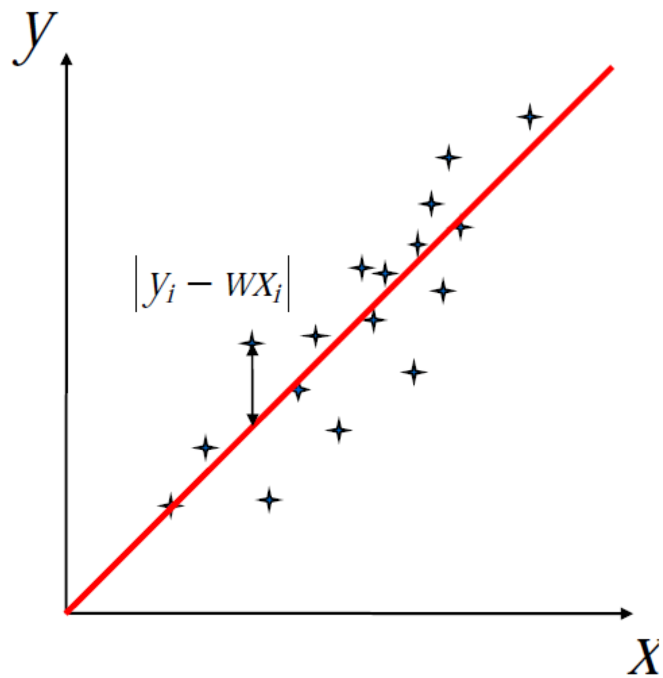
- Let's assume the relationship between x and y is linear

Linear Regression: One-Dimensional Case



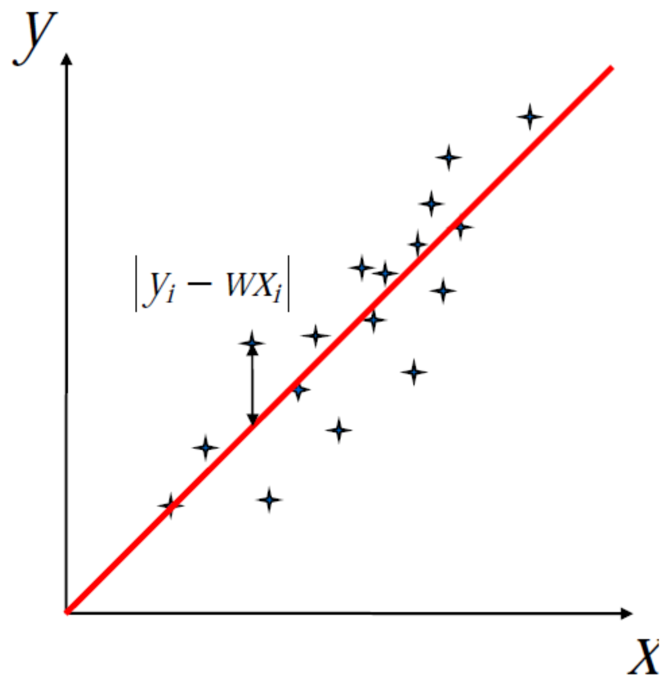
- Let's assume the relationship between x and y is linear
- Linear relationship can be defined by a straight line with *parameter* w
- Equation of the straight line: $y = wx$

Linear Regression: One-Dimensional Case



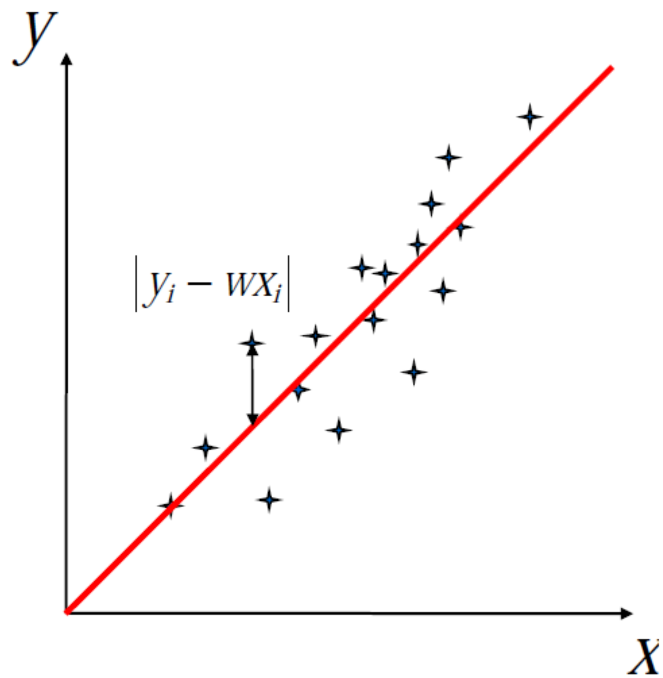
- The line may not fit the data *exactly*

Linear Regression: One-Dimensional Case



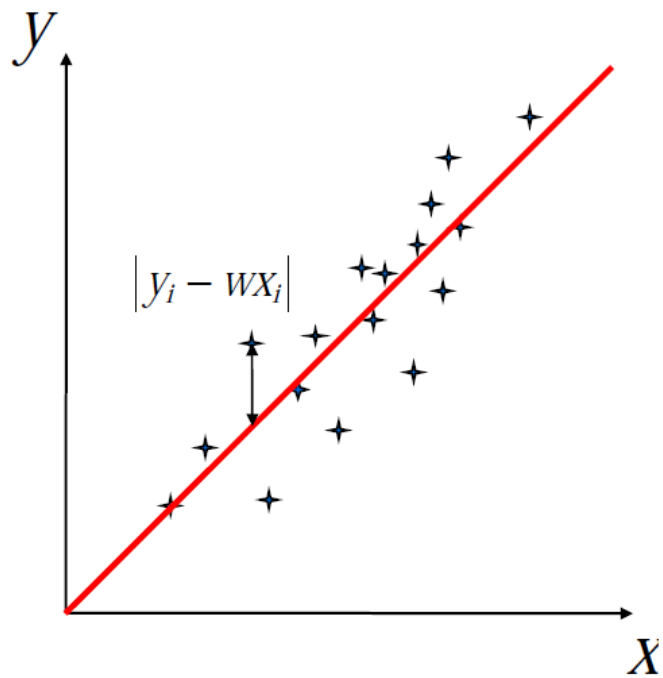
- The line may not fit the data *exactly*
- But we can try making the line a reasonable approximation

Linear Regression: One-Dimensional Case



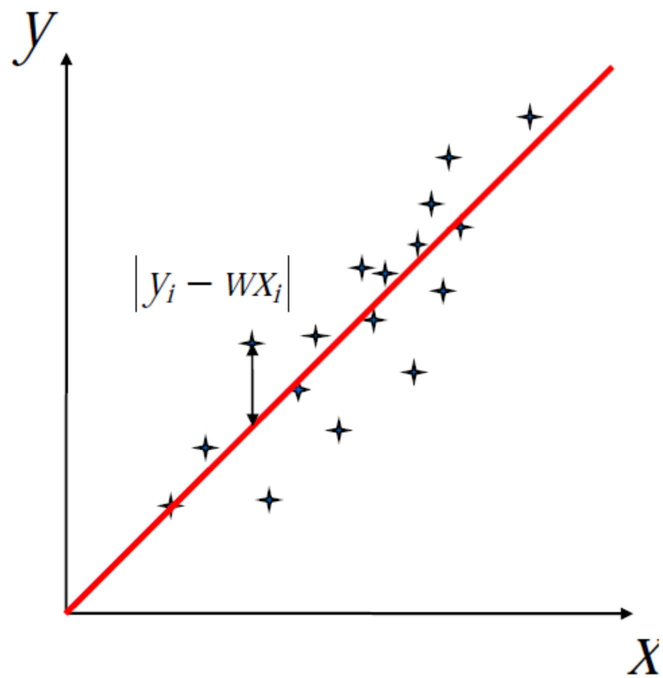
- The line may not fit the data *exactly*
- But we can try making the line a reasonable approximation
- Error for the pair (x_i, y_i) pair: $e_i = y_i - wx_i$

Linear Regression: One-Dimensional Case



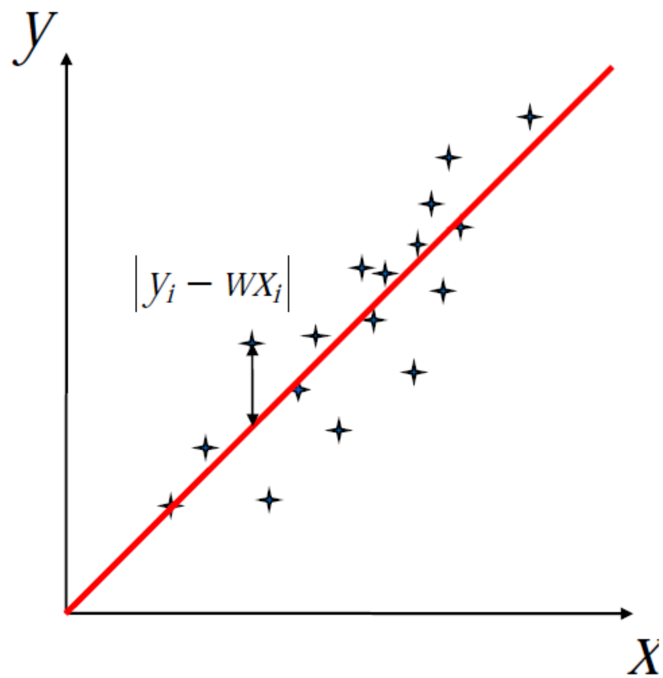
- The line may not fit the data *exactly*
- But we can try making the line a reasonable approximation
- Error for the pair (x_i, y_i) pair: $e_i = y_i - wx_i$
- The total squared error: $E = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - wx_i)^2$

Linear Regression: One-Dimensional Case



- The line may not fit the data *exactly*
- But we can try making the line a reasonable approximation
- Error for the pair (x_i, y_i) pair: $e_i = y_i - wx_i$
- The total squared error: $E = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - wx_i)^2$
- The best fitting line is defined by w minimizing the total error E

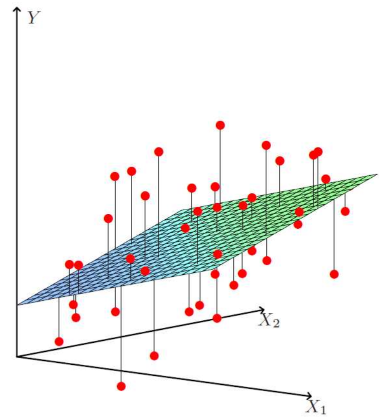
Linear Regression: One-Dimensional Case



- The line may not fit the data *exactly*
- But we can try making the line a reasonable approximation
- Error for the pair (x_i, y_i) pair: $e_i = y_i - wx_i$
- The total squared error: $E = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N}(y_i - wx_i)^2$
- The best fitting line is defined by w minimizing the total error E
- Just requires a little bit of calculus to find it (take derivative, equate to zero..)

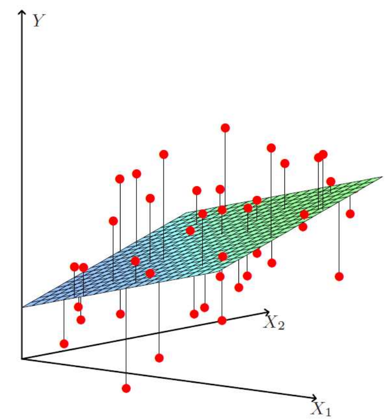
Linear Regression: In Higher Dimensions

- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



Linear Regression: In Higher Dimensions

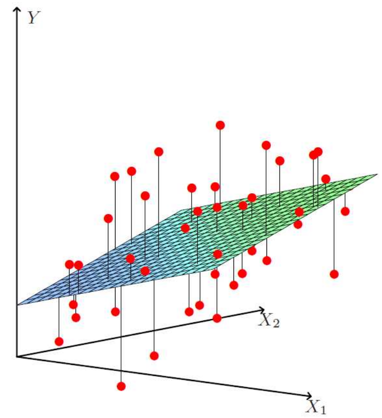
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?

Linear Regression: In Higher Dimensions

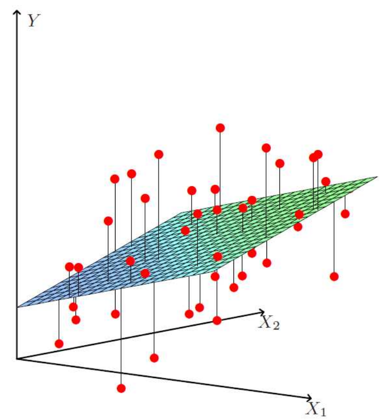
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y

Linear Regression: In Higher Dimensions

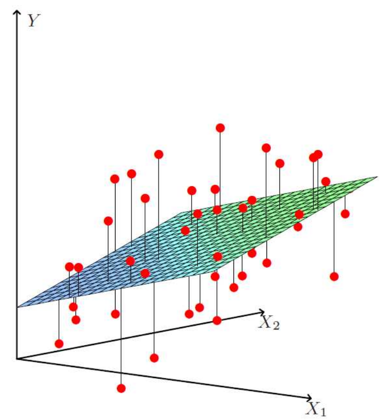
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y
  - Linear regression uses the sum-of-squared error notion of closeness

Linear Regression: In Higher Dimensions

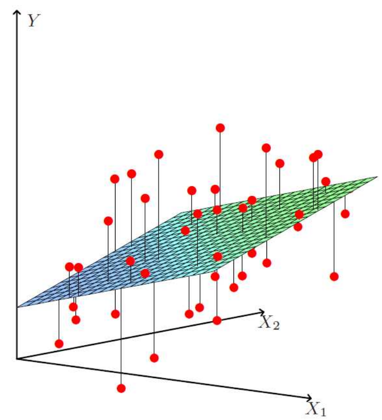
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y
    - Linear regression uses the sum-of-squared error notion of closeness
- Similar intuition carries over to higher dimensions too

Linear Regression: In Higher Dimensions

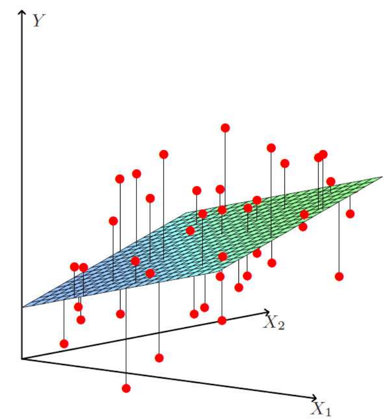
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y
  - Linear regression uses the sum-of-squared error notion of closeness
- Similar intuition carries over to higher dimensions too
  - Fitting a D-dimensional hyperplane to the data

Linear Regression: In Higher Dimensions

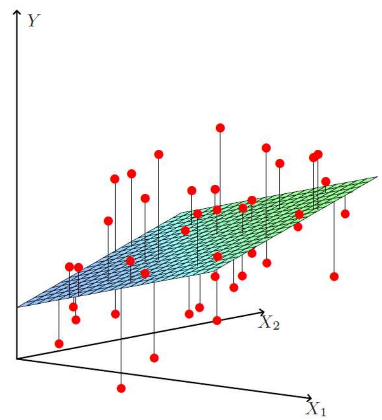
- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y
  - Linear regression uses the sum-of-squared error notion of closeness
- Similar intuition carries over to higher dimensions too
  - Fitting a D-dimensional hyperplane to the data
  - Hard to visualize in pictures though..

Linear Regression: In Higher Dimensions

- **Analogy to line fitting:** In higher dimensions, we will fit hyperplanes
- For 2-dim. inputs, linear regression fits a 2-dim. plane to the data



- Many planes are possible. Which one is the best?
- **Intuition:** Choose the one which is (on average) closest to the responses Y
  - Linear regression uses the sum-of-squared error notion of closeness
- Similar intuition carries over to higher dimensions too
  - Fitting a D-dimensional hyperplane to the data
  - Hard to visualize in pictures though..
- The hyperplane is defined by parameters $\mathbf{w}$ (a $D \times 1$ weight vector)

Linear Regression: In Higher Dimensions (Formally)

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Inputs $\mathbf{x}_i$: D-dimensional vectors ($\mathbb{R}^D$), responses y_i: scalars ($\mathbb{R}$)

Linear Regression: In Higher Dimensions (Formally)

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Inputs $\mathbf{x}_i$: D-dimensional vectors ($\mathbb{R}^D$), responses y_i: scalars ($\mathbb{R}$)
- The linear model: response is a linear function of the model parameters

$$y = f(\mathbf{x}, \mathbf{w}) = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x})$$

Linear Regression: In Higher Dimensions (Formally)

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Inputs $\mathbf{x}_i$: D-dimensional vectors ($\mathbb{R}^D$), responses y_i: scalars ($\mathbb{R}$)
- The linear model: response is a linear function of the model parameters

$$y = f(\mathbf{x}, \mathbf{w}) = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x})$$

- w_j's and b are the model parameters (b is an offset)
  - Parameters define the mapping from the inputs to responses

Linear Regression: In Higher Dimensions (Formally)

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Inputs $\mathbf{x}_i$: D-dimensional vectors ($\mathbb{R}^D$), responses y_i: scalars ($\mathbb{R}$)
- The linear model: response is a linear function of the model parameters

$$y = f(\mathbf{x}, \mathbf{w}) = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x})$$

- w_j's and b are the model parameters (b is an offset)
  - Parameters define the mapping from the inputs to responses
- Each ϕ_j is called a basis function
  - Allows change of representation of the input $\mathbf{x}$ (often desired)

Linear Regression: In Higher Dimensions

The linear model:

$$y = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) = b + \mathbf{w}^T \phi(\mathbf{x})$$

- $\phi = [\phi_1, \ldots . \phi_M]$
- $\mathbf{w} = [w_1, \ldots, w_M]$, the **weight vector** (to learn using the training data)

Linear Regression: In Higher Dimensions

The linear model:

$$y = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) = b + \mathbf{w}^T \phi(\mathbf{x})$$

- $\phi = [\phi_1, \ldots . \phi_M]$
- $\mathbf{w} = [w_1, \ldots, w_M]$, the **weight vector** (to learn using the training data)
- We consider the simplest case: $\phi(\mathbf{x}) = \mathbf{x}$
  - $\phi_j(\mathbf{x})$ is the j-th feature of the data (total D features, so $M = D$)

Linear Regression: In Higher Dimensions

The linear model:

$$y = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) = b + \mathbf{w}^T \phi(\mathbf{x})$$

- $\phi = [\phi_1, \ldots. \phi_M]$
- $\mathbf{w} = [w_1, \ldots, w_M]$, the **weight vector** (to learn using the training data)
- We consider the simplest case: $\phi(\mathbf{x}) = \mathbf{x}$
  - $\phi_j(\mathbf{x})$ is the j-th feature of the data (total D features, so $M = D$)
- The linear model becomes

$$\boxed{y = b + \sum_{j=1}^{D} w_j x_j = b + \mathbf{w}^T \mathbf{x}}$$

Linear Regression: In Higher Dimensions

The linear model:

$$y = b + \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) = b + \mathbf{w}^T \phi(\mathbf{x})$$

- $\phi = [\phi_1, \ldots . \phi_M]$
- $\mathbf{w} = [w_1, \ldots, w_M]$, the **weight vector** (to learn using the training data)
- We consider the simplest case: $\phi(\mathbf{x}) = \mathbf{x}$
  - $\phi_j(\mathbf{x})$ is the j-th feature of the data (total D features, so $M = D$)
- The linear model becomes

$$\boxed{y = b + \sum_{j=1}^{D} w_j x_j = b + \mathbf{w}^T \mathbf{x}}$$

- **Note:** Nonlinear relationships between $\mathbf{x}$ and y can be modeled using suitably chosen ϕ_j's (more when we cover Kernel Methods)

Linear Regression: In Higher Dimensions

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Fit each training example $(\mathbf{x}_i, y_i)$ using the linear model

$$y_i = b + \mathbf{w}^T \mathbf{x}_i$$

Linear Regression: In Higher Dimensions

- Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
- Fit each training example $(\mathbf{x}_i, y_i)$ using the linear model

$$y_i = b + \mathbf{w}^T \mathbf{x}_i$$

- A bit of notation abuse: write $\mathbf{w} = [b, \mathbf{w}]$, write $\mathbf{x}_i = [1, \mathbf{x}_i]$

$$y_i = \mathbf{w}^T \mathbf{x}_i$$