

Good morning!  
Доброе утро!  
早上好!

**Lecture4.**

**Linear Regression with  
Multiple Variables**

- **Linear regression** is a linear approach to model the relationship between a **dependent variable** (target variable) and one (simple regression) or more (multiple regression) **independent variables**.

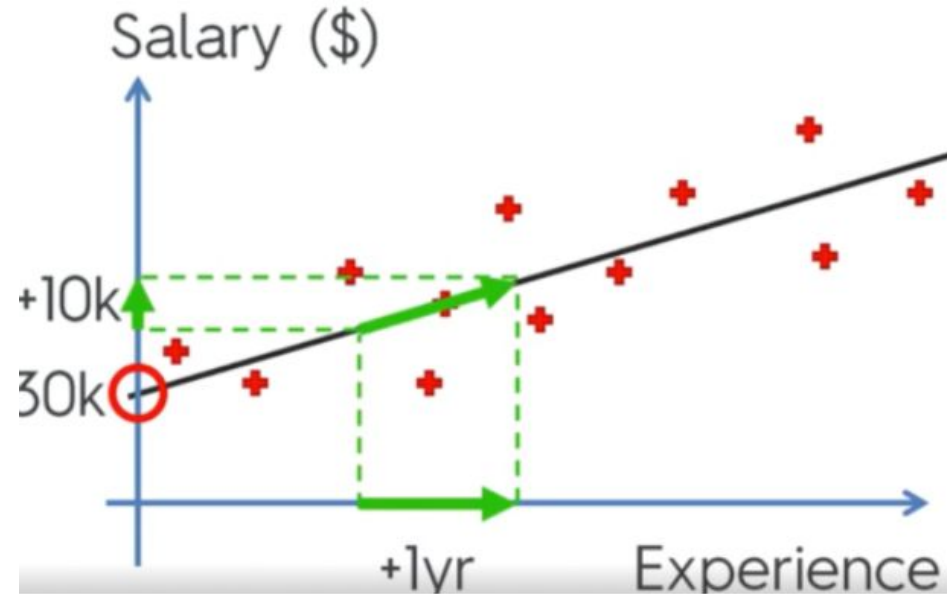
Simple linear regression uses a traditional slope-intercept form, where  $a$  and  $b$  are the coefficients that we try to “learn” and produce the most accurate predictions.  $X$  represents our input data and  $Y$  is our prediction.

$$Y = bX + a$$

**Y' = A + B \* X** SIMPLE REGRESSION EQUATION

→ **X**: predictor (present in data)  
→ **B**: coefficient (estimated by regression)  
→ **A**: intercept (estimated by regression)  
→ **Y'**: predicted value (calculated from A, B and X)

## Simple Linear Regression:



$$y = b_0 + b_1 * x$$



$$\text{Salary} = \textcircled{b_0} + \textcircled{b_1} * \text{Experience}$$

the model shows the dependence of salary on seniority. if we train the model, she will predict salary

Simple linear regression is a linear approach to modeling the relationship between a **dependent variable** and an **independent variable**, obtaining a line that best fits the data.

where **x** is the **independent variable** (height), **y** is the **dependent variable** (weight), **b** is the **slope**, and **a** is the **intercept**. The **intercept** represents the value of **y** when **x** is 0 and the **slope** indicates the steepness of the line. The objective is to obtain the line that best fits our data (**the line that minimize the sum of square errors**). The error is the difference between the real value **y** and the predicted value **y\_hat**, which is the value obtained using the calculated linear equation.

$$\text{error} = y(\text{real}) - y(\text{predicted}) = y(\text{real}) - (a + bx)$$

# The Simple Linear Regression Problem (yet vague)

Given

- ▶ a set  $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \subseteq \mathbb{R} \times \mathbb{R}$  called **training data**,

compute the line that describes the data generating process best.

# The Simple Linear Model

For given predictor/input  $x \in \mathbb{R}$ , the **simple linear model** predicts/outputs

$$\hat{y}(x) := \hat{\beta}_0 + \hat{\beta}_1 x$$

with **parameters**  $(\hat{\beta}_0, \hat{\beta}_1)$  called

$\hat{\beta}_0$  **intercept / bias / offset**

$\hat{\beta}_1$  **slope**

- 1: **procedure** PREDICT-SIMPLE-LINREG( $x \in \mathbb{R}, \hat{\beta}_0, \hat{\beta}_1 \in \mathbb{R}$ )
- 2:      $\hat{y} := \hat{\beta}_0 + \hat{\beta}_1 x$
- 3:     **return**  $\hat{y}$



# When is a Model Good?

We still need to specify what “describes the data generating process best” means. — What are good predictions  $\hat{y}(x)$ ?

Predictions are considered better the smaller the difference between

- ▶ an **observed**  $y_n$  (for predictors  $x_n$ ) and
- ▶ a **predicted**  $\hat{y}_n := \hat{y}(x_n)$

are, e.g., the smaller the **L2 loss** / **squared error**:

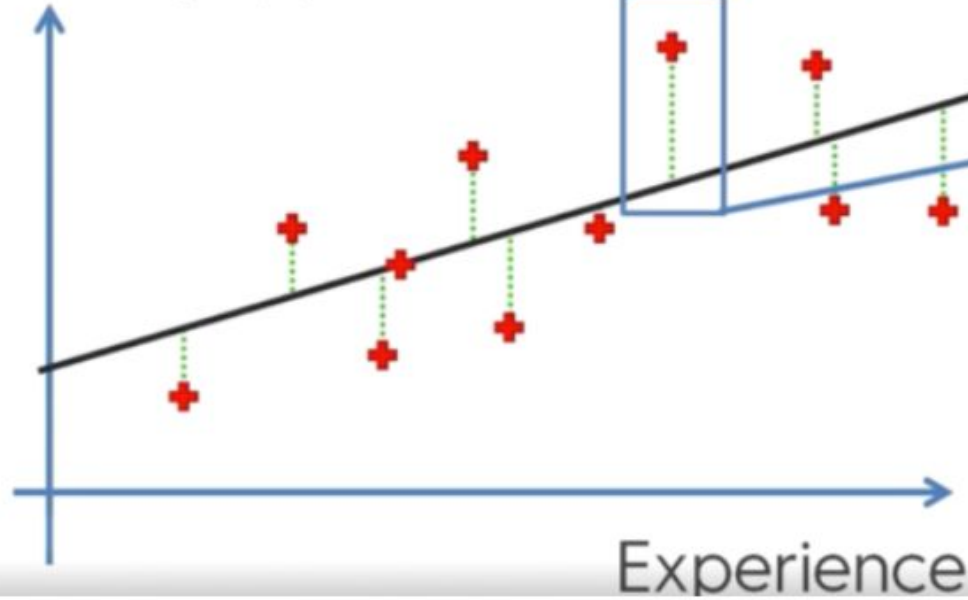
$$\ell(y_n, \hat{y}_n) := (y_n - \hat{y}_n)^2$$

Note: Other error measures such as absolute error  $\ell(y_n, \hat{y}_n) = |y_n - \hat{y}_n|$  are also possible, but more difficult to handle.



## Simple Linear Regression:

Salary (\$)



$$\text{SUM } (y - \hat{y})^2 \rightarrow \min$$

## When is a Model Good?

Pointwise losses are usually averaged over a dataset  $\mathcal{D}$

$$\text{err}(\hat{y}; \mathcal{D}) := \frac{1}{N} \text{RSS}(\hat{y}; \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}(x_n))^2$$

$$\text{or } \text{err}(\hat{y}; \mathcal{D}) := \text{RSS}(\hat{y}; \mathcal{D}) := \sum_{n=1}^N (y_n - \hat{y}(x_n))^2$$

called **residual sum of squares** (RSS) or generally **error/risk**.

Equivalently, often **Root Mean Square Error** (RMSE) is used:

$$\text{err}(\hat{y}; \mathcal{D}) := \text{RMSE}(\hat{y}; \mathcal{D}) := \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}(x_n))^2}$$

# The Simple Linear Regression Problem

Given

- ▶ a set  $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \subseteq \mathbb{R} \times \mathbb{R}$  called **training data**,

compute the **parameters**  $(\hat{\beta}_0, \hat{\beta}_1)$  of a linear **regression function**

$$\hat{y}(x) := \hat{\beta}_0 + \hat{\beta}_1 x$$

s.t. for a set  $\mathcal{D}^{\text{test}} \subseteq \mathbb{R} \times \mathbb{R}$  called **test set** the **test error**

$$\text{err}(\hat{y}; \mathcal{D}^{\text{test}}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} (y - \hat{y}(x))^2$$

is minimal.

# Learning the Least Squares Estimates

The least squares estimates can be written in closed form:

$$\hat{\beta}_1 = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

1: **procedure**

LEARN-SIMPLE-LINREG( $\mathcal{D}^{\text{train}} := \{(x_1, y_1), \dots, (x_N, y_N)\} \in \mathbb{R} \times \mathbb{R}$ )

2:  $\bar{x} := \frac{1}{N} \sum_{n=1}^N x_n$

3:  $\bar{y} := \frac{1}{N} \sum_{n=1}^N y_n$

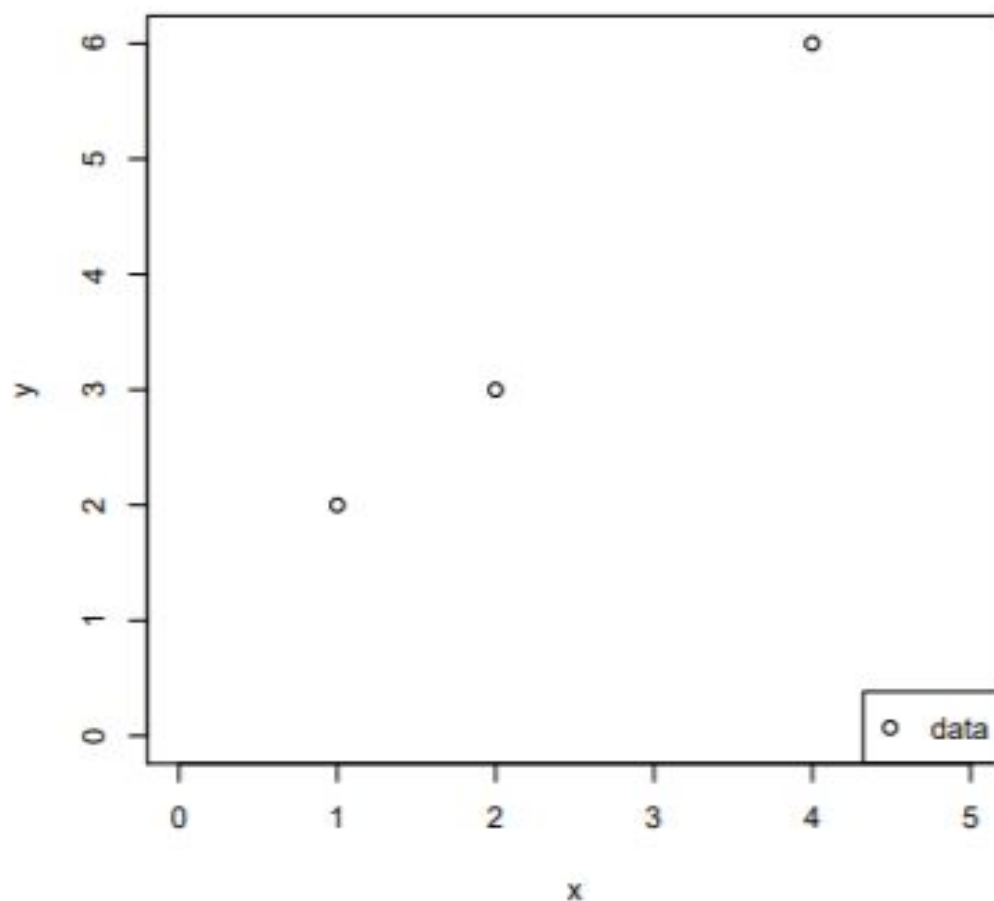
4:  $\hat{\beta}_1 := \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2}$

5:  $\hat{\beta}_0 := \bar{y} - \hat{\beta}_1 \bar{x}$

6: **return**  $(\hat{\beta}_0, \hat{\beta}_1)$

## A Toy Example

Given the data  $\mathcal{D} := \{(1, 2), (2, 3), (4, 6)\}$ , predict a value for  $x = 3$ .





## A Toy Example / Least Squares Estimates

Given the data  $\mathcal{D} := \{(1, 2), (2, 3), (4, 6)\}$ , predict a value for  $x = 3$ .

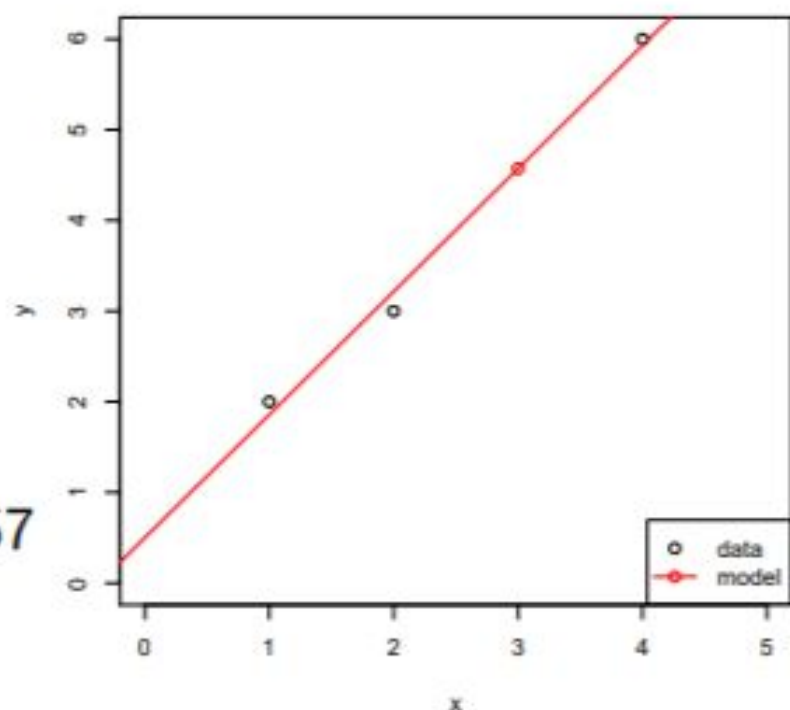
Use a simple linear model.

$$\bar{x} = 7/3, \bar{y} = 11/3.$$

| $n$      | $x_n - \bar{x}$ | $y_n - \bar{y}$ | $(x_n - \bar{x})^2$ | $(x_n - \bar{x}) \cdot (y_n - \bar{y})$ |
|----------|-----------------|-----------------|---------------------|---|
| 1        | $-4/3$          | $-5/3$          | $16/9$              | $20/9$                                  |
| 2        | $-1/3$          | $-2/3$          | $1/9$               | $2/9$                                   |
| 3        | $5/3$           | $7/3$           | $25/9$              | $35/9$                                  |
| $\Sigma$ |                 |                 | $42/9$              | $57/9$                                  |

$$\hat{\beta}_1 = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2} = 57/42 = 1.357$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = \frac{11}{3} - \frac{57}{42} \cdot \frac{7}{3} = \frac{63}{126} = 0.5$$





# A Toy Example / Least Squares Estimates

Given the data  $\mathcal{D} := \{(1, 2), (2, 3), (4, 6)\}$ , predict a value for  $x = 3$ .

Use a simple linear model.

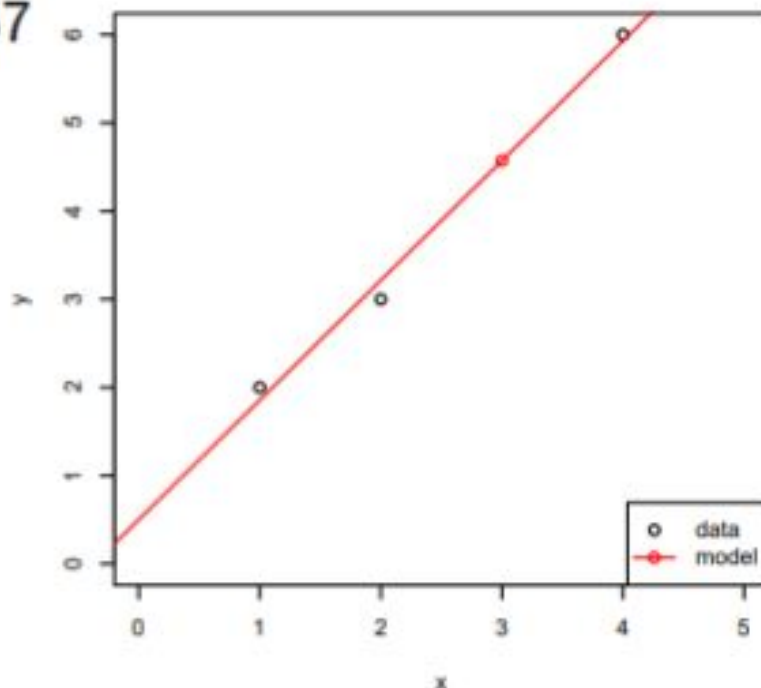
$$\hat{\beta}_1 = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2} = 57/42 = 1.357$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = \frac{11}{3} - \frac{57}{42} \cdot \frac{7}{3} = \frac{63}{126} = 0.5$$

RSS:

| $n$      | $y_n$ | $\hat{y}_n$ | $(y_n - \hat{y}_n)^2$ |
|----------|-------|-------------|-----------------------|
| 1        | 2     | 1.857       | 0.020                 |
| 2        | 3     | 3.214       | 0.046                 |
| 3        | 6     | 5.929       | 0.005                 |
| $\Sigma$ |       |             | 0.071                 |

$$\hat{y}(3) = 4.571$$



- Example ☐




<http://localhost:8890/notebooks/Regression%20for%20height-weight.ipynb>

This is a link to the lecture. You now need to view it, preferably using headphones.  
There are subtitles in Chinese here.

- <https://www.coursera.org/lecture/machine-learning/model-representation-db3jS>
- <https://www.coursera.org/learn/machine-learning/home/week/1>

Previously, we have calculated two linear models, one for men and another for women, to predict the weight based on the height of a person, obtaining the following results:

- Males  $\rightarrow \text{Weight} = -224.50 + 5.96 * \text{Height}$
- Females  $\rightarrow \text{Weight} = -246.01 + 5.99 * \text{Height}$

So far, we have employed one independent variable to predict the weight of the person  $\text{Weight} = f(\text{Height})$ , creating two different models. Maybe you are thinking   Can we create a model that predicts the weight using both height and gender as independent variables? The answer is YES! 

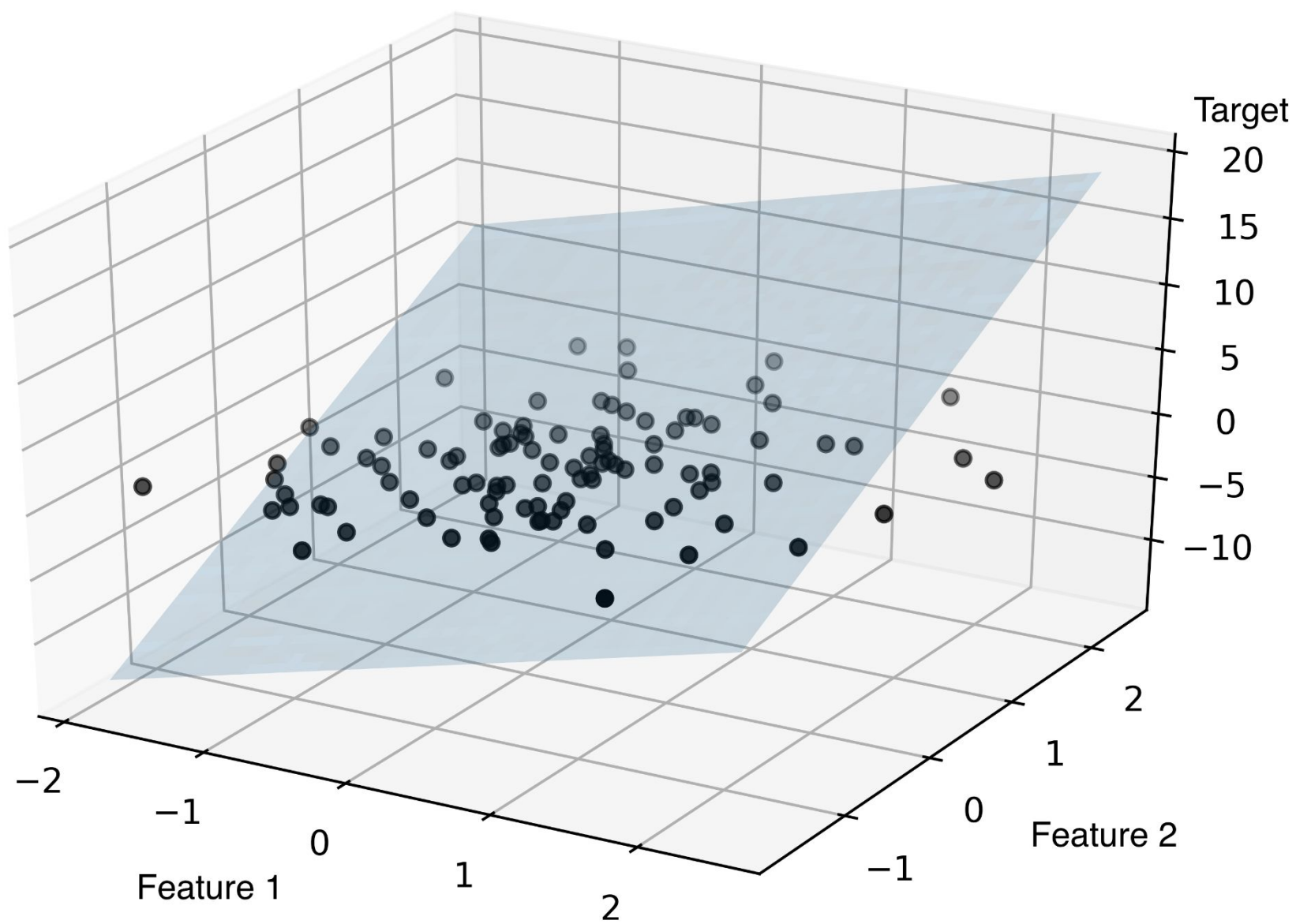
★ And here is where multiple linear regression comes into play!

**Multiple linear regression** uses a linear function to predict the value of a target variable  $y$ , containing the function  $n$  independent variable  $x = [x_1, x_2, x_3, \dots, x_n]$ .

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

We obtain the values of the parameters  $b_i$ , using the same technique as in simple linear regression (**least square error**). After fitting the model, we can use the equation to predict the value of the target variable  $y$ . In our case, we use height and gender to predict the weight of a person *Weight = f(Height, Gender)*.







# watch the video lecture

- <https://www.coursera.org/lecture/machine-learning/what-is-machine-learning-Ujm7v>
- <https://www.coursera.org/learn/machine-learning/lecture/6Nj1q/multiple-features>

- **Numerical variables** represent values that can be measured and sorted in ascending and descending order such as the **height** of a person.
- **Categorical variables** are values that can be sorted in groups or categories such as the **gender** of a person.
- **Multiple linear regression** accepts not only **numerical variables**, but also **categorical** ones. To include a **categorical variable** in a regression model, the variable has to be encoded as a **binary variable (dummy variable)**.

# Preprocessing Data

## If data set are strings

- We saw in our initial exploration that most of the columns in our data set are strings, but the algorithms in scikit-learn understand only numeric data. Luckily, the scikit-learn library provides us with many methods for converting string data into numerical data. One such method is the `LabelEncoder()` method. We will use this method to convert the categorical labels in our data set like 'won' and 'loss' into numerical labels. To visualize what we are trying to achieve with the `LabelEncoder()` method let's consider the images below.

- The image below represents a dataframe that has one column named 'color' and three records 'Red', 'Green' and 'Blue'.

|   | Color |
|---|-------|
| 0 | Red   |
| 1 | Green |
| 2 | Blue  |

|   | Color |
|---|-------|
| 0 | 1     |
| 1 | 2     |
| 2 | 3     |

- Since the machine learning algorithms in scikit-learn understand only numeric inputs, we would like to convert the categorical labels like 'Red', 'Green' and 'Blue' into numeric labels. When we are done converting the categorical labels in the original dataframe, we would get something like this

# For home work

- [https://www.youtube.com/watch?v=EuBBz3bl-aA&list=PLblh5JKOoLUICTaGLRoHQDuF\\_7q2GfuJF&index=5](https://www.youtube.com/watch?v=EuBBz3bl-aA&list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&index=5)
- [https://www.youtube.com/watch?v=Q81RR3yKn30&list=PLblh5JKOoLUICTaGLRoHQDuF\\_7q2GfuJF&index=18](https://www.youtube.com/watch?v=Q81RR3yKn30&list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&index=18)