Linear Regression

GENERAL ASSEMBLY

Kinds of ML

- **Supervised learning** - Given X, can we predict Y?
- **Unsupervised learning** - What patterns can we find in X? There is no Y.
- **Reinforcement learning** - How can a virtual agent maximize a reward.

Supervised learning

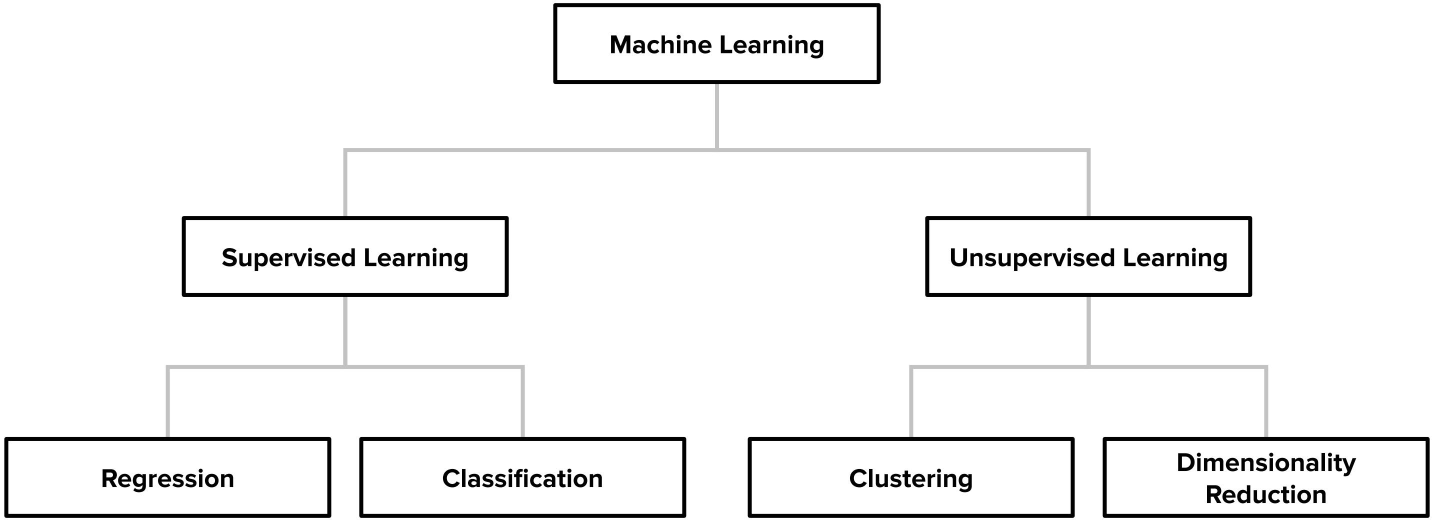
- Classification
- Regression

Two Kinds of Supervised Learning

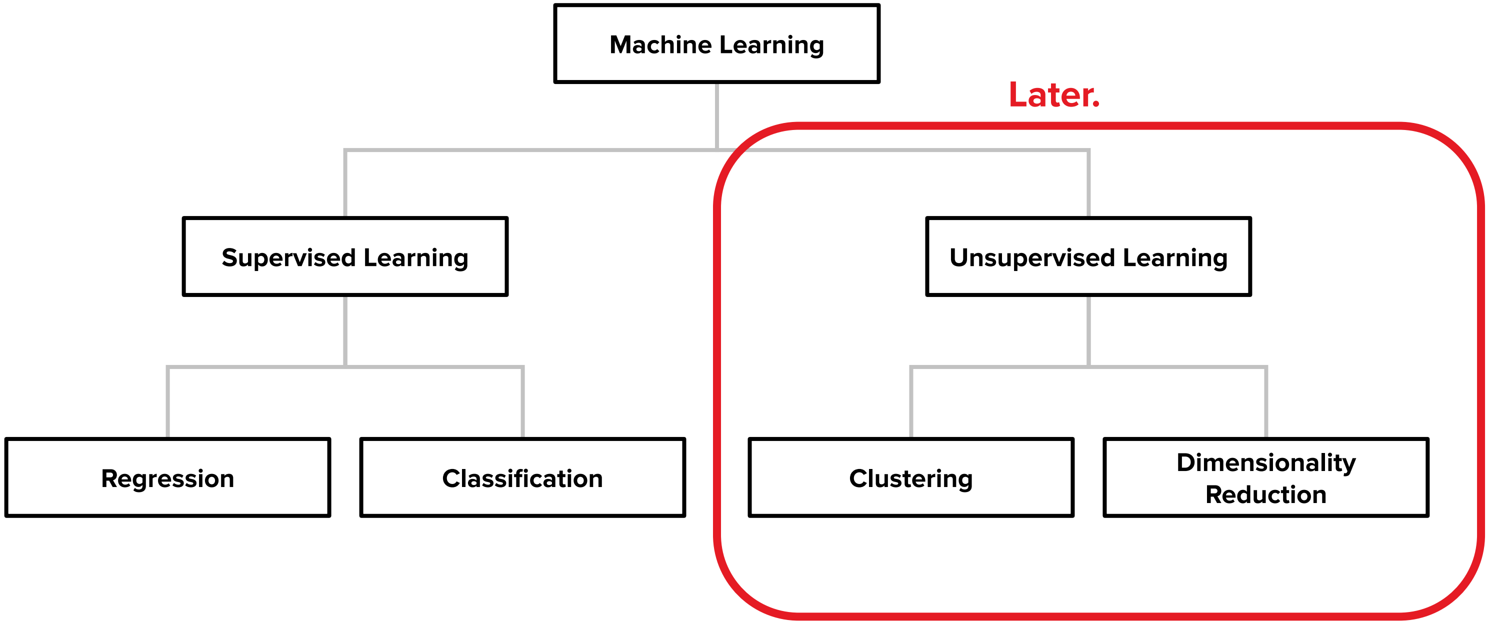
Regression: *y*-variable is a continuous number

- *Crop yields*
- *House sale price*
- *Other examples?*

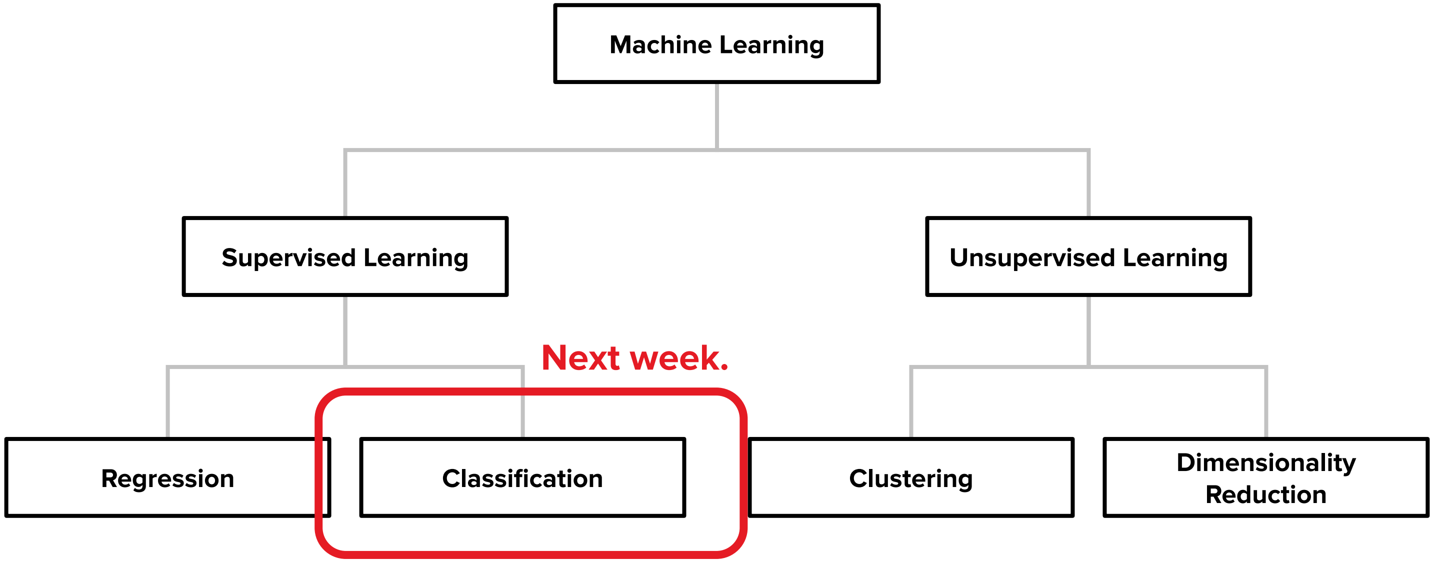
Roadmap of ML



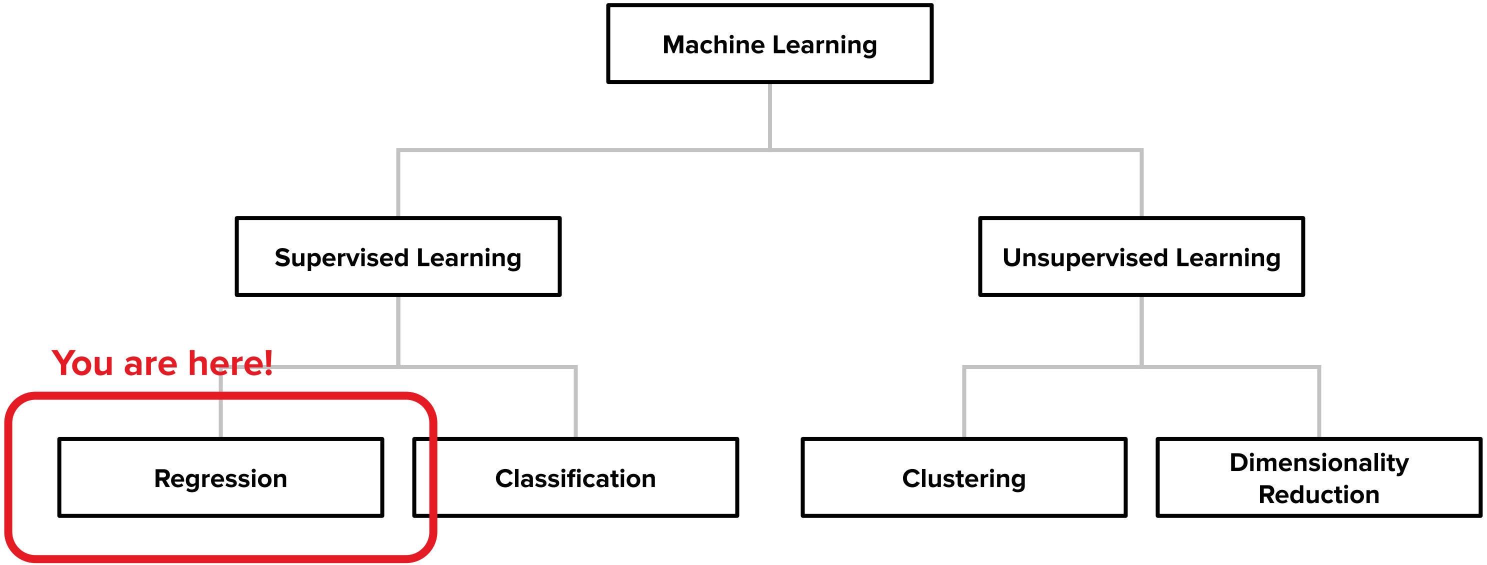
Roadmap of ML



Roadmap of ML

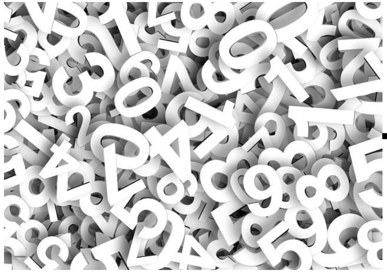


Roadmap of ML



Supervised Learning Transparency

Data



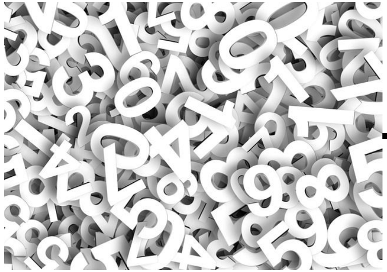
img source

???

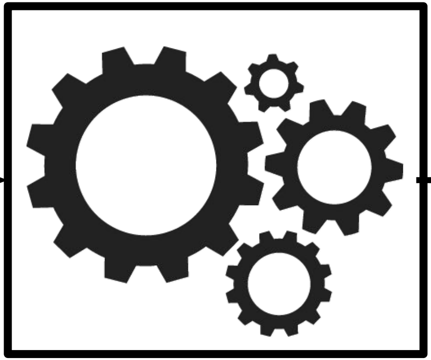
Profitable Results!

Supervised Learning Transparency

Data



img source



Profitable Results!

Let’s do it!

Linear Regression

Supervised, white-box, regression

Ordinary least squares linear regression (**OLS**)

Predict response variable (*y*) from at least one independent variable (*x*).

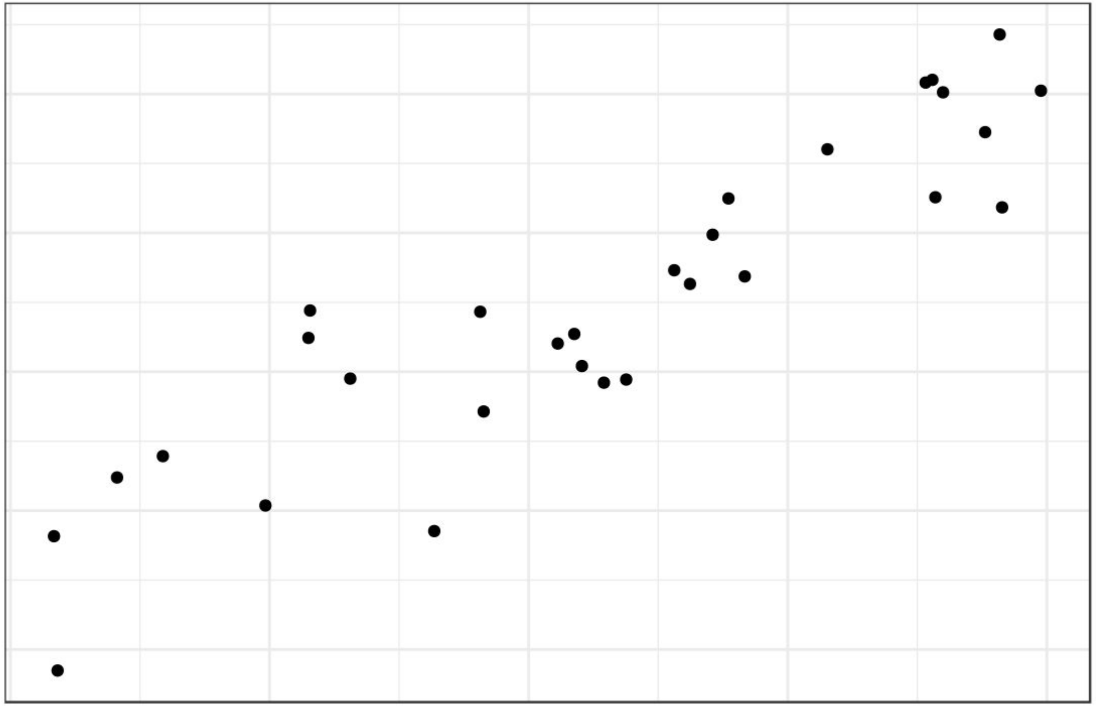
$$y = \beta_0 + \beta_1 x + \varepsilon$$

Ordinary least squares linear regression (**OLS**)

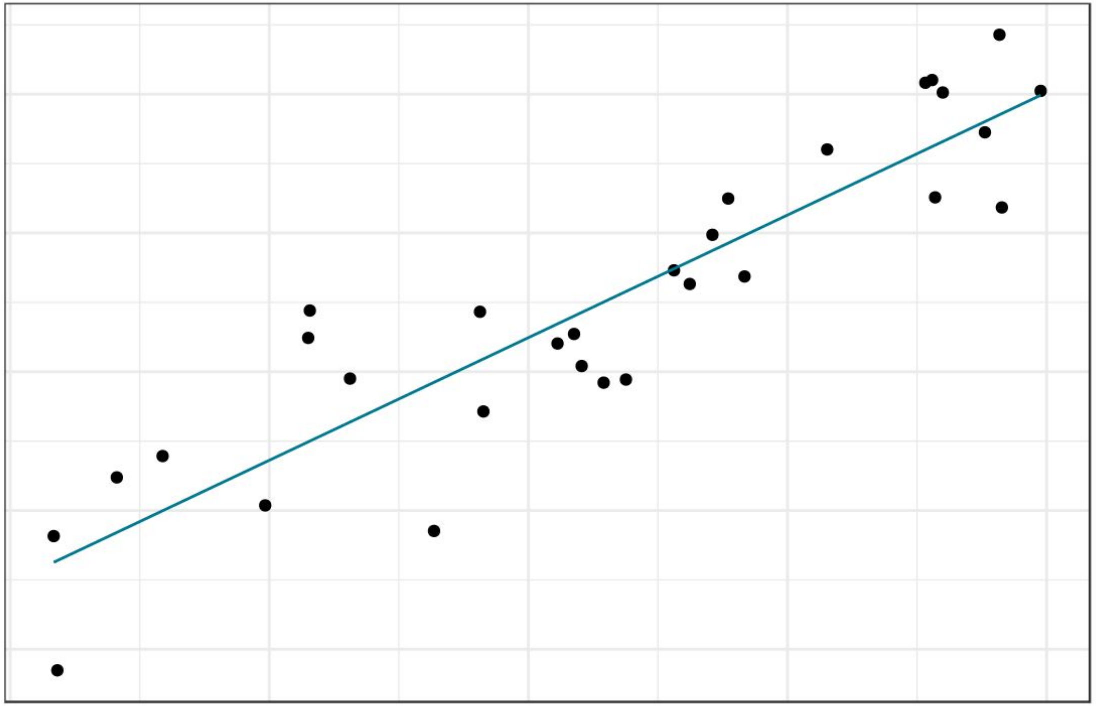
$$y = \beta_0 + \beta_1 x + \varepsilon$$

The game is “find the best betas.”

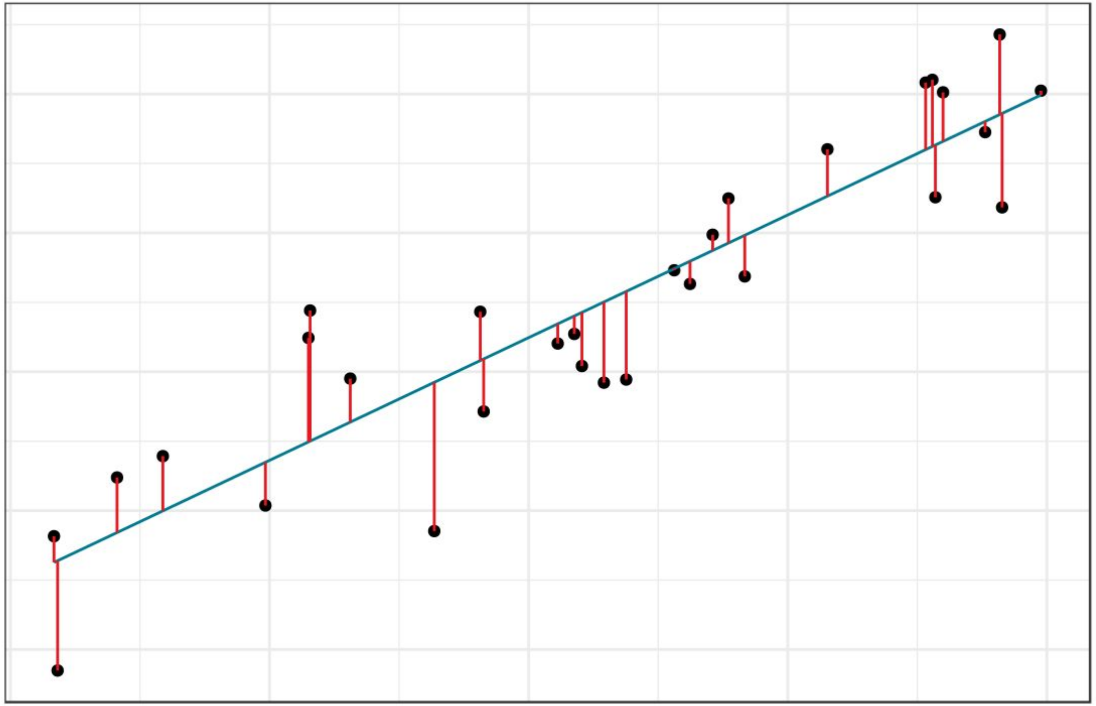
Graphically, this is:



Graphically, this is:



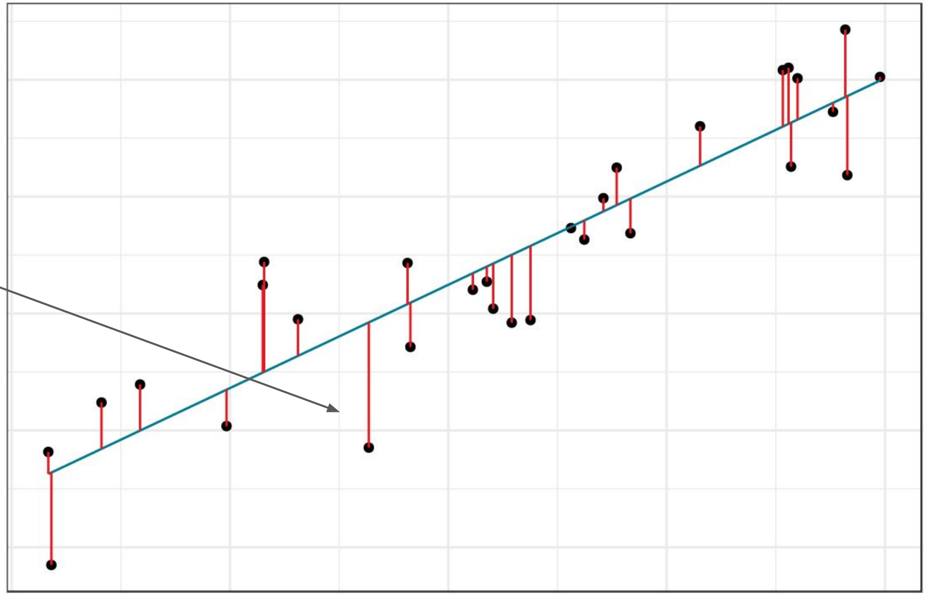
Graphically, this is:



Graphically, this is:

The difference between the actual and the predicted value is called a **residual**.

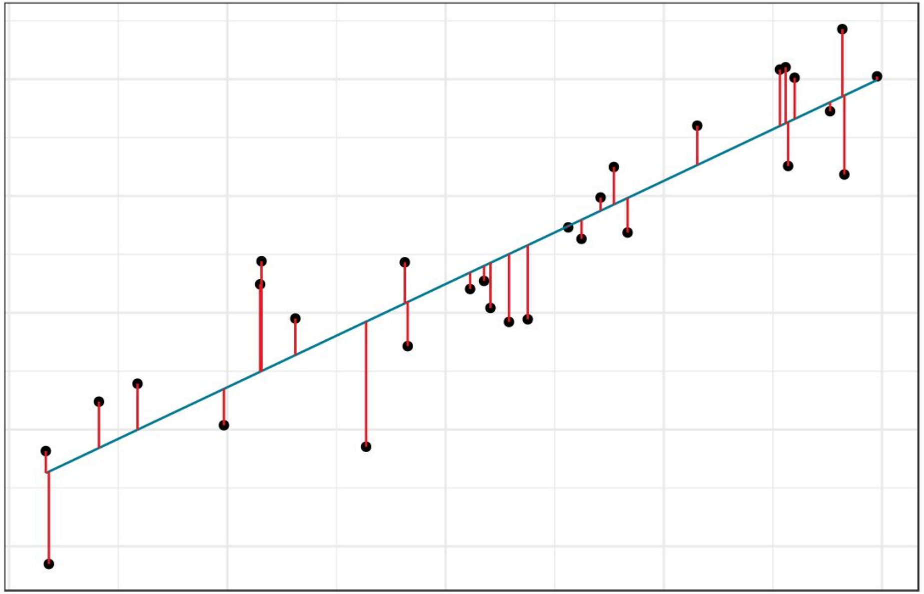
The line of “best fit” **minimizes the sum of the squared residuals**.



Graphically, this is:

Minimize the **sum of the squared residuals**

OLS: “ordinary least squares”



The Residual

A **residual** is the same as an error

$$e_i = y_i - \hat{y}_i$$

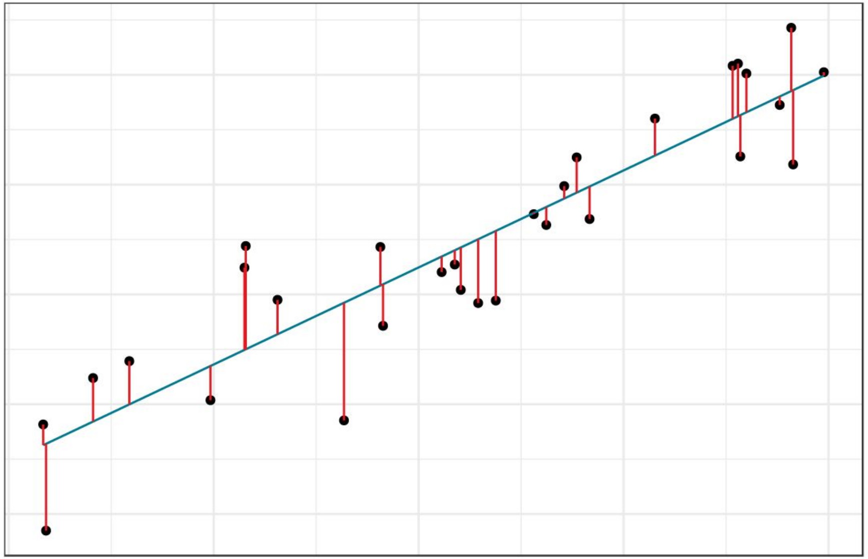
This measures how “off” a prediction was.

Sum of Squared Errors (SSE)

$$SSE = \sum (y_i - \hat{y}_i)^2 = \sum e_i^2$$

Fitting OLS models

This is the quantity we seek to **minimize** to find the best values for our betas!



Multiple Linear Regression

Supervised, white-box, regression

Multiple Linear Regression

Generally, you'll have more than one predictor variable.

Formula for a predicted y value:

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 + \ldots \hat{\beta}_\mathbf{p} \mathbf{x}_\mathbf{p} + \varepsilon$$

Notation: Design Matrix

Mathematically speaking, every time we fit a model, we need a data matrix, sometimes called a **design matrix**:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Notation: Design Matrix

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \boxed{1 & x_{21} & x_{22} & \cdots & x_{2p}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Each **row** is an **observation.**

Notation: Design Matrix

$$X = \begin{bmatrix} 1 & x_{11} & \boxed{x_{12}} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Each **column**
is a variable.

Notation: Design Matrix

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

The first column is all 1s and corresponds to the **intercept**. (scikit-learn handles this automatically)

Multiple linear regression

Interpreting results

Now when you interpret a beta weight, you must use the caveat: “holding everything else constant”.

Multiple linear regression

Interpreting results example:

If beta1 is 5.0 then a 1 degree celsius increase in temperature is associated with a 5MW increase in energy usage, **holding everything else constant**.

Assumption

For inference to make sense:

There is no **M**ulticollinearity in the predictor variables.

Check for highly correlated variables using df.corr & seaborn. (Or better, VIF in statsmodels. Picks up patterns of correlation.).

Let’s do it!

The Residual

A **residual** is:

$$e_i = y_i - \hat{y}_i$$

How far “off” a prediction was.

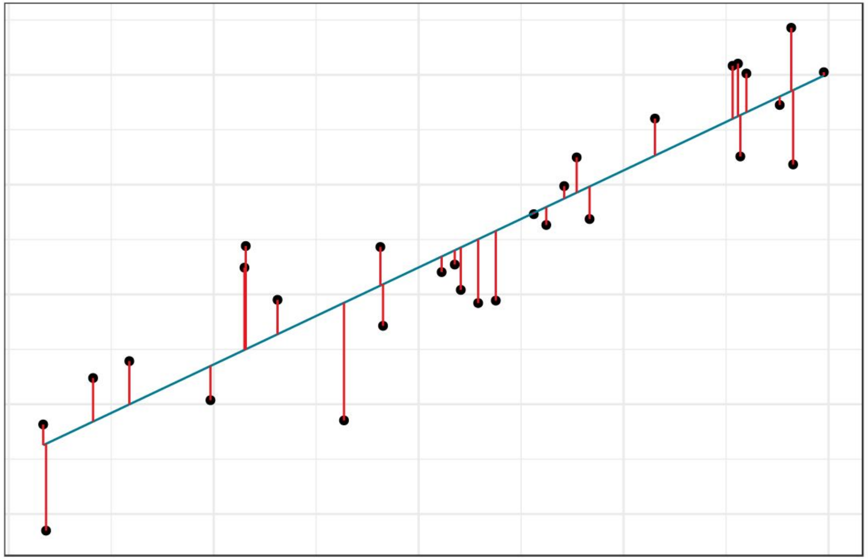
The error.

Sum of Squared Errors (SSE)

$$SSE = \sum (y_i - \hat{y}_i)^2 = \sum e_i^2$$

Fitting OLS models

This is the quantity we seek to **minimize** to find the best values for our betas!



LINE Assumptions

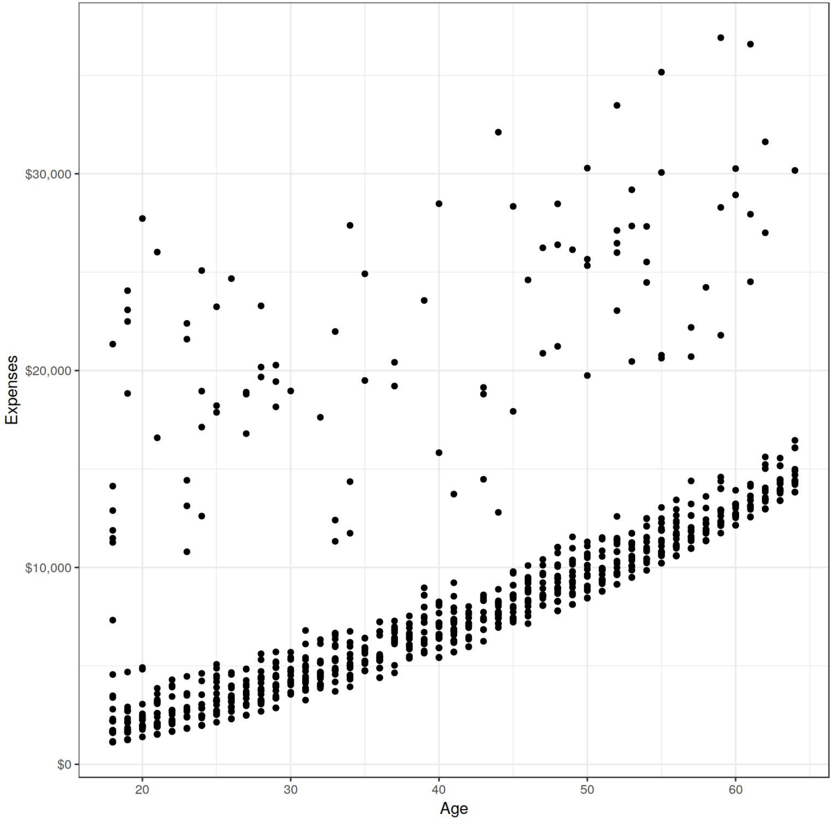
OLS Assumptions

OLS comes with some assumptions. Some matter for performance and some matter for inference.

OLS Regression Assumptions

- **L** - Linearity. Relationship between *x* and *y* should be approximately linear.
- **I** - Independence of errors. (Time series issue)
- **N** - Normality. Residuals should be approximately normally distributed.
- **E** - Equal variances, aka “**homoscedasticity**”. Residuals should have approximately equal variances.

L is for Linearity



L is for Linearity

We will learn how to transform X or y to try to make the relationship more linear.

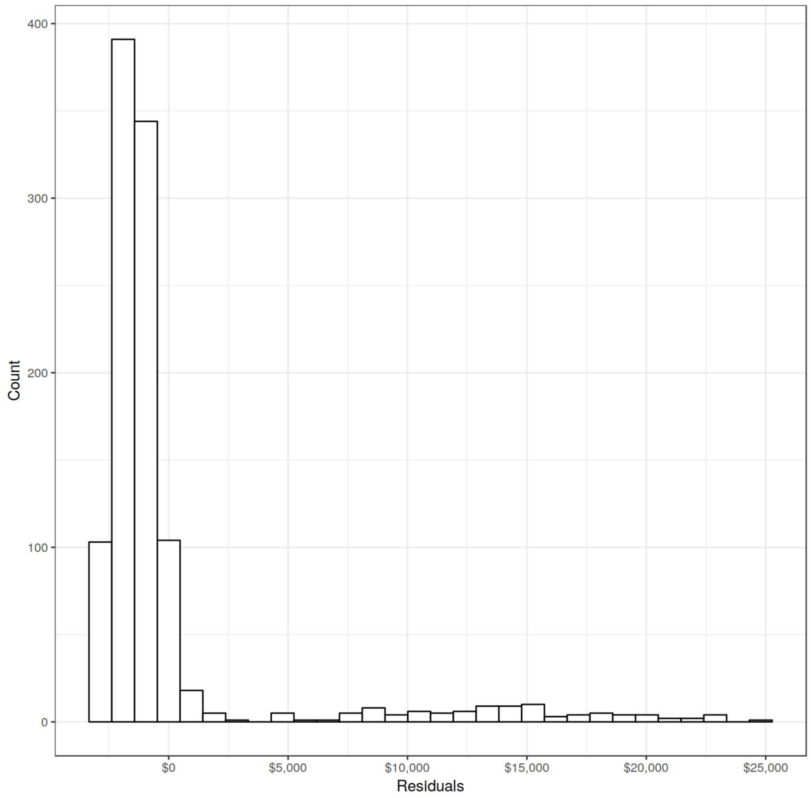
Or maybe another algorithm would be better.

I is for Independence

Are our samples independent from one another?

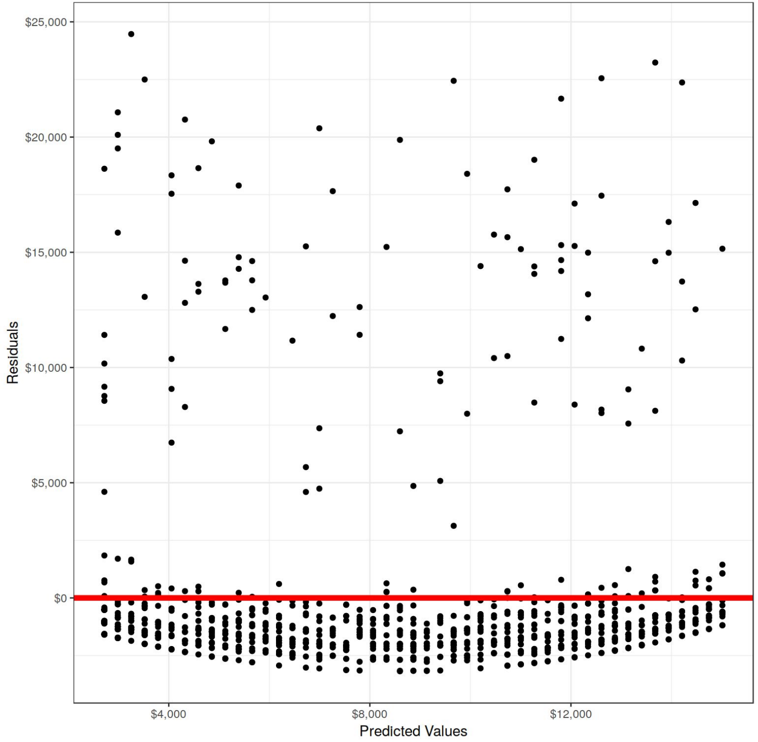
With time series, they usually are not independent. The previous row tells us something about the next row.

N is for Normality of errors



E is for Equal Variances

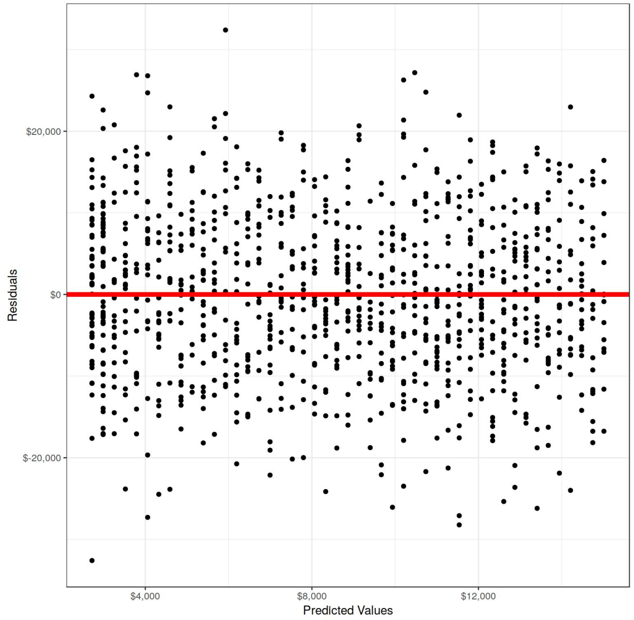
We can see a parabolic pattern in our residual plot.



E is for Equal Variance

We want to see **absolute randomness** in our residual plots. That is, no pattern whatsoever.

Here's an example of an ideal residual plot.



 |

What to do if our LINE assumptions are violated?

A common scenario:

- A slightly **curvilinear** relationship between *x* and *y*
- Very right-skew residuals
- Residuals that tend to spread out from right to left (a “fan shape”)

A log transformation often helps fix these issues

LINEM Assumptions

- **L** - Linear relationship
- **I** - Independent errors (time series issue, often)
- **N** - Normally distributed errors
- **E** - Equal variance of errors (homoscedasticity)

Add an M for multiple regression (more than 1 X)

- **M** - No multicollinearity (highly correlated predictors)

Let’s do it!