

# — Linear Regression

*... and the modeling philosophy*



# Agenda



What is ~~statistical modeling~~ machine learning?



What is linear regression?



The inner mechanics of linear regression



Assumptions of linear regression, and how to check them




What do we do if those assumptions aren't met?

# Learning Objectives

- Describe statistical modeling and "machine learning".
- Carry out simple and multiple linear regressions.
- Calculate mean squared error and other error metrics.
- Be able to interpret the coefficients of a linear regression model.
- State the assumptions of a linear regression model.
- Fit, generate predictions from, and evaluate a linear regression model in sklearn.





“Essentially, all models are wrong, but some are useful.”  
—George Box, 1987

# DATA SCIENCE PROCESS

1. Define problem.
2. Gather data.
3. Explore data.
4. Model with data.
5. Evaluate model.
6. Answer problem.





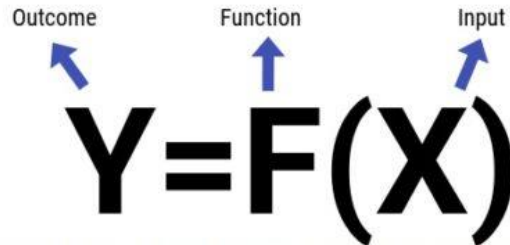
# What is Machine Learning?

Machine learning is the process of letting your **machine** use data to **learn** the relationship between some predictors ( $x$ -variables) and some response(s) ( $y$ -variables).

# What is Machine Learning?

Machine learning is the process of letting your **machine** use data to **learn** the relationship between some predictors (x-variables) and some response(s) (y-variables).

This is a fancy way of saying **statistical modeling**.



A diagram illustrating the equation  $Y = F(X)$ . The equation is written in large, bold, black font. Above the equation, three labels are positioned: "Outcome" above the  $Y$ , "Function" above the  $F$ , and "Input" above the  $X$ . Three blue arrows point from each label down to its corresponding part of the equation: one from "Outcome" to  $Y$ , one from "Function" to  $F$ , and one from "Input" to  $X$ .



# Ok then, so what is statistical modeling?

Statistical modeling is the process of combining data with **statistical theory** to **model** the real-world relationship between predictors ( $x$ -variables) and some response(s) ( $y$ -variables).



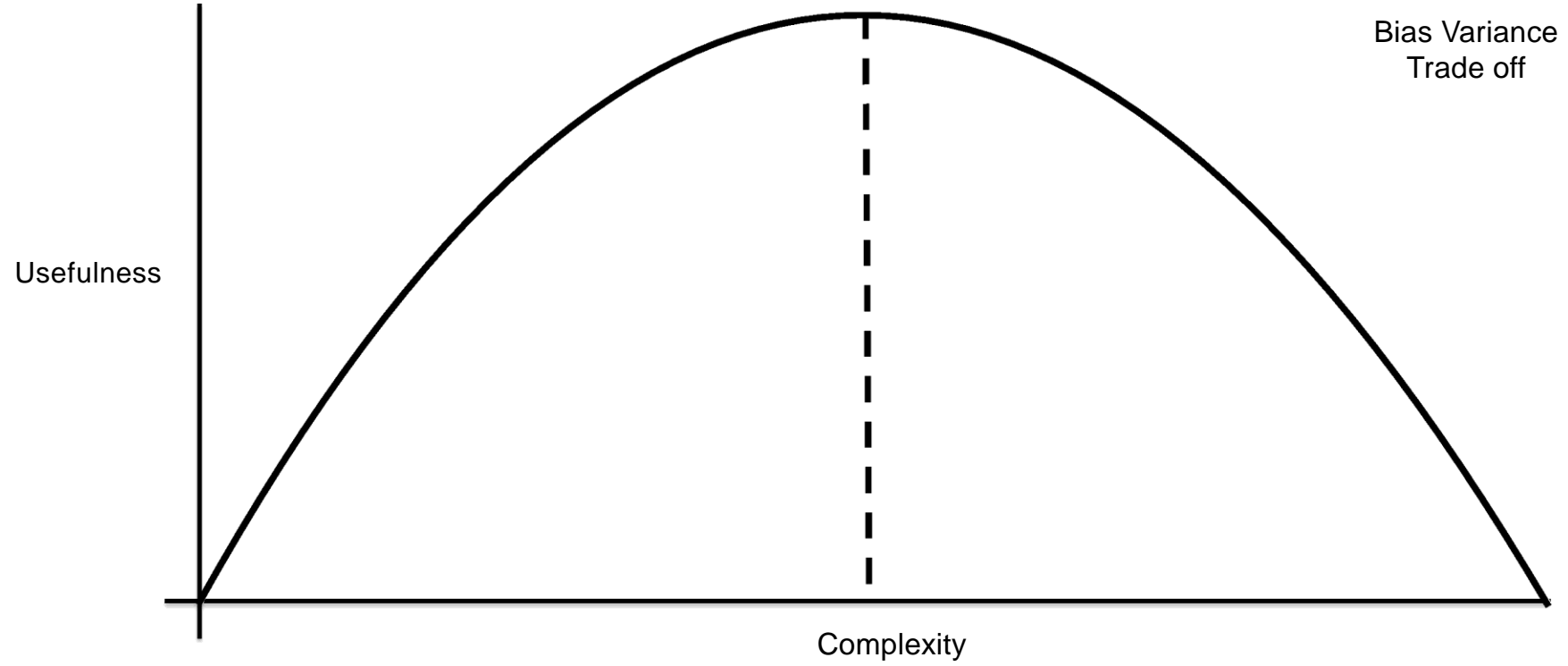


## Ok then, so what is statistical modeling?

Statistical modeling is the process of combining data with **statistical theory** to **model** the real-world relationship between predictors ( $x$ -variables) and some response(s) ( $y$ -variables).

This is a more down-to-earth way of saying **machine learning**.

# Modeling



# Two\* Kinds of ML

In essence, all machine learning models fall into one of two categories:

- **Supervised learning** - Given X, can we predict Y?
- **Unsupervised learning** - What does X look like, *really*? There is no Y.

\*There are more. Kinda. The big third category is **reinforcement learning**, which is a field still in the process of being invented.

# Two Kinds of Supervised Learning

Supervised learning models fall into two different buckets:

**Regression** - this is when our y-variable is numeric.

- *“Given the past values of the stock price of Apple, what will tomorrow’s closing price be?”*
- *“Given the annual precipitation, average temperature, and soil pH, what will this year’s harvest yield be?”*
- *“Given the square footage, number of bedrooms, number of bathrooms, and quality of school district, what will the price of this home be?”*

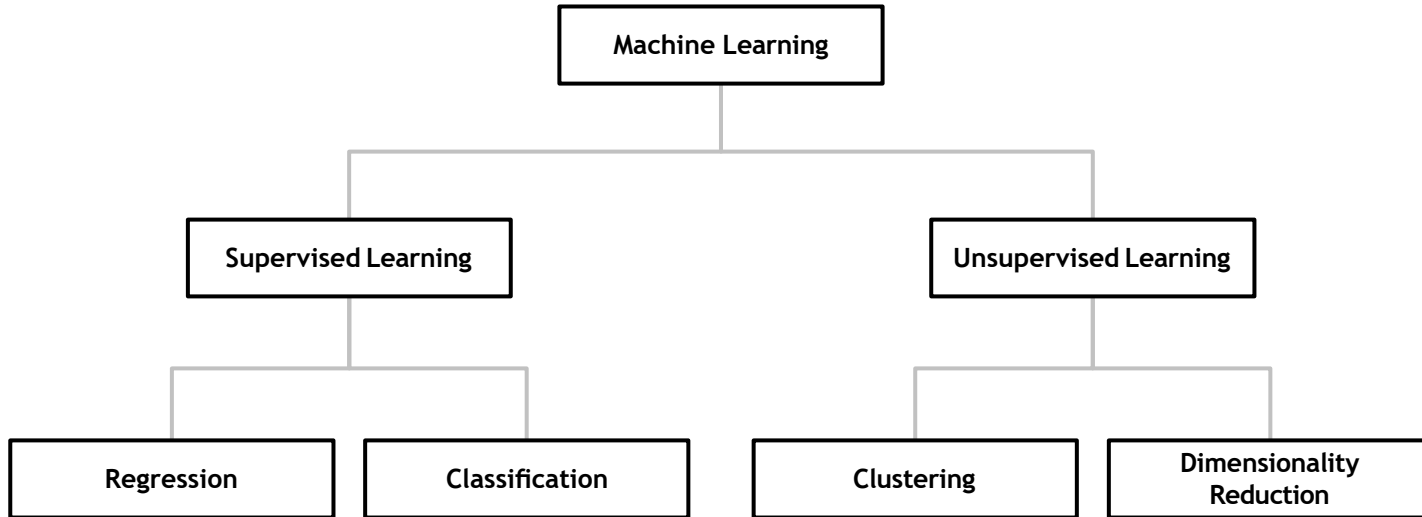
# Two Kinds of Supervised Learning

Supervised learning models fall into two different buckets:

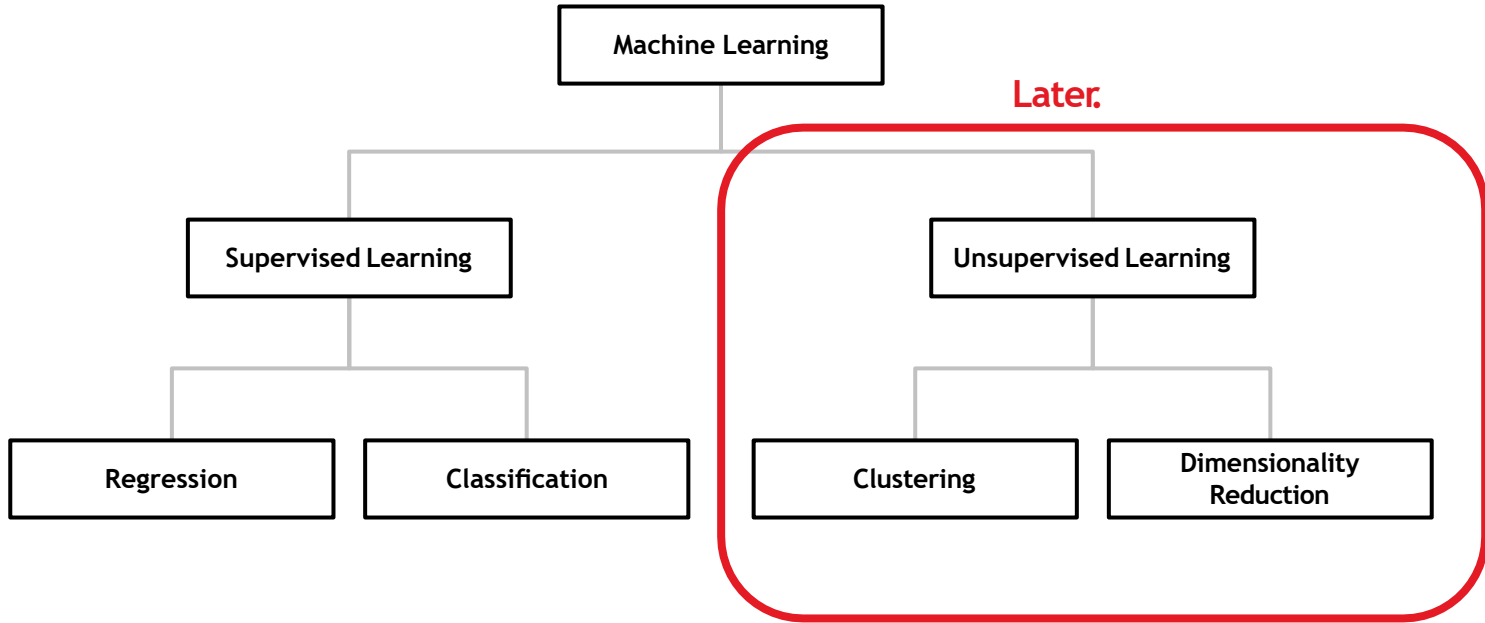
**Classification** - this is when our y-variable is a category. If it's a 0/1 yes/no kind of variable, we often call it **binary classification**. Otherwise, **multiclass classification**.

- *“Given this person’s demographic information, how many tabs they have open, and where they live, will they make a purchase on my site?”*
- *“Given radar readouts, past weather, and almanac data, will it rain tomorrow?”*
- *“Given how many hours you study, how many hours you sleep, and your course load, will you pass the final exam?”*

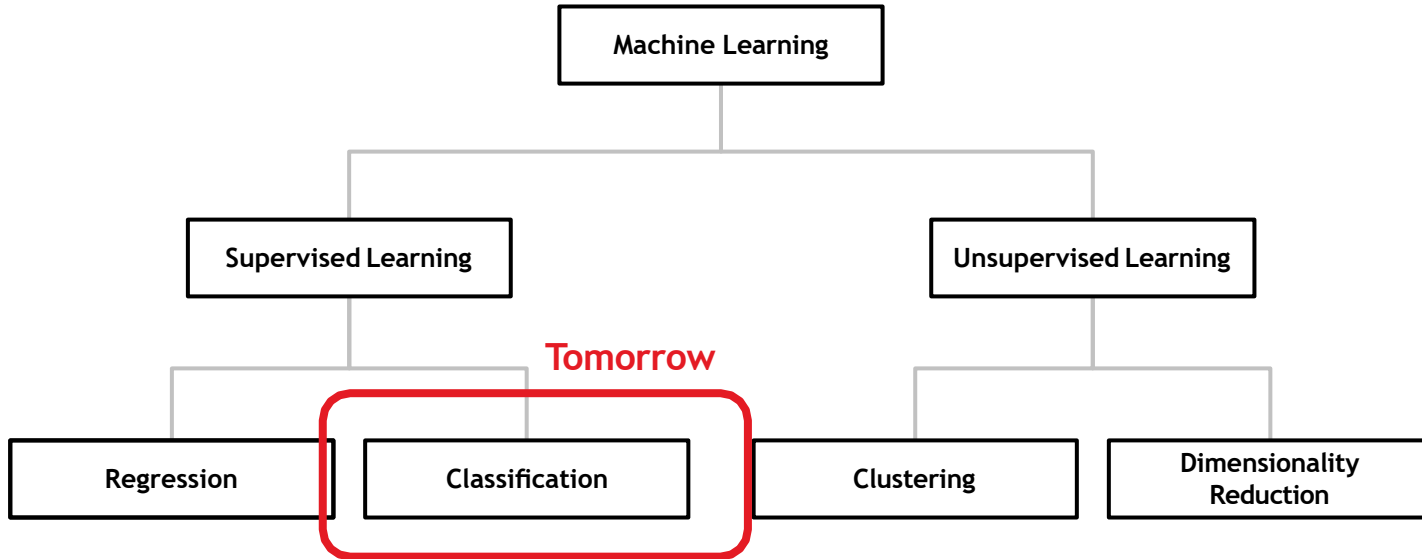
# Roadmap of ML



# Roadmap of ML

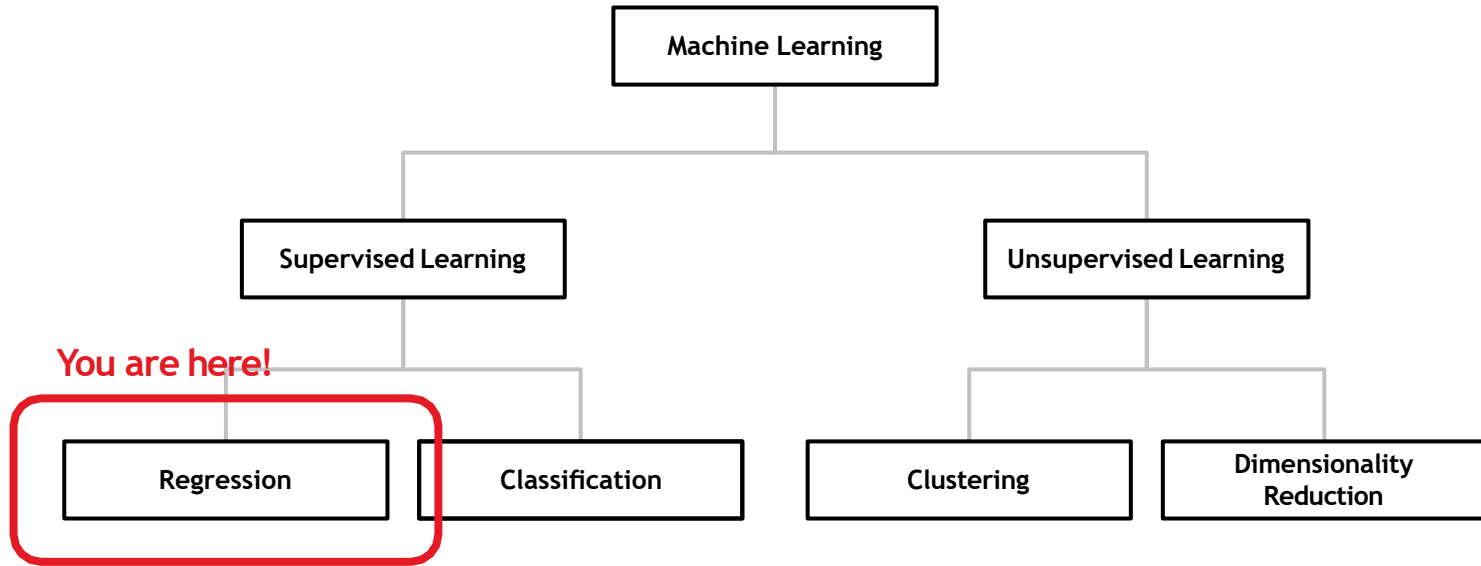


# Roadmap of ML

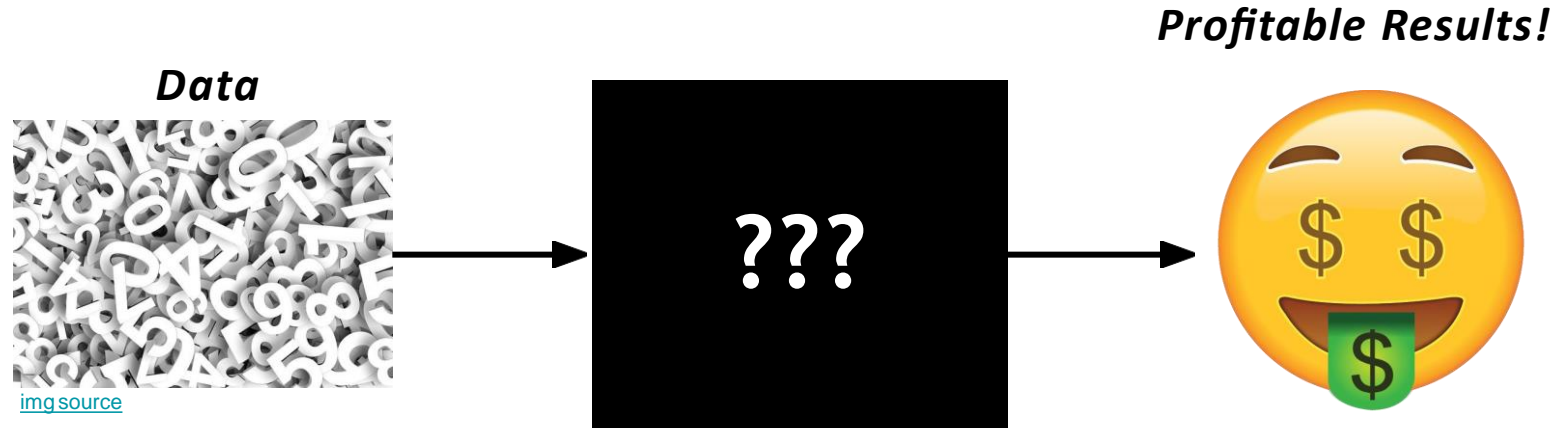




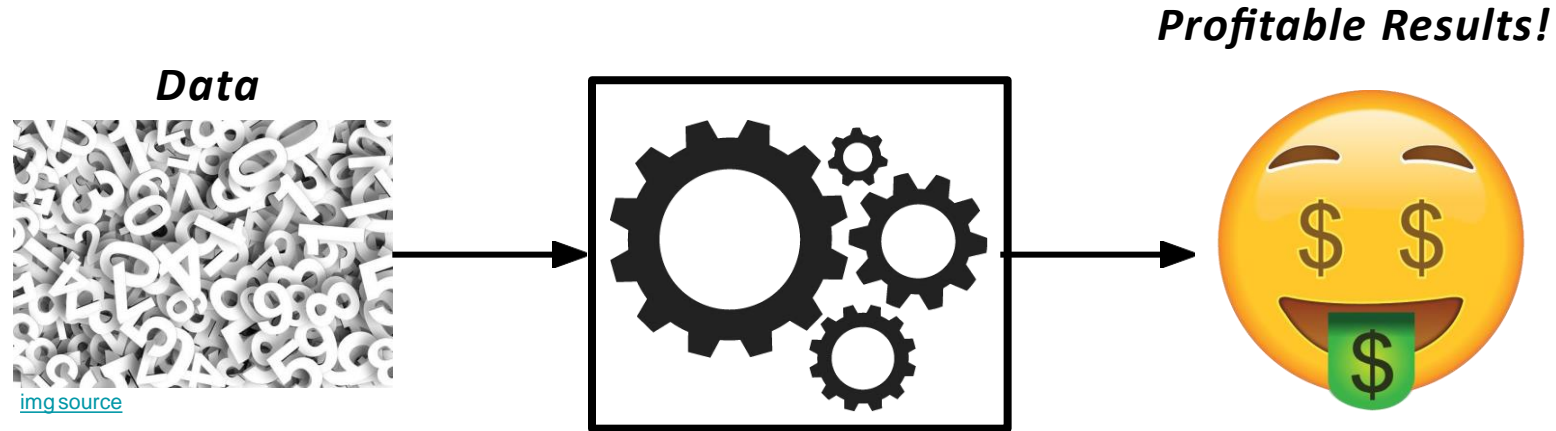
# Roadmap of ML



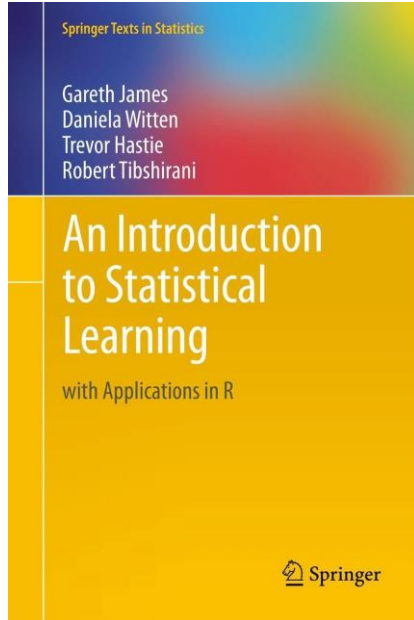
# Supervised Learning Transparency



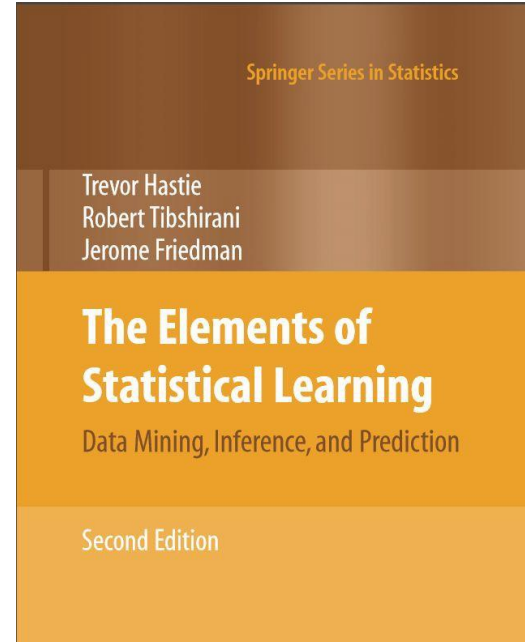
# Supervised Learning Transparency



# The DSI Book Club



***Undergraduate math level, very readable***



***All topics ISL has but at the graduate math level. A few additional chapters.***

---

# Linear Regression

*Supervised, white-box, regression*

# What is linear regression?

Remember this?

$$y = mx + b$$

# What is linear regression?

$$y = mx + b$$

In data science it gets changed to:

$$y = \beta_0 + \beta_1 x_1$$

## What is linear regression?

$$y = \beta_0 + \beta_1 x_1$$

Our model isn't going to be perfect. The things our model doesn't capture are errors and denoted by  $\varepsilon$  (epsilon).

$$y = \beta_0 + \beta_1 x_1 + \varepsilon$$



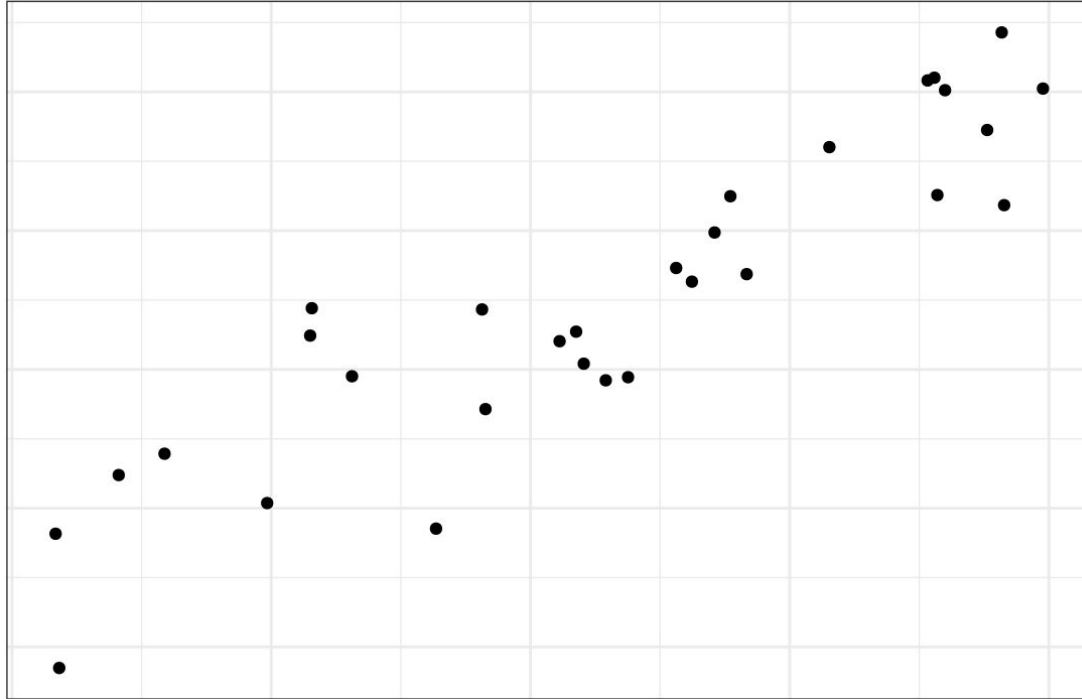
# What is linear regression?

In ordinary least squares linear regression (often just referred to as **OLS**), we try to predict some response variable ( $y$ ) from at least one independent variable ( $x$ ). We believe there is a **linear** relationship between the two:

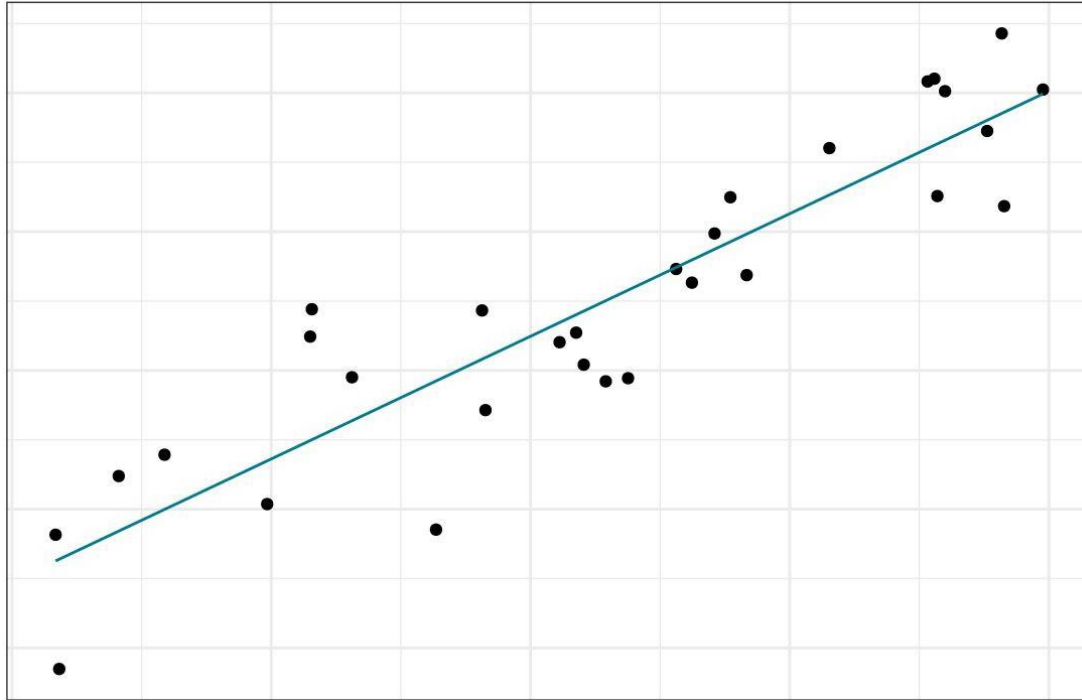
$$y = \beta_0 + \beta_1 x + \varepsilon$$

Again, that funny looking “e” stands for “error” - it’s random noise inherent in our prediction because nothing will be perfect.

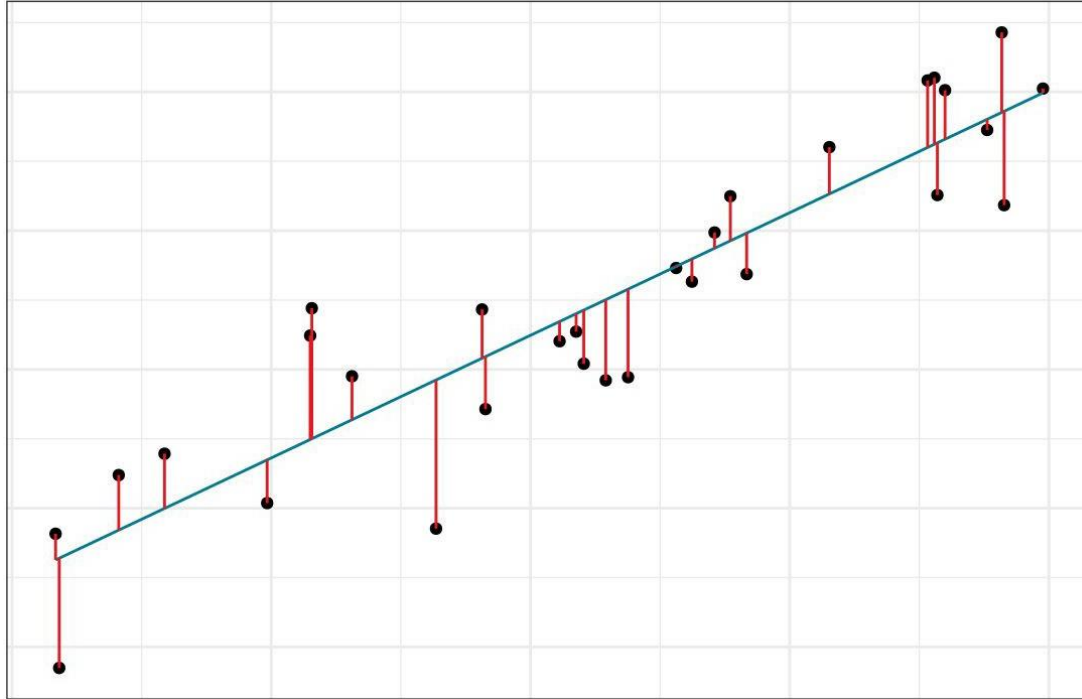
Graphically, this is:



**Graphically, this is:**



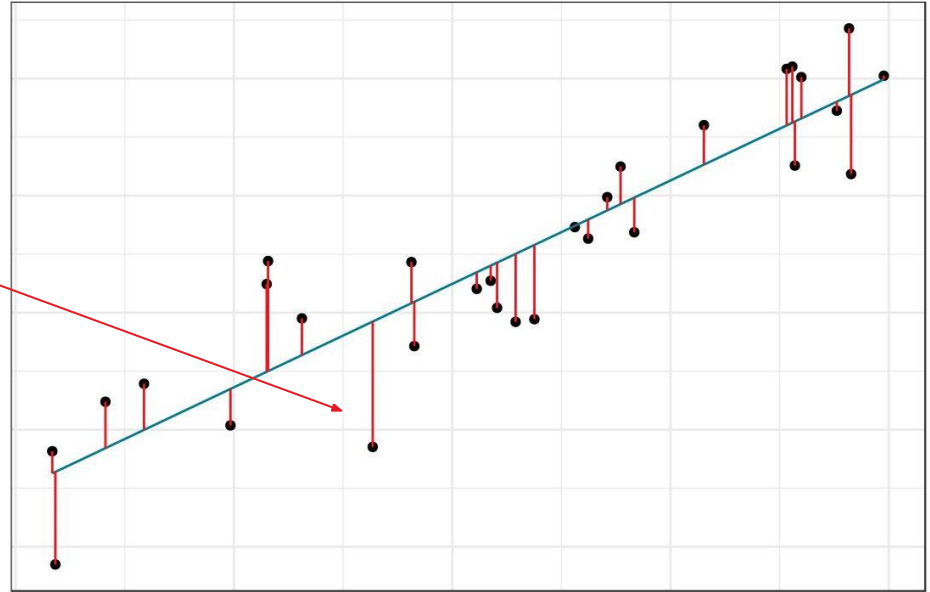
Graphically, this is:



## Graphically, this is:

The difference between the actual and the predicted is called a **residual**, and the line of “best fit” minimizes all of these residuals.

Specifically, we minimize the **sum of the squared residuals**, hence the term “least squares”.



—  
Let's fit one ourselves!



# Notation

A boldface  $\mathbf{y}$  denotes *all* of our response data. It's **bold because it's a vector**. We typically reserve the letter  $n$  to be our **sample size**.

$$\mathbf{y} = (y_1, y_2, \dots, y_n)$$

# Notation

We put “hats” over variables to denote they are **predicted values**. That is, “y-hat” represents the predictions based on our original data.

$$\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$$



## Notation

Similarly, the betas are things we **estimate**, so they get hats too! Our y-hats are the results of using these estimated values to get predictions.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

# Notation

We often need to look at formulas in **summation notation**. Let's discuss these:

$$\sum_{i=1}^4 i$$

$$\sum_{k=2}^4 (k^2 - 1)$$

$$\frac{1}{n} \sum_{i=1}^n x_i$$

# Notation

We often need to look at formulas is **summation notation**. Let's discuss these:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10$$

$$\sum_{k=2}^4 (k^2 - 1)$$

$$\frac{1}{n} \sum_{i=1}^n x_i$$

# Notation

We often need to look at formulas is **summation notation**. Let's discuss these:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10$$

$$\sum_{k=2}^4 (k^2 - 1)$$



$$\frac{1}{n} \sum_{i=1}^n x_i$$

# Notation

We often need to look at formulas is **summation notation**. Let's discuss these:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10$$

$$\sum_{k=2}^4 (k^2 - 1) = 3 + 8 + 15 = 26$$

$$\frac{1}{n} \sum_{i=1}^n x_i$$

# Notation

We often need to look at formulas is **summation notation**. Let's discuss these:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10$$

$$\sum_{k=2}^4 (k^2 - 1) = 3 + 8 + 15 = 26$$

$$\frac{1}{n} \sum_{i=1}^n x_i = \bar{x}$$

The sample  
mean!

# The Residual

Next, we define a **residual** as:

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

This measures how “off” our predictions were. It’s either positive or negative depending on whether we overestimate or underestimate.

# The Sum Squared Error

To get an aggregate measurement of the quality of our model, we often look at the **sum squared error**, or **SSE**:

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

Or more commonly, the **mean squared error (MSE)**:

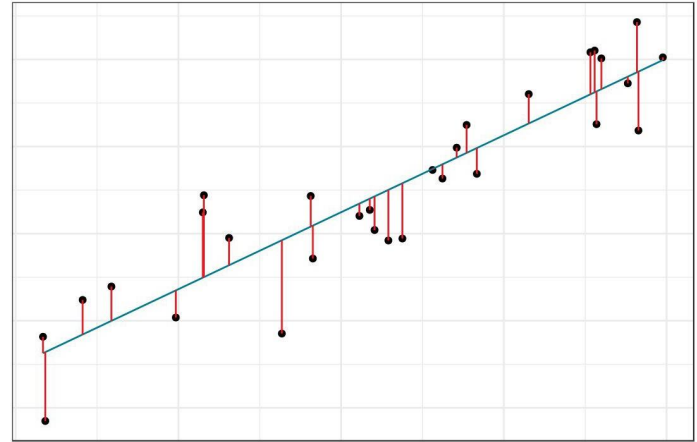
$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum e_i^2$$



# Fitting OLS models

Remember, this is the quantity we actually seek to **minimize** in order to find the best values of our betas!

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$



—  
Let's check out the results!



# — LINE Assumptions



# OLS Assumptions

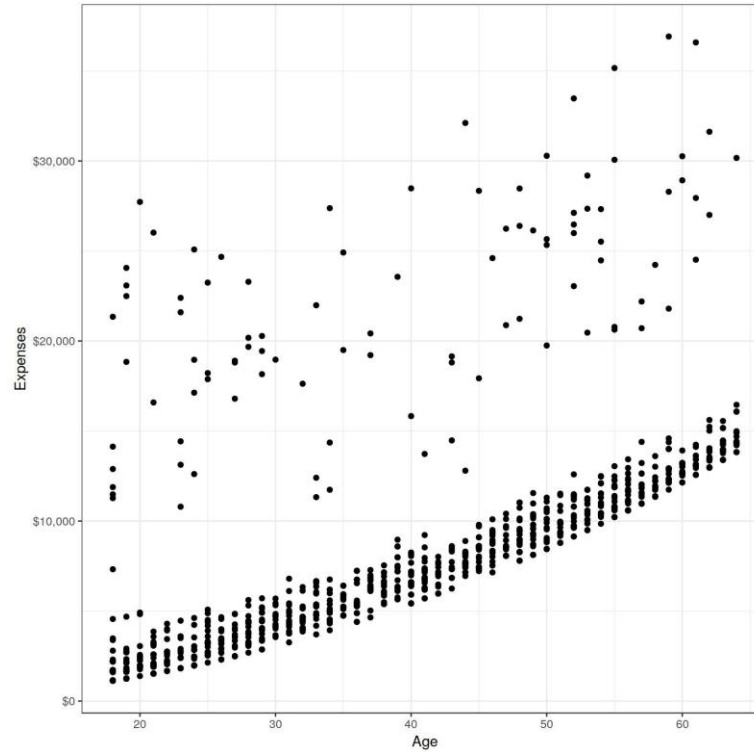
Conducting OLS comes with some pretty steep assumptions that should be satisfied before believing the results. Luckily, there's a nice acronym to remember them:

# OLS Assumptions

Conducting OLS comes with some pretty steep assumptions that should be satisfied before believing the results. Luckily, there's a nice acronym to remember them:

- **L** - Linearity. Relationship between  $x$  and  $y$  should be approximately linear.
- **I** - Independence. Your observations should not affect one another.
- **N** - Normality. Our residuals should be approximately normally distributed.
- **E** - Equal variances, aka “**homoscedasticity**”. Residuals should have approximately equal variances for each  $x$ .

# L is for Linearity



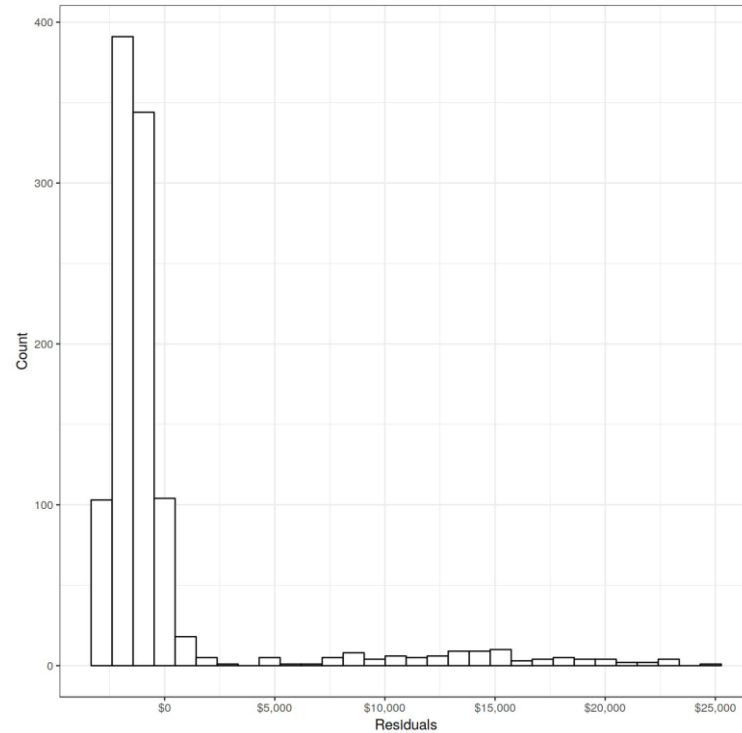
# I is for Independence

Are our samples independent from one another?

**Yes**, these samples were collected independently.

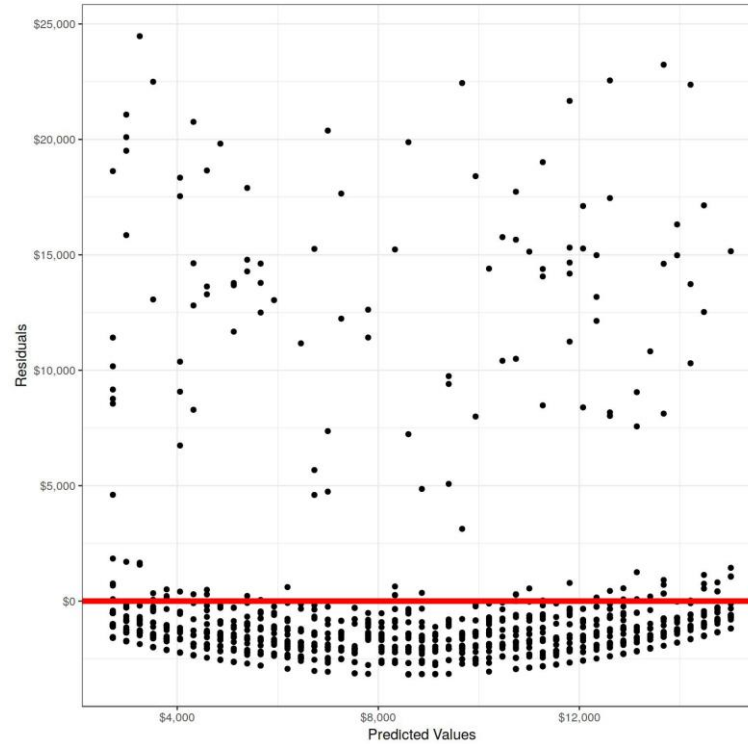
The most common time we'd have to worry about this assumption is when we have **time series data**, when multiple measurements are made on a subject over time.

# N is for Normality





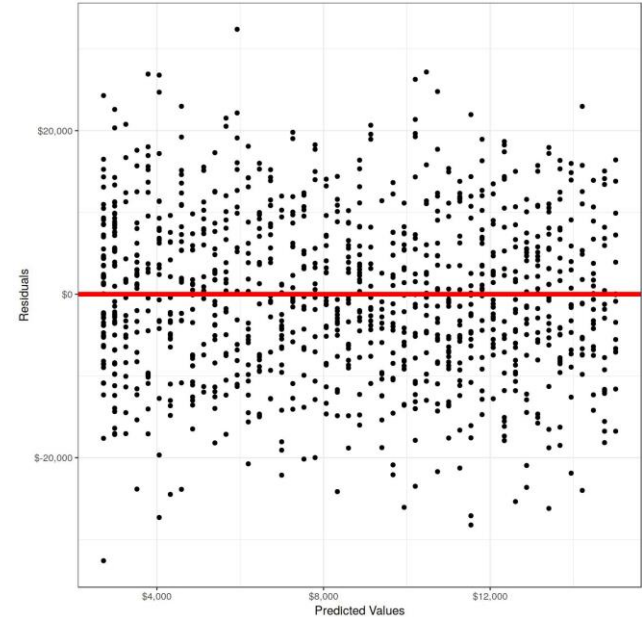
# E is for Equal Variances



# E is for Equal Variance

Yuck! We want to see **absolute randomness** in our residual plots. That is, no pattern whatsoever. We can clearly see a parabolic pattern in our residual plot.

Here's an example of an ideal residual plot.



# What to do if our LINE assumptions are violated?!

A common scenario in linear model is when you have:

- 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”)

One quick fix that should improve all of these issues is doing **log regression**. That is, simply take the natural log of  $y$  before modeling!

—  
Let's see if our model passes!



# — Categorical Features

# Categorical Features

How do we work with categorical variables in our model? We can simply use **binary categorical features** as 0/1 variables.

But what if our variable has more than two **levels**?

First some more notation!

## 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

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

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

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



## Notation: Design Matrix

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

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

Each **column** is a variable.

## Notation: Design Matrix

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

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

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

# One-Hot Encoding

For a categorical variable with  $k$  levels, we need to make one **dummy column** for each:

Season
Summer
Spring
Spring
Fall
Fall
Summer
Winter



$X =$

	W	Sp	Su	F
1	0	0	1	0
1	0	1	0	0
1	0	1	0	0
1	0	0	0	1
1	0	0	0	1
1	0	0	1	0
1	1	0	0	0

# One-Hot Encoding

But wait! This is actually not ok - the intercept term is simply the sum of all of these four columns! In linear algebra terms, this is called being **rank-deficient**, and will make our model impossible to fit.

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

# One-Hot Encoding

So, we next have to **drop any column**. The default in pandas is to drop the first. This means our first column (in this case, Winter) corresponds to the **baseline category**. When interpreting, everything is **relative to this column**.

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

# One-Hot Encoding

So, we next have to **drop any column**. The default in pandas is to drop the first. This means our first column (in this case, Winter) corresponds to the **baseline category**. When interpreting, everything is **relative to this column**.

$$X = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

—  
Let's see it for ourselves!

