Linear Regression

Prof. Murillo
Computational Mathematics, Science and Engineering
Michigan State University

Plan for Next Few Weeks

Oct 9: Today’s lecture: Linear Regression

Oct 16: Doc for webapp link is shared, provide link before our class on Wed

Oct 18: Project presentation

Presentation reports are due at 11:59 PM

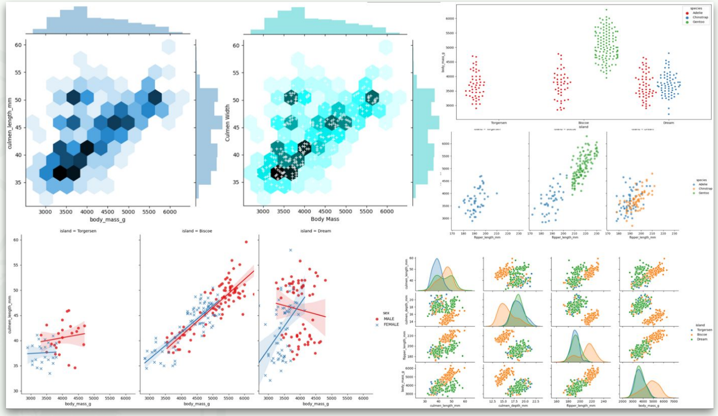
Oct 25: ICA on previous topics

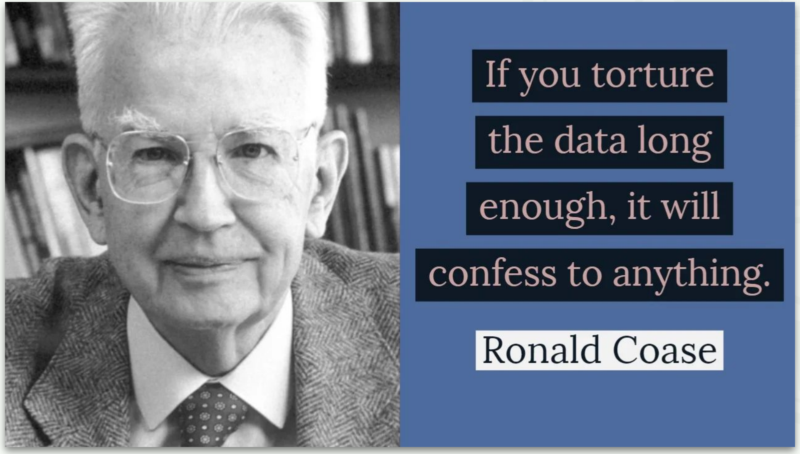
Oct 27: Project material due at 11:59 PM

Oct 30: Lecture: Probability and Statistics (David Butts)

Getting More From Your Data

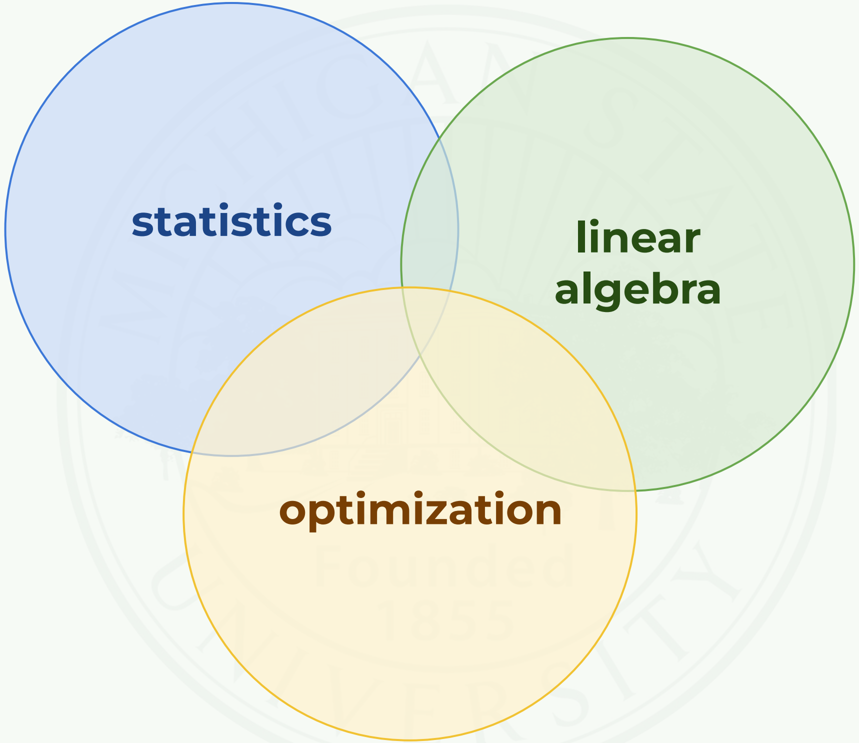
- EDA is great!
- Storytelling is better.



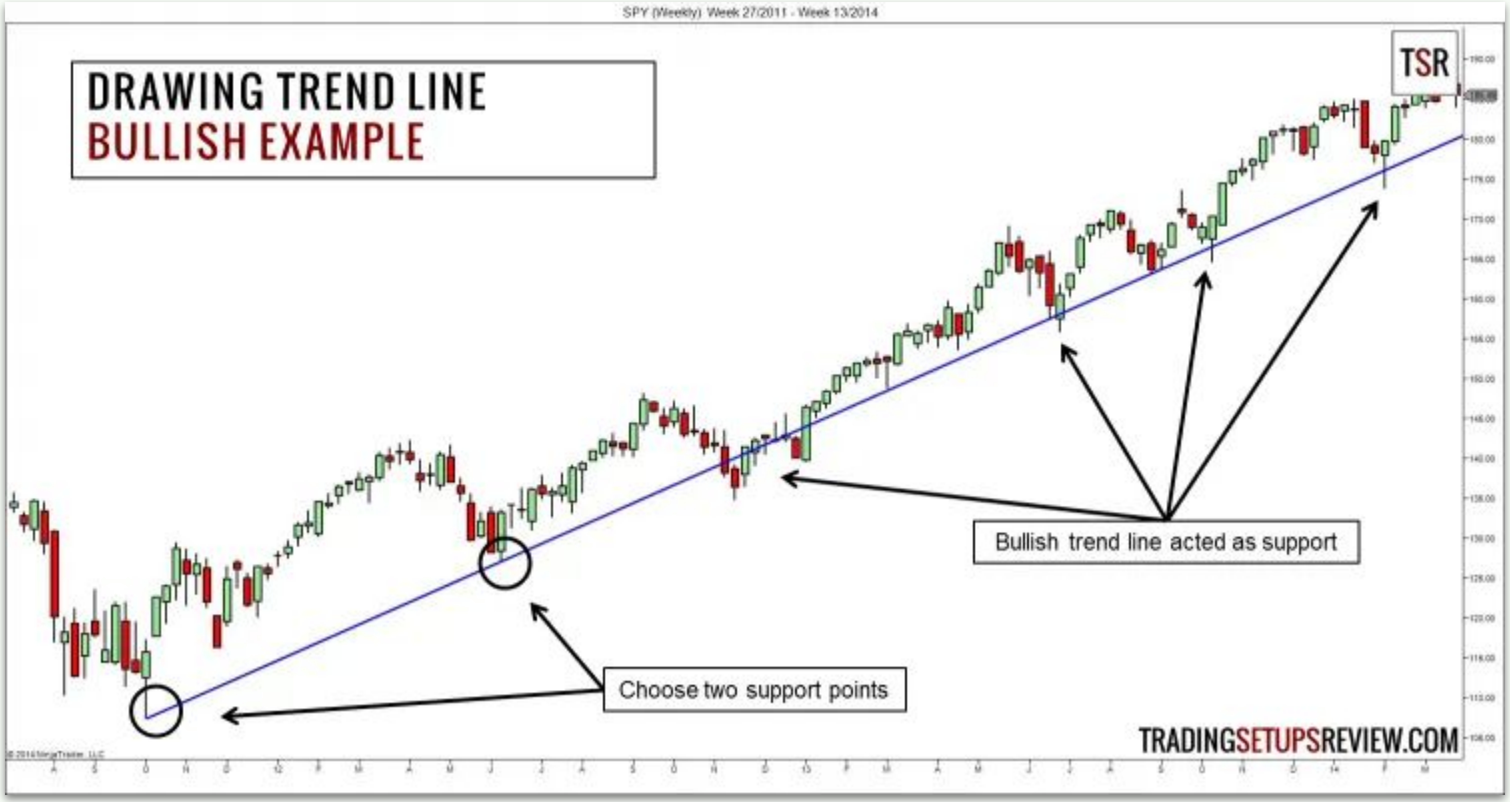


What tools enable one to get even more from the data?

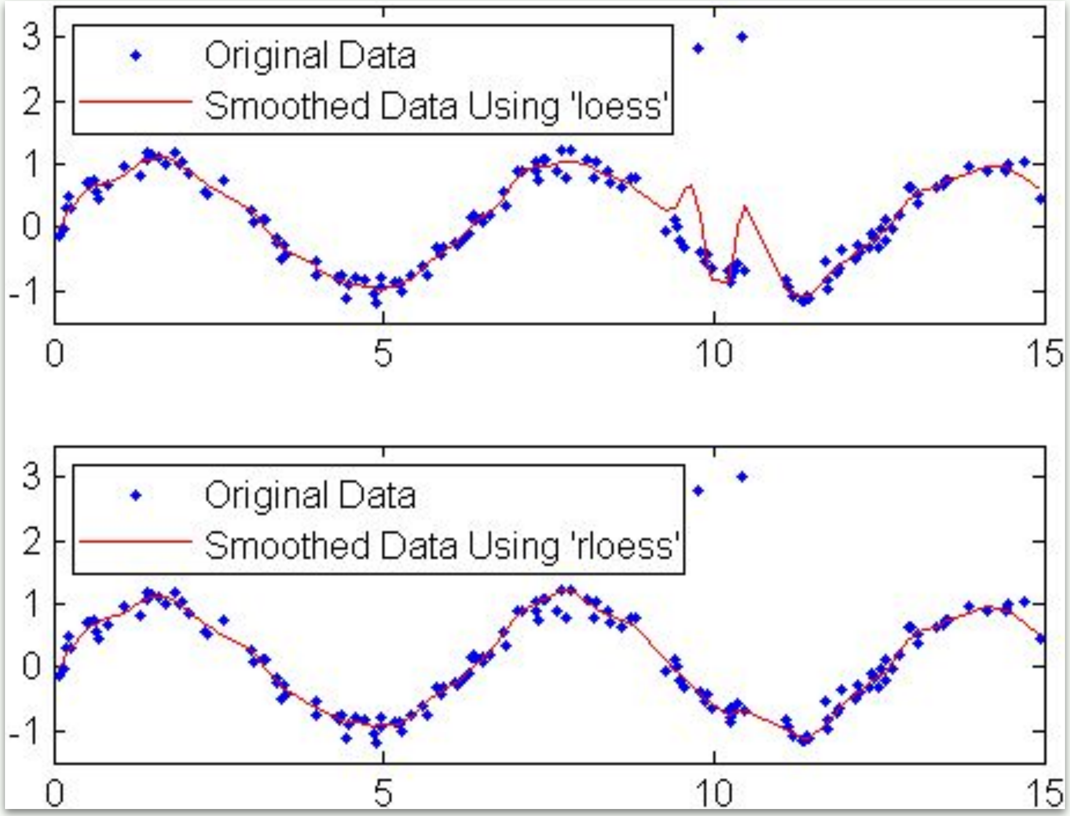
Math!



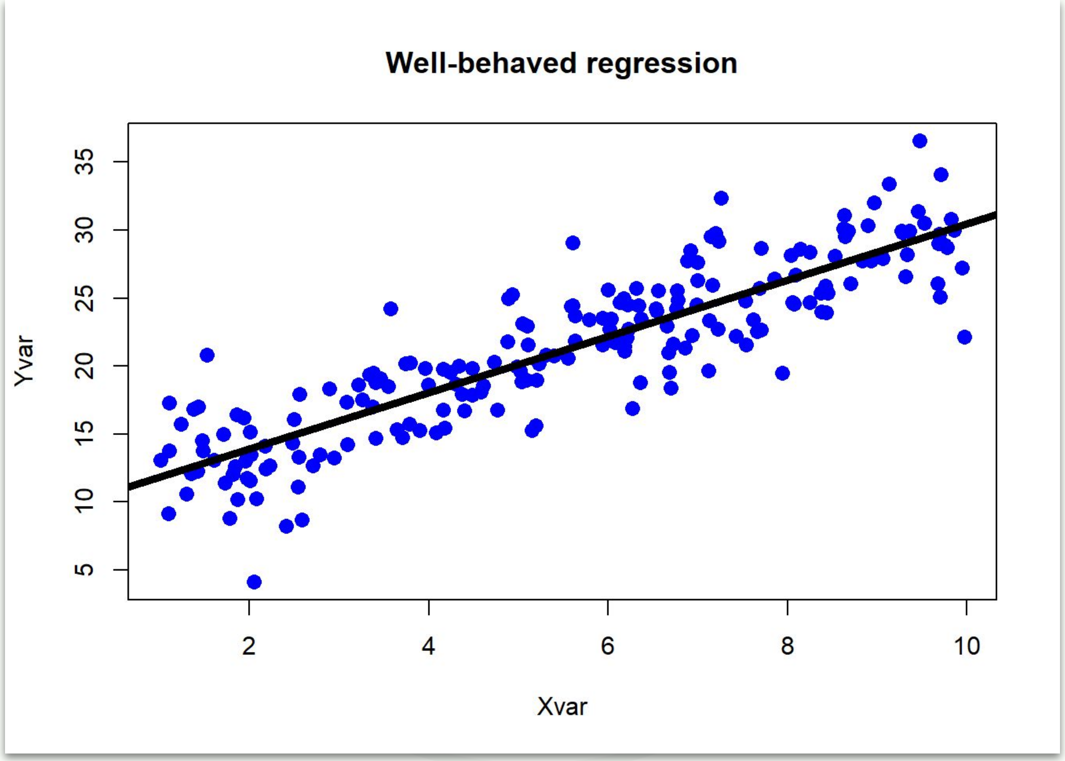
Trend Lines, Smoothing and Regression



Trend Lines, Smoothing and Regression

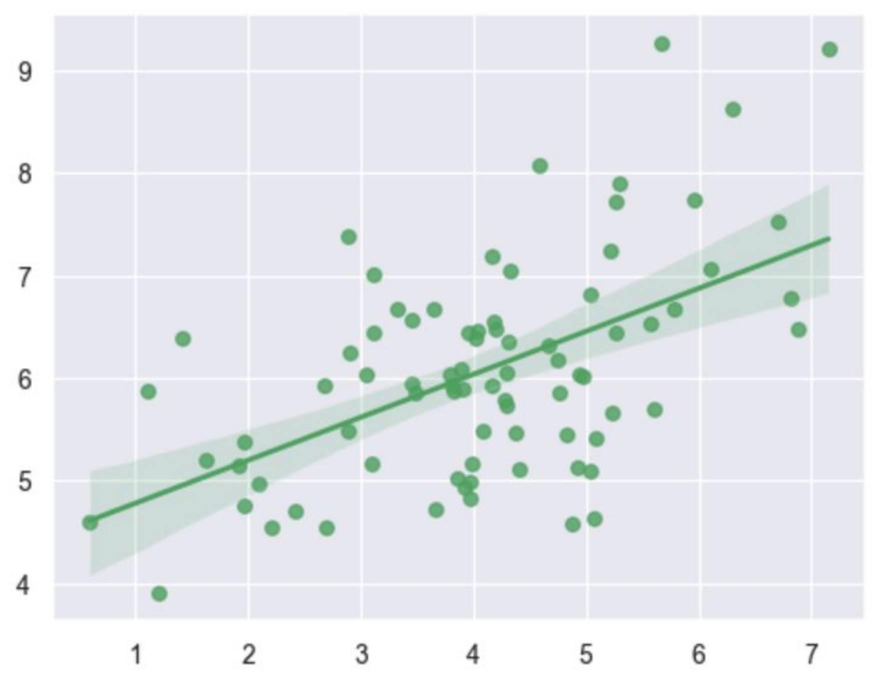


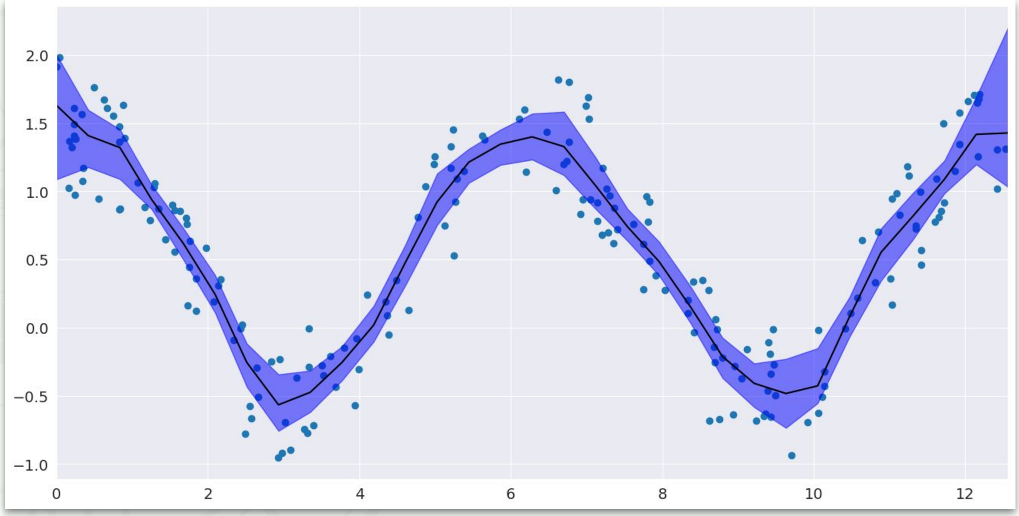
Trend Lines, Smoothing and Regression



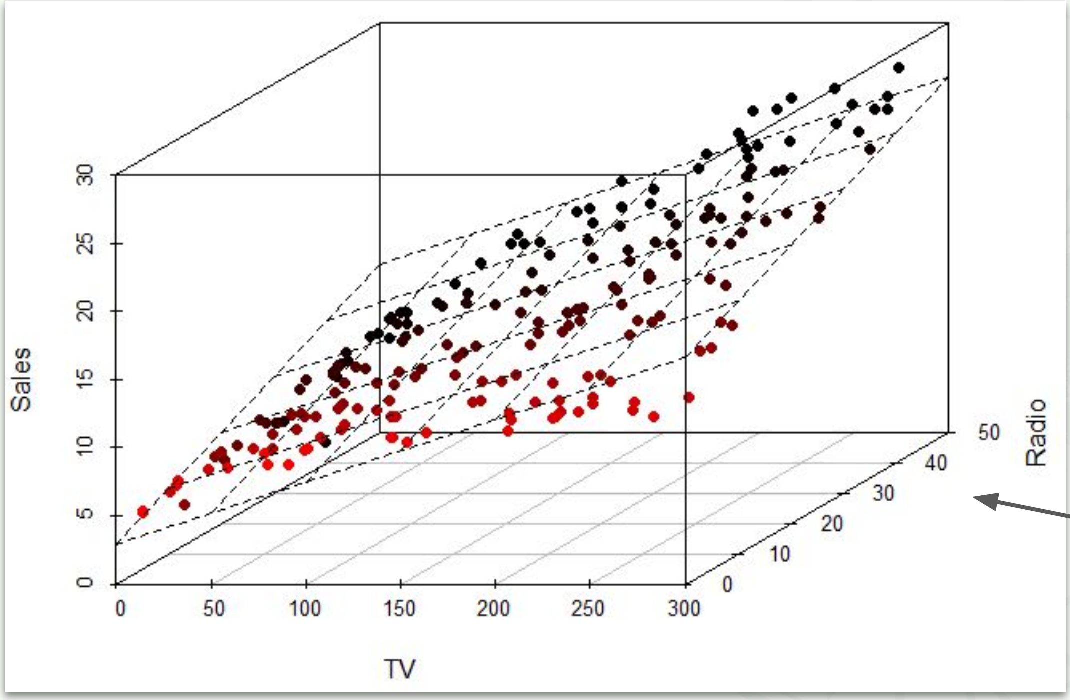
CMSE

Examples: Seaborn and Statsmodels





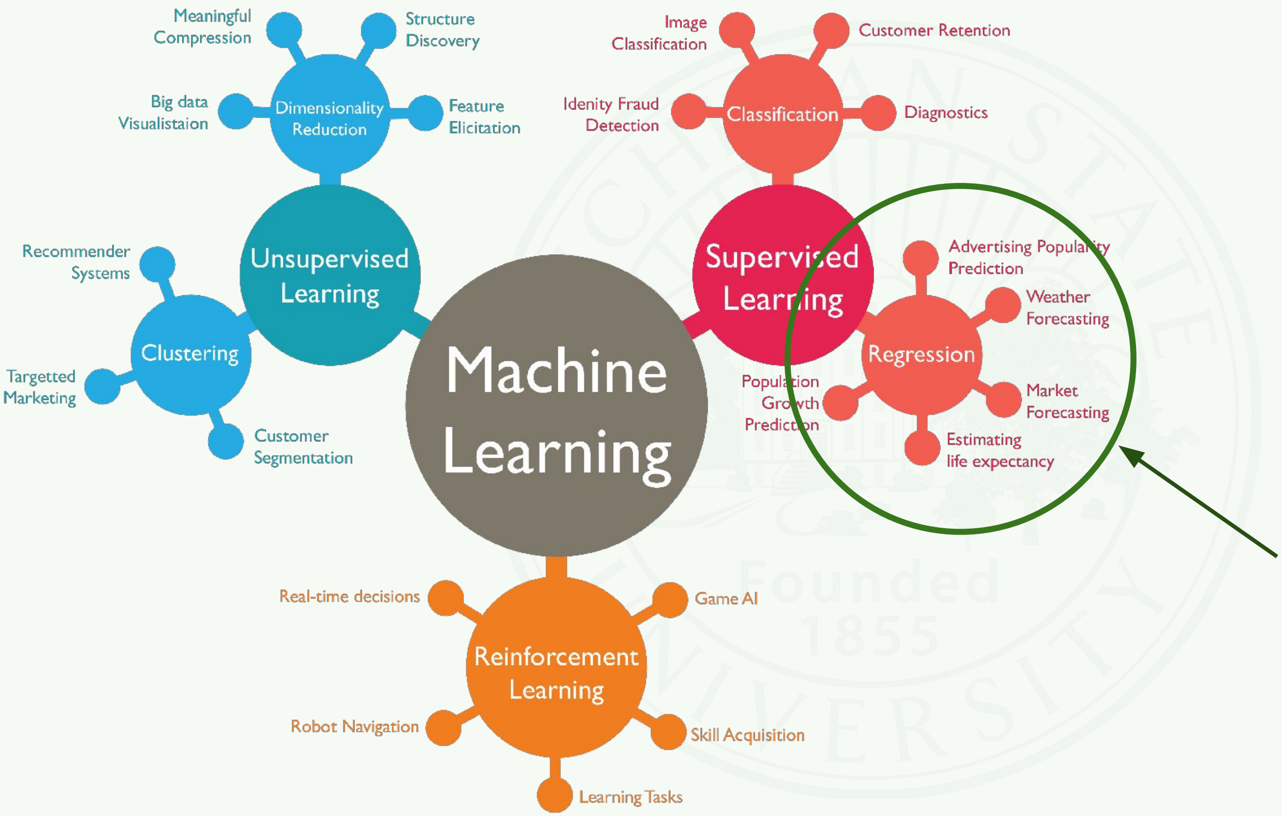
These Examples are the Easy Cases!



In general, you will **not** be able to see the regression.

This is about as high of a dimension as we can plot.

Regression for Prediction: Machine Learning

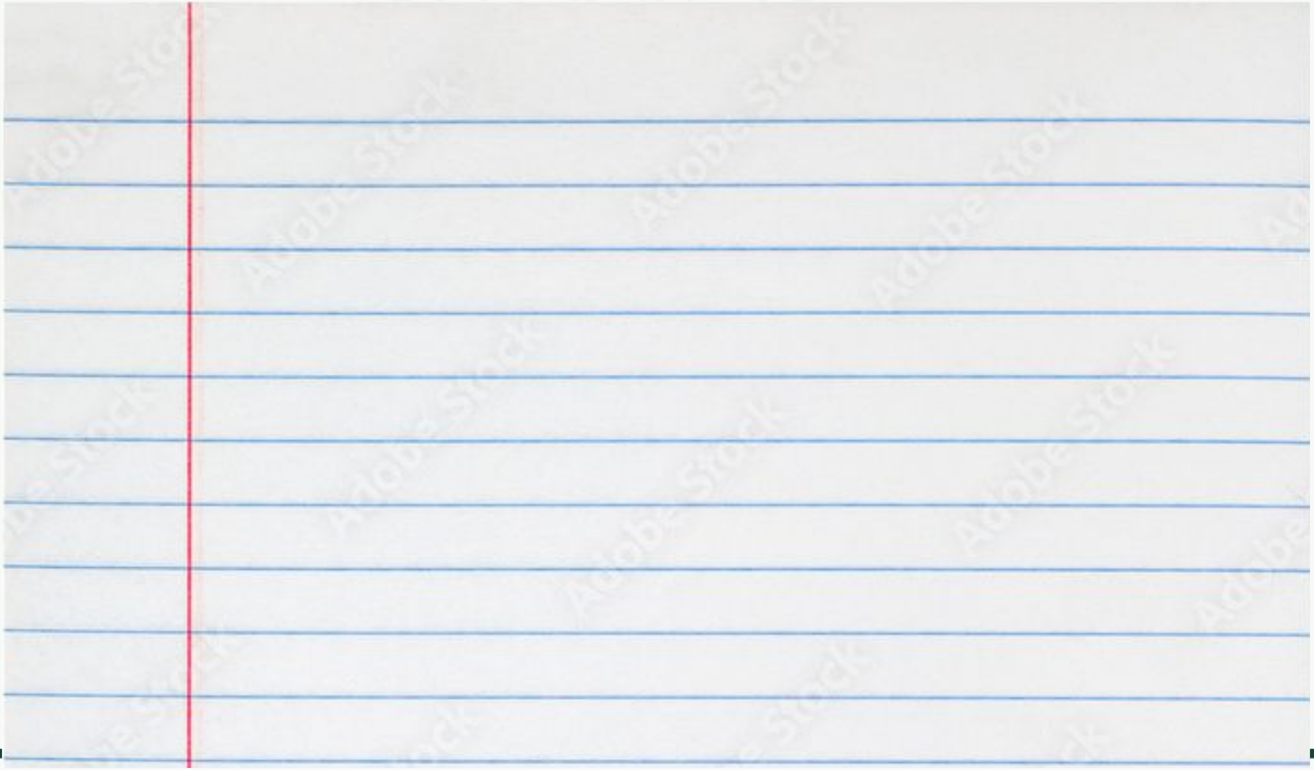


regression branch

CMSE

What is Linear Regression?

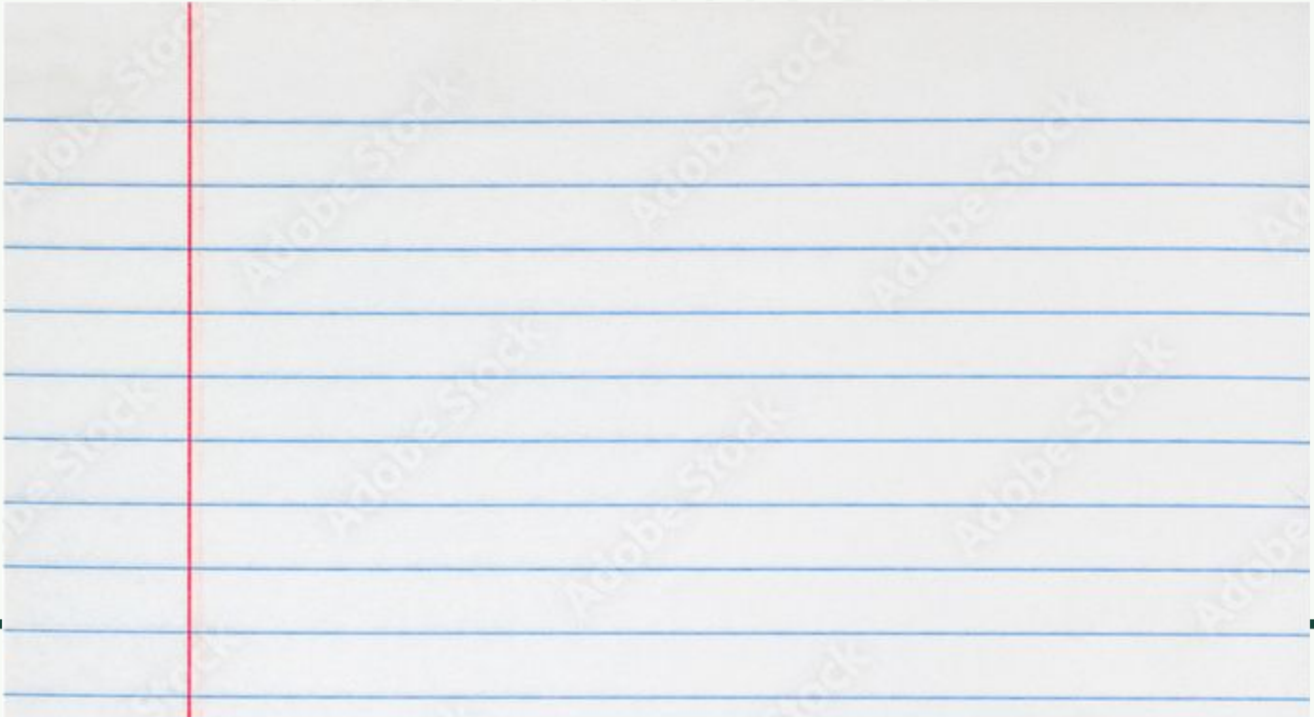
Quiz! Which of these is linear regression?



What is Linear Regression?

Quiz! Which of these is linear regression?

$$y = mx + b,$$

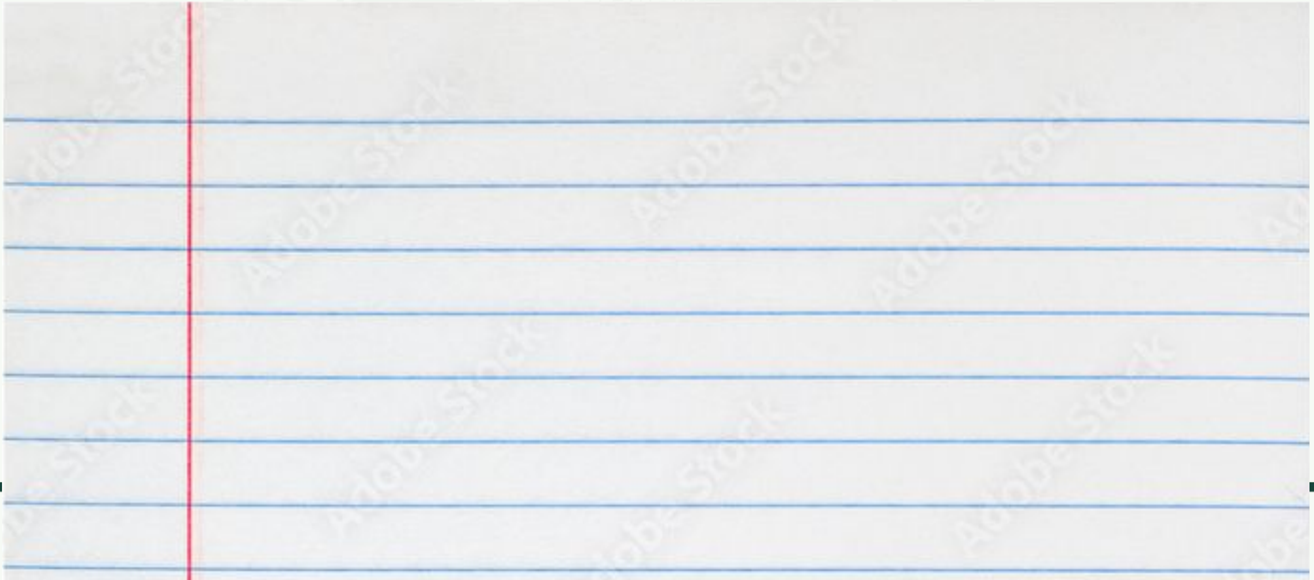


What is Linear Regression?

Quiz! Which of these is linear regression?

$$y = mx + b,$$

$$y = a + bx + cx^2 + dx^3 + \ldots,$$



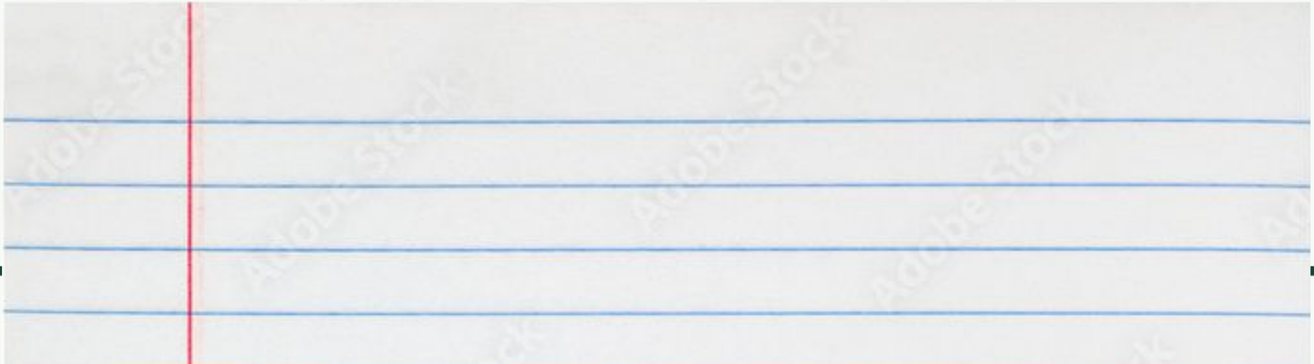
What is Linear Regression?

Quiz! Which of these is linear regression?

$$y = mx + b,$$

$$y = a + bx + cx^2 + dx^3 + \dots,$$

$$f(x) = \sum_d w_d e^{-(x - x_d)^2},$$



What is Linear Regression?

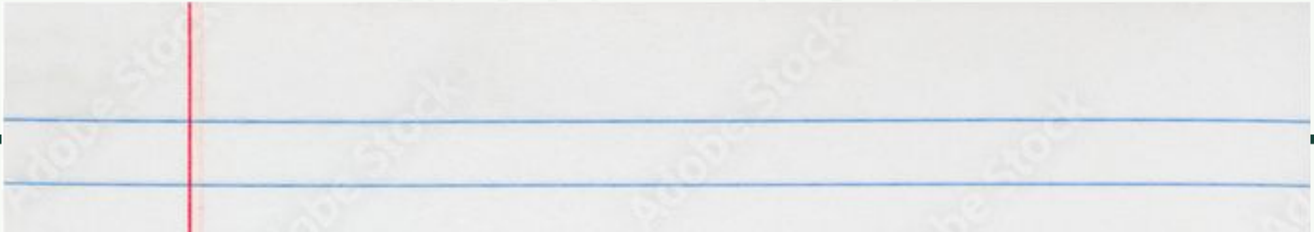
Quiz! *Which of these is linear regression?*

$$y = mx + b,$$

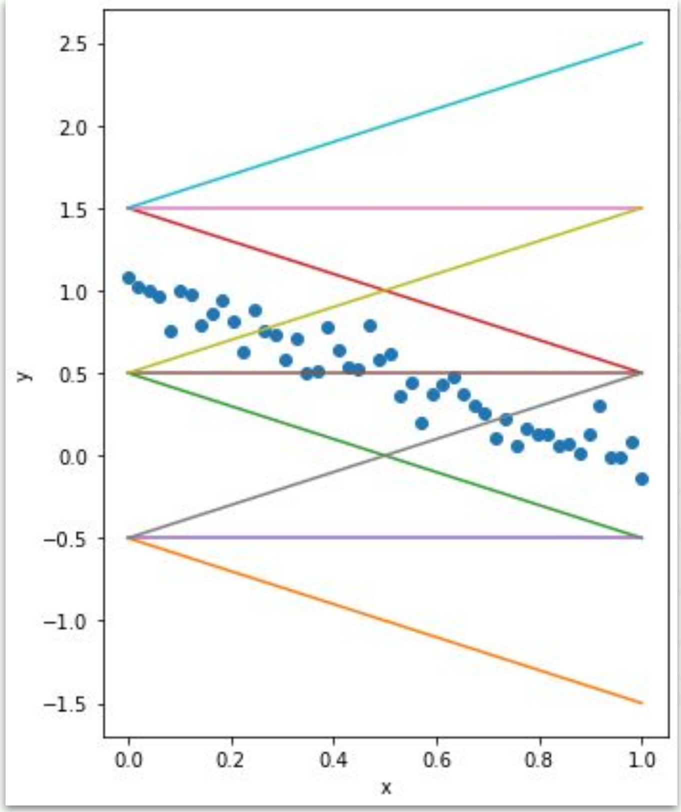
$$y = a + bx + cx^2 + dx^3 + \dots,$$

$$f(x) = \sum_d w_d e^{-(x-x_d)^2},$$

$$p(x_1, x_2) = c\sin(x_1) + d\cos(x_2)$$

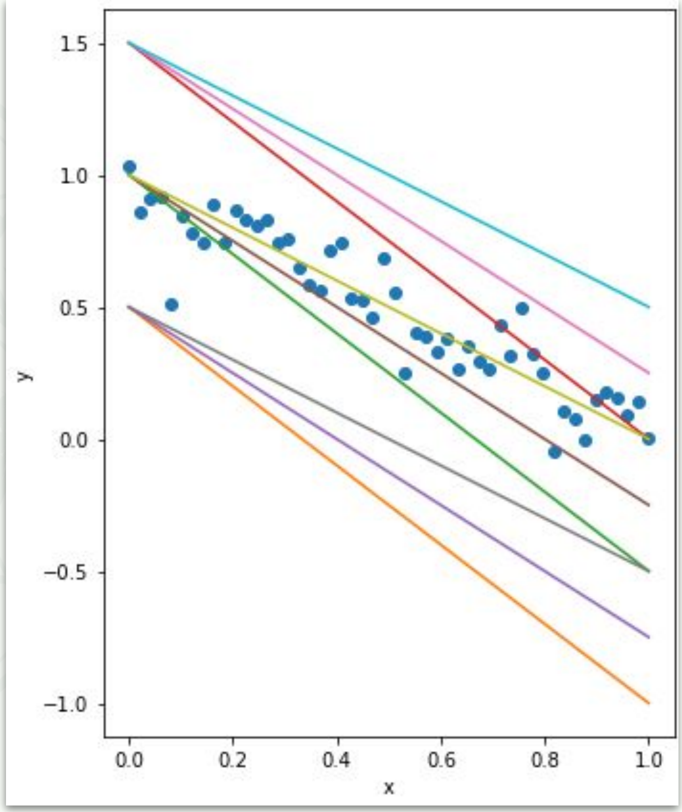


What does is mean to be the “best” line?



Which is these lines is “best”?

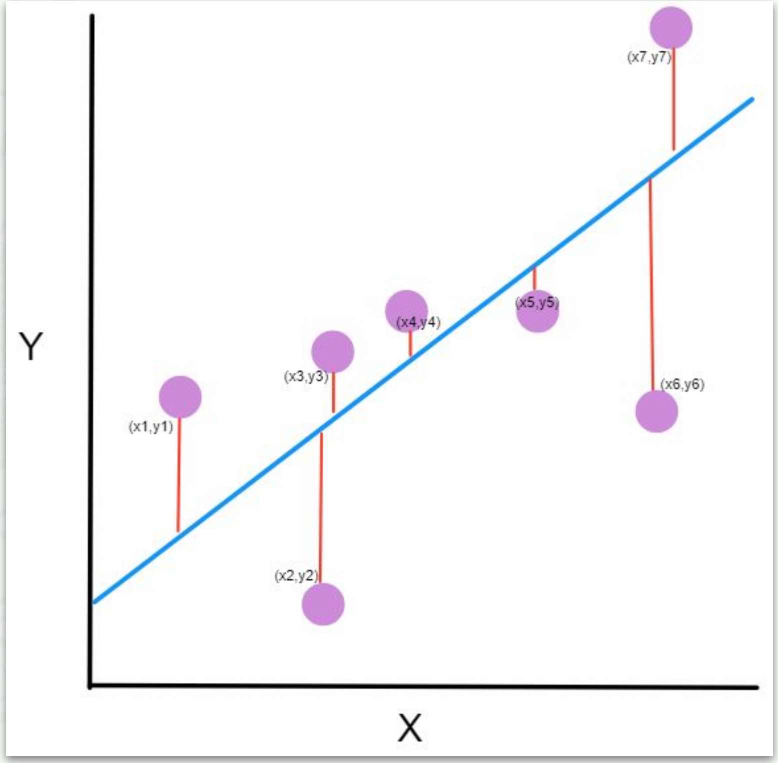
We could iterate until we all agree?



Math is Needed!

- There is no way we would all agree to what “best” means.
- Each of us might define “best” differently for different questions.

Let’s compute the distance from the line to the data points and ensure that this is as small as possible.



CMSE

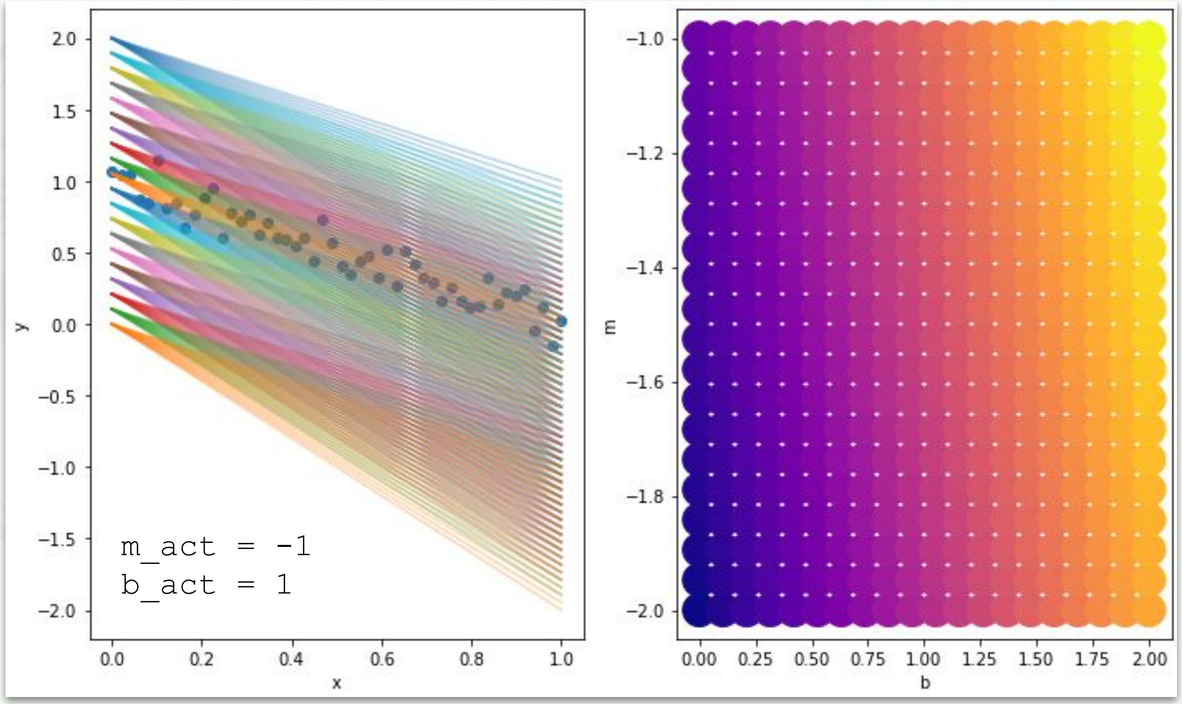
We Can Automate This With a “Grid Search”

$$\mathcal{L}(m,b) = \sum_d (y_d - [mx_d + b])$$

there is no bottom to the error!

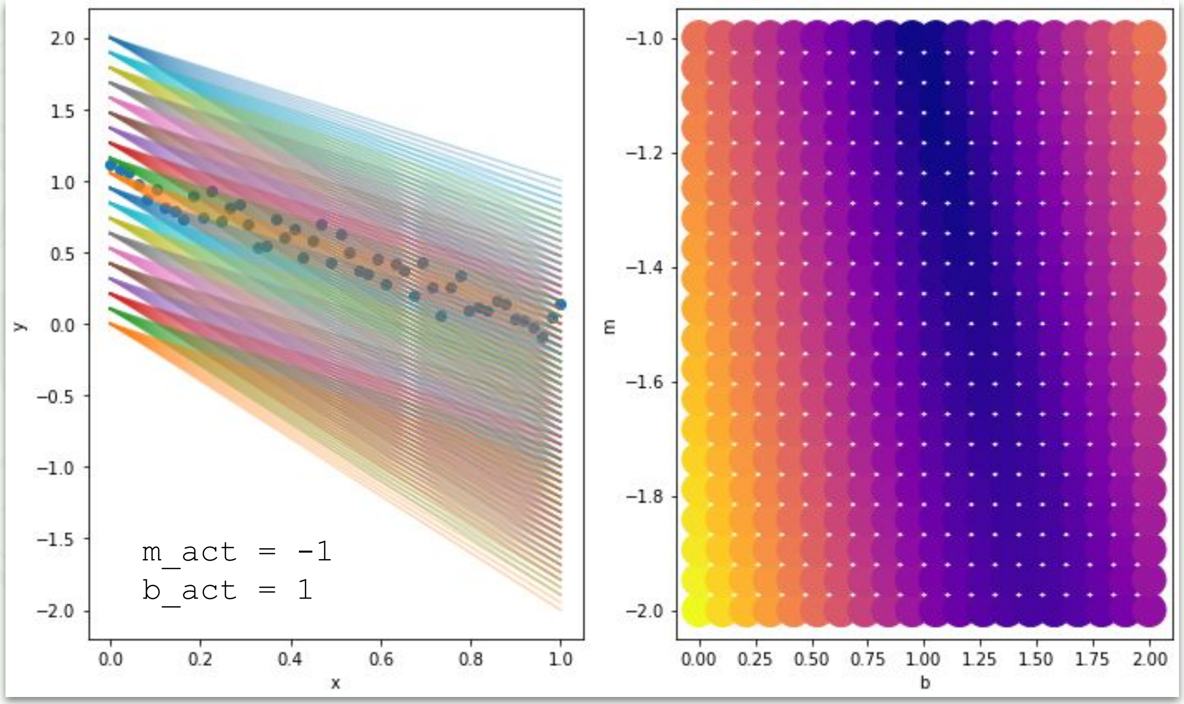
grid search:

- identify the parameters of your model
- make a guess that brackets the values of the parameters
- loop over the parameters
- for each set of parameters, compute the error
- keep track of which one was lowest

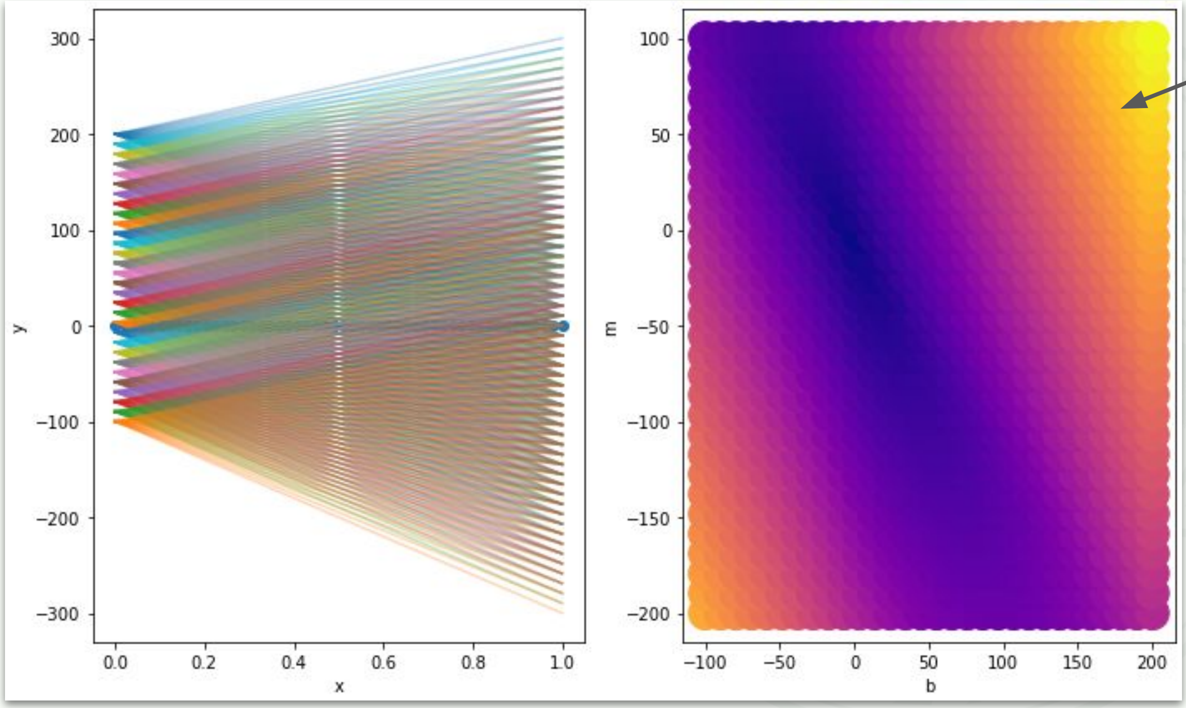


Use MAE (Mean Absolute Error)

$$\mathcal{L}(m, b) = \sum_d |y_d - [mx_d + b]|$$



Use MAE (Mean Absolute Error): Wider Range



$$\mathcal{L}(m, b)$$

Finding the best regression line is an optimization problem.

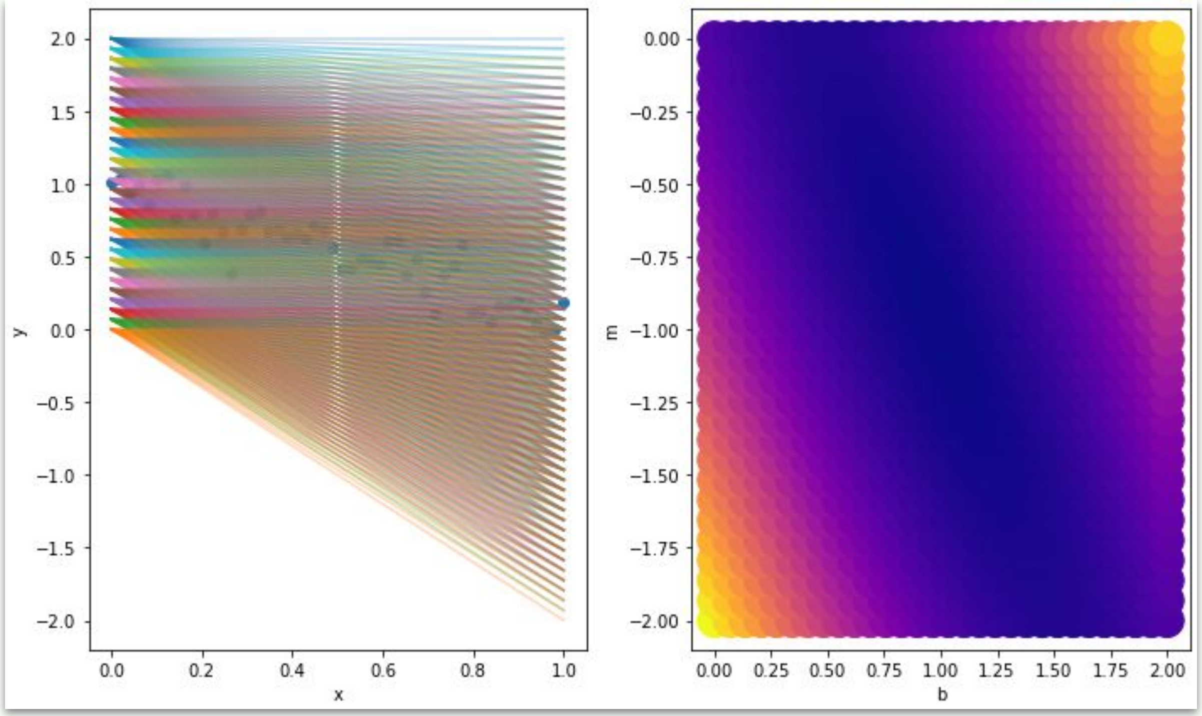
We have defined $L(m,b)$ and we are minimizing it.

This can be done numerically, as shown here, but can also be done using calculus.

However, this is tricky for the MAE.

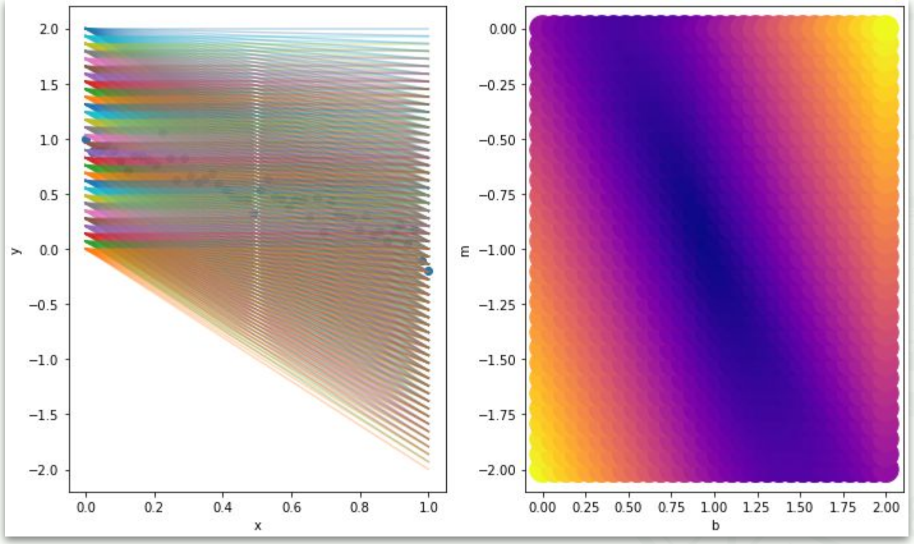
Use MSE (Mean Squared Error)

$$\mathcal{L}(m, b) = \sum_d \left(y_d - [mx_d + b]\right)^2$$

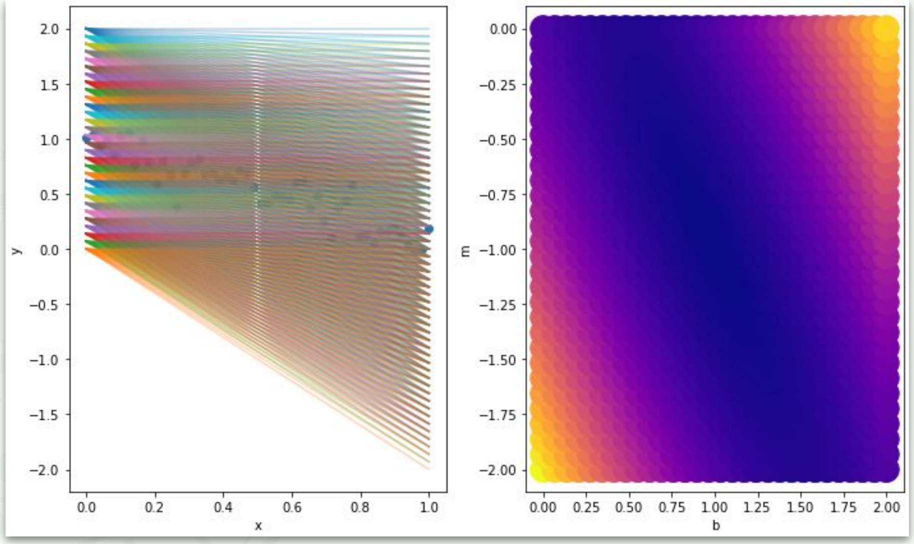


MAE-MSE Comparison

MAE



MSE



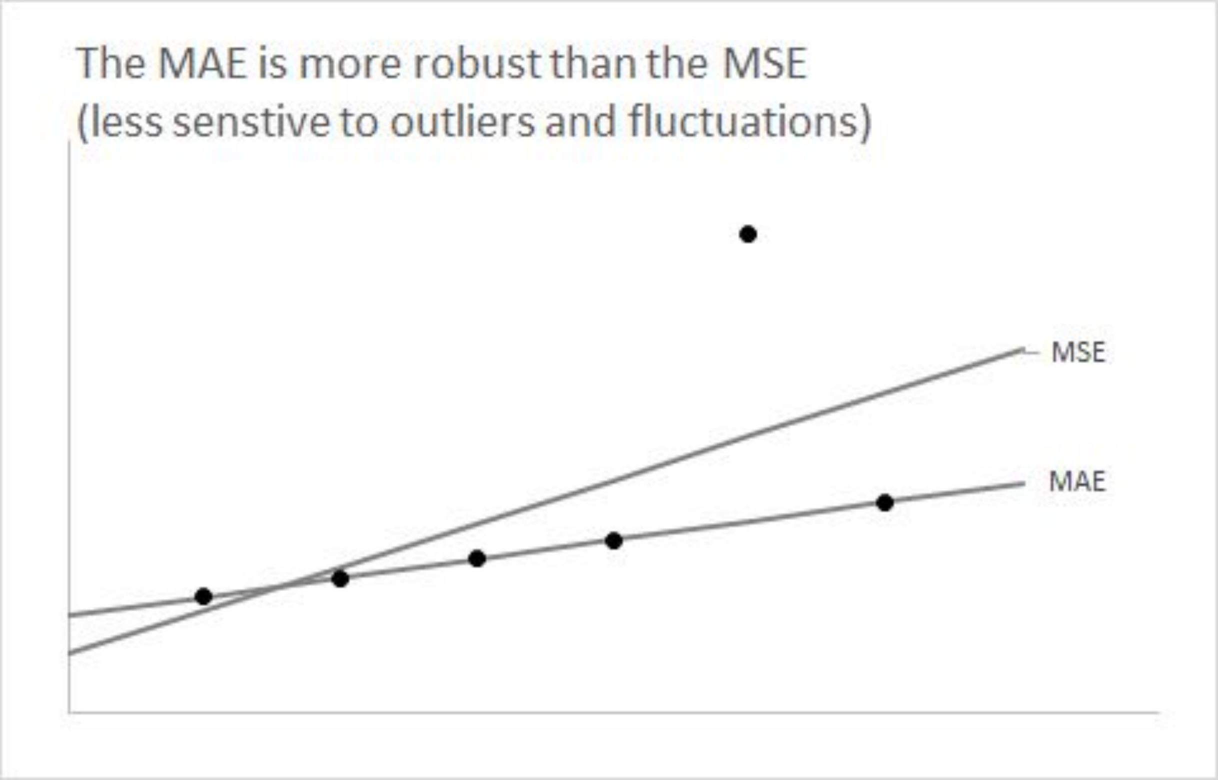
MAE-MSE Comparison

The MSE and MAE **do not** give the same answer!

They perform different tasks.

In *some* cases, you might want the MAE.

In *other* cases, you might want the MSE.



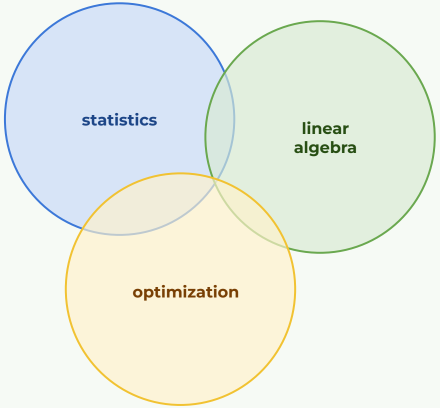
Using **Algebra**, Solve for the MSE Once and For All

$$m = \frac{\operatorname{cov}(X,Y)}{\operatorname{cov}(X,X)},$$

$$= \frac{\mathrm{E}[X,Y] - \mathrm{E}[X]\mathrm{E}[Y]}{\mathrm{E}[X,X] - \mathrm{E}[X]^2},$$

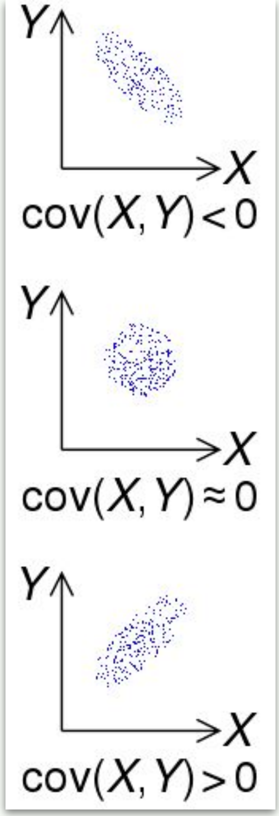
$$b = \mathrm{E}[Y] - m\mathrm{E}[X]$$

The regression is solved in terms of quantities we find in **statistics**.

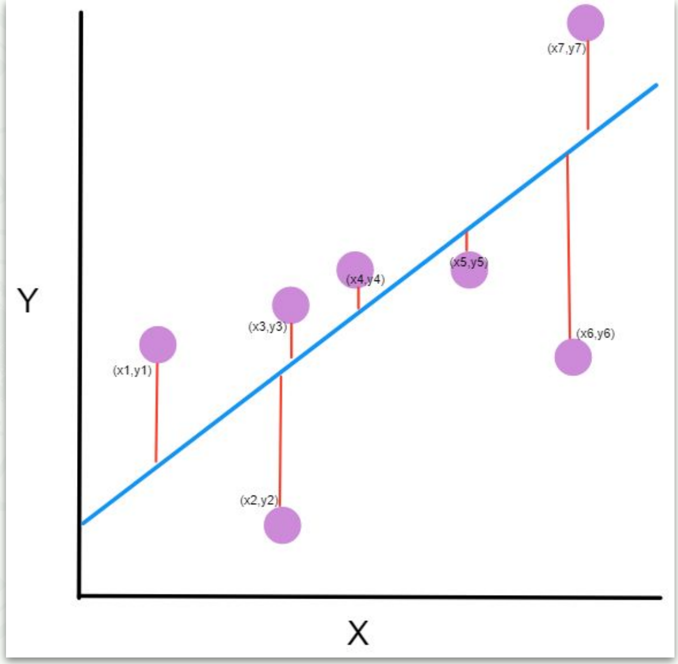


Prediction is cast as an **optimization** problem, which we solve using **algebra** to reveal that predictions are made from **statistics** of the data.

Using **Algebra**, Solve for the MSE Once and For All



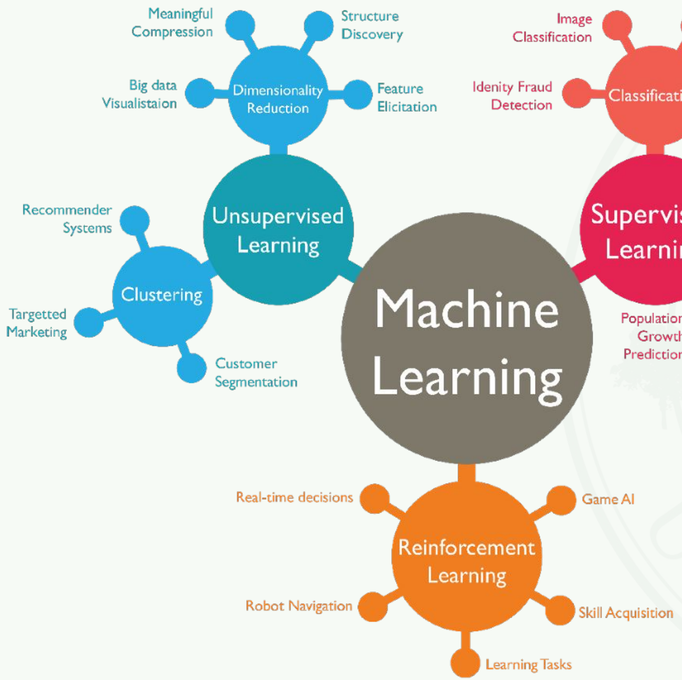
$$m = \frac{\mathrm{cov}(X,Y)}{\mathrm{cov}(X,X)},$$
$$= \frac{\mathrm{E}[X,Y] - \mathrm{E}[X]\mathrm{E}[Y]}{\mathrm{E}[X,X] - \mathrm{E}[X]^2},$$
$$b = \mathrm{E}[Y] - m\mathrm{E}[X]$$



Recap: Where are we so far?

- Using linear regression (LR) we can find the best fit line, which can be used as a trend, to smooth the data or to make predictions.
- There are many ways to define best, but using the MSE is convenient as it results in a closed-form (analytic) expression.
- Minimizing the MSE, an optimization problem, and solving the algebraic equations that result, yields predictions in terms of the statistical properties of the data.

Regression



independent variable

data matrix

$$y = f(\mathbf{x}, X),$$

$$y = f(\mathbf{x}, \mathbf{w})$$

dependent variable

parameters

Instance-Based Learning

independent variable

data matrix

$$y = f(\mathbf{x}, X),$$

$$y = f(\mathbf{x}, \mathbf{w})$$

dependent variable

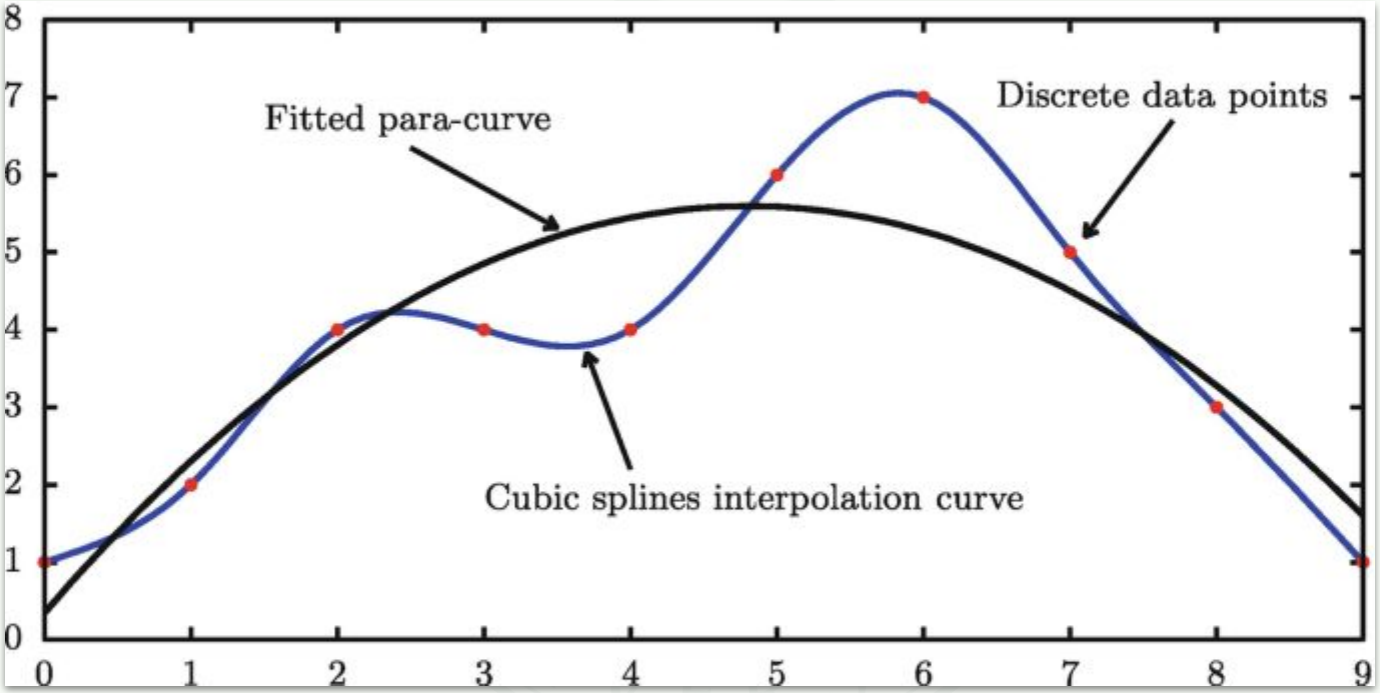
parameters

This type is referred to as “instance based” because instances of the data are needed to make predictions.

This approach can be very accurate, but comes with potentially problems:

- dataset could be extremely large
- data could be proprietary

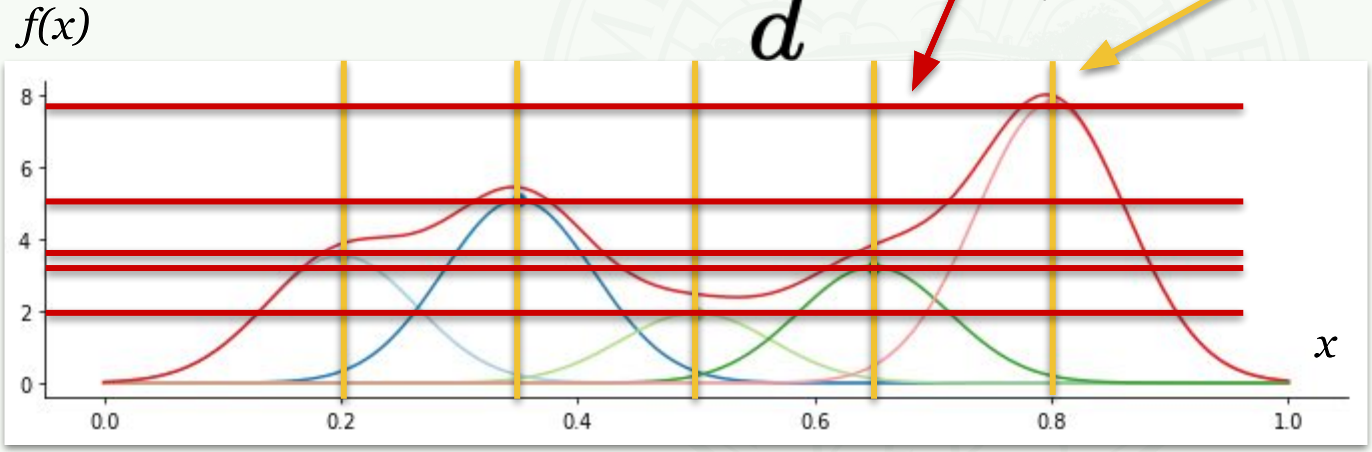
Fitting, Overfitting, Interpolation



Radial Basis Function Neural Networks

$$f(x) = \sum_d w_d e^{-(x - x_d)^2}$$

not literally

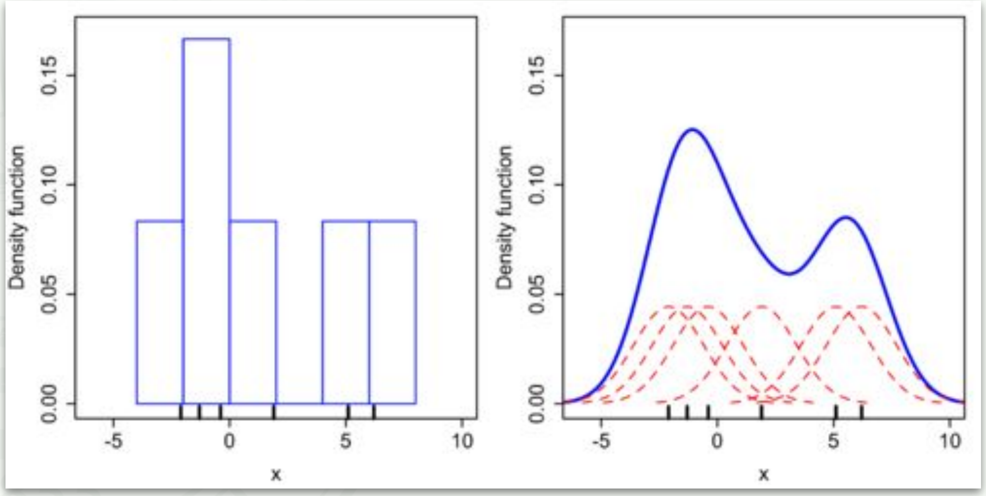


The x_d are literally the positions of the data points.

The weights w_d are not literally the heights: *we need to solve for them* so that they add up correctly.

Kernels Appear Everywhere!

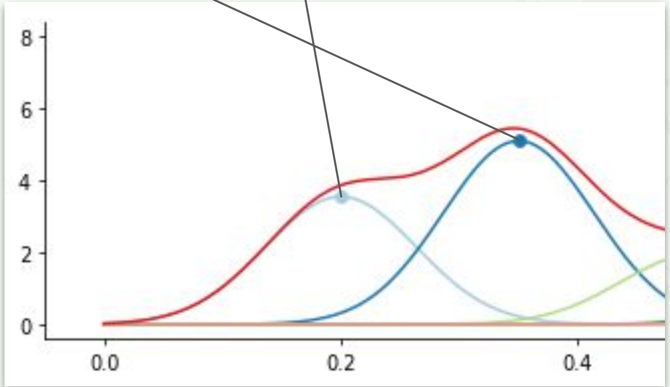
Don't confuse RBF-NNs with KDEs!



As Written, RBF-NNs is a Linear Regression Problem

$$f(x_1) = w_1 e^{-(x_1 - x_1)^2} + w_2 e^{-(x_1 - x_2)^2},$$

$$f(x_2) = w_1 e^{-(x_2 - x_1)^2} + w_2 e^{-(x_2 - x_2)^2}$$



We have two equations in two unknowns.

Everything else is known.

ICA: Code RBF-NN by Hand

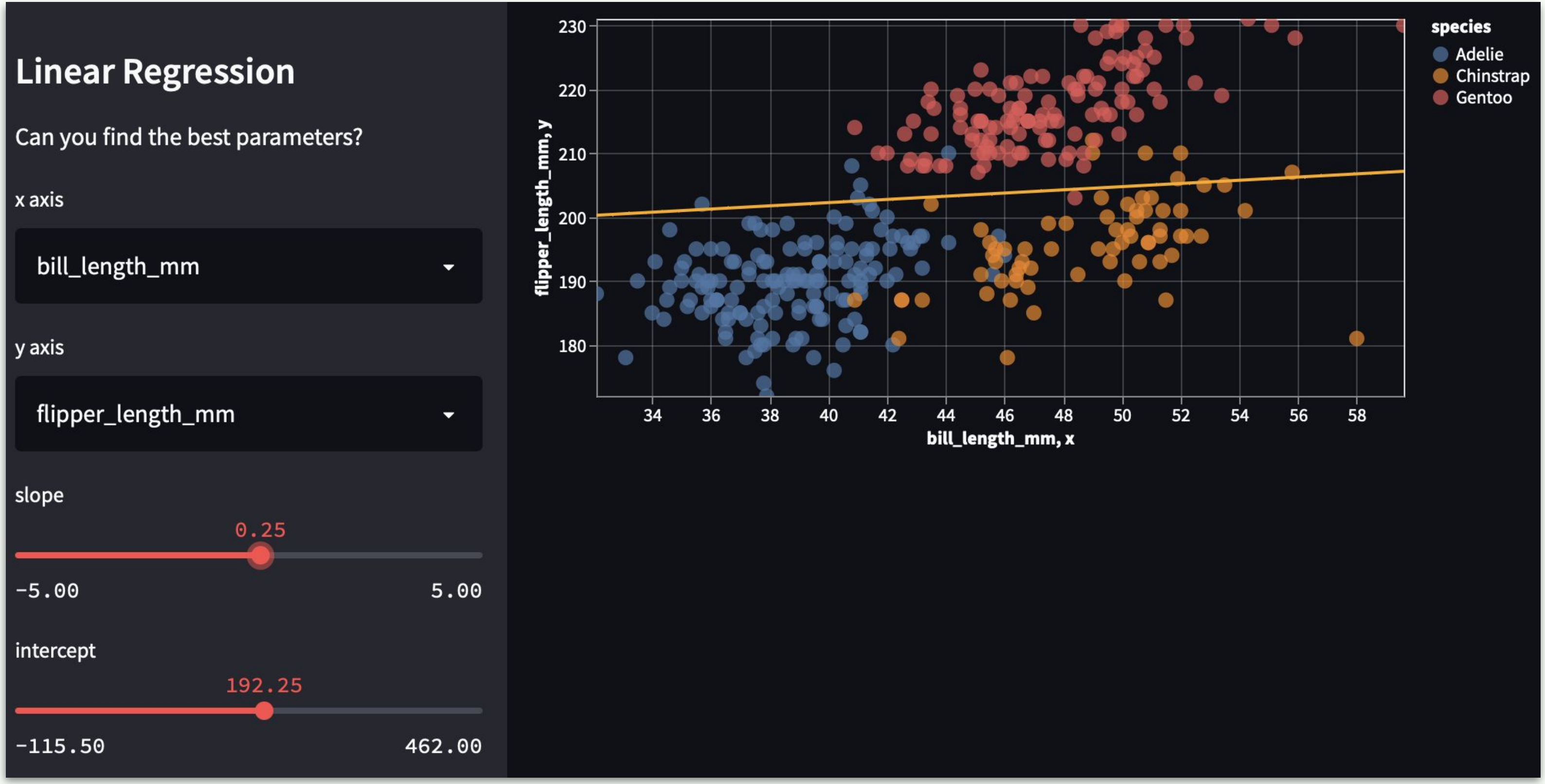
If we are going to code this by hand, we will go one step further: include a non-linearity.

This allows you to see what "linear" really means, and why (true) non-linearity is so much harder.

$$f(x) = \sum_d w_d e^{-(x-x_d)^2/L^2}$$

The "bandwidth" *L* is in the non-linear exponential function.

Write a Web App That Does RBF-NN “By Hand”

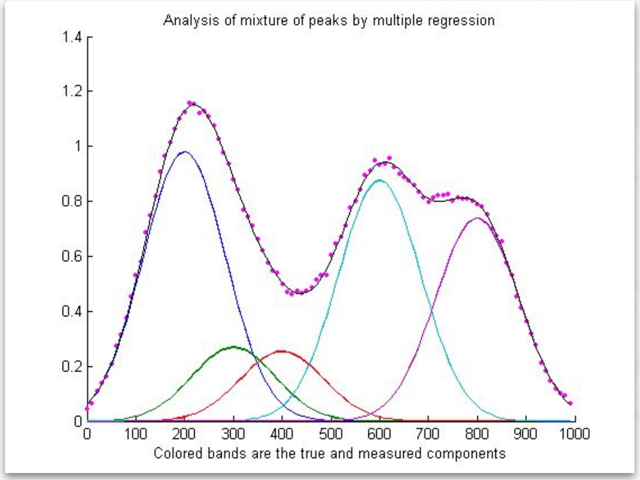


RBF-NN: Two Use Cases

When we use the data points as centers, there are *N* equations in *N* unknowns. The curve will go through every data point.

$$f(x_1) = w_1 e^{-(x_1 - x_1)^2} + w_2 e^{-(x_1 - x_2)^2},$$

$$f(x_2) = w_1 e^{-(x_2 - x_1)^2} + w_2 e^{-(x_2 - x_2)^2}$$

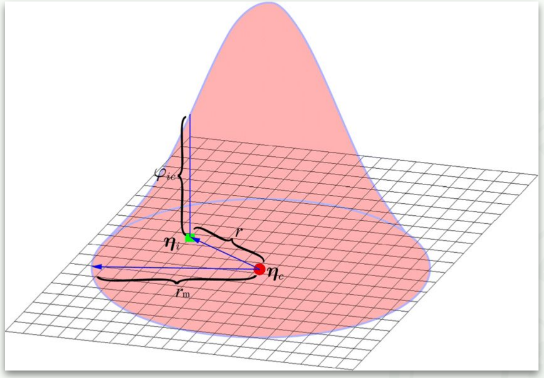


Usually, we have have far fewer parameters than data points.

This causes issues in the algebra.

We'll deal with this in a couple of weeks.

RBF-NN: What Does That Mean?



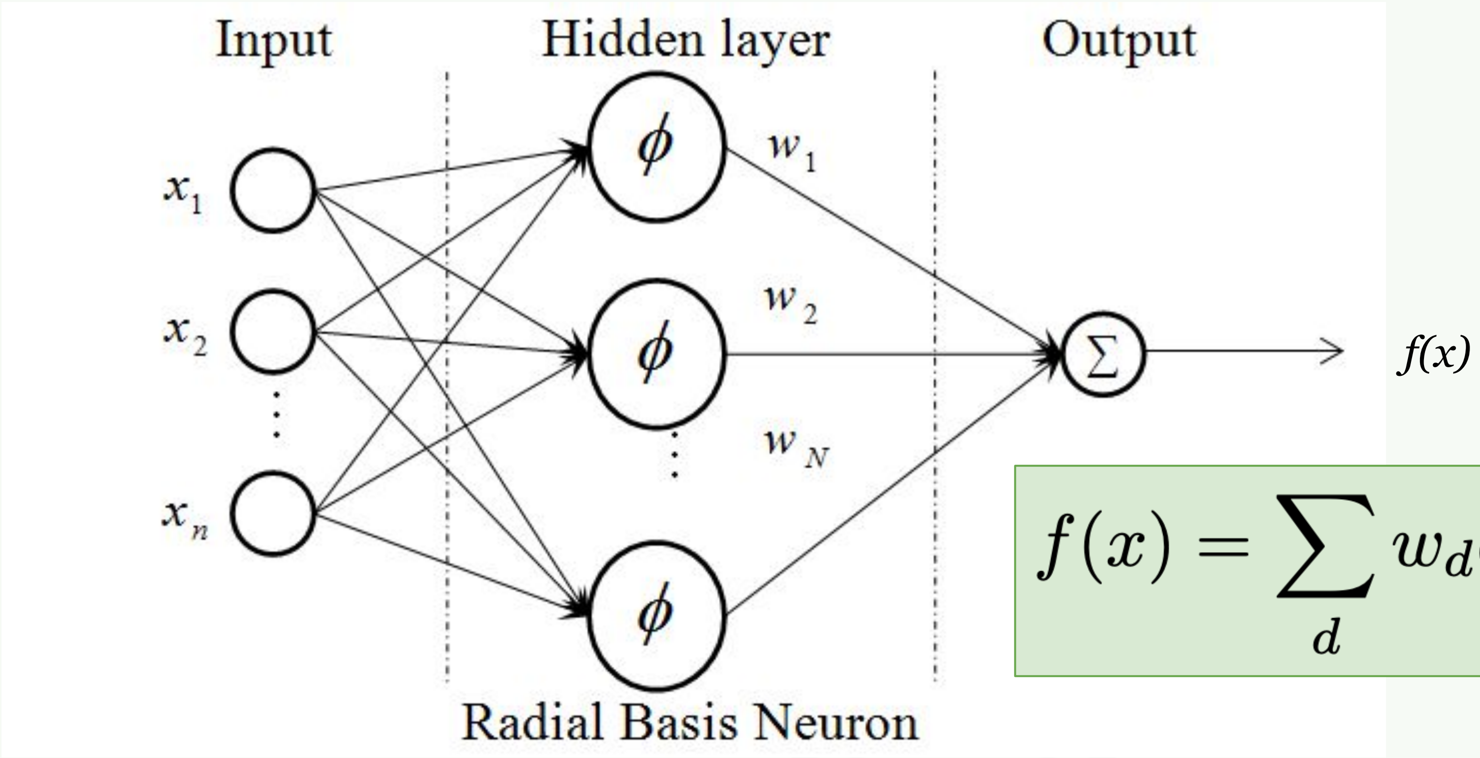
"Radial" functions are functions that vary away from a point the same way in all directions.

Basis expansions write a function in terms of a sum of "basis" functions.

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X),$$

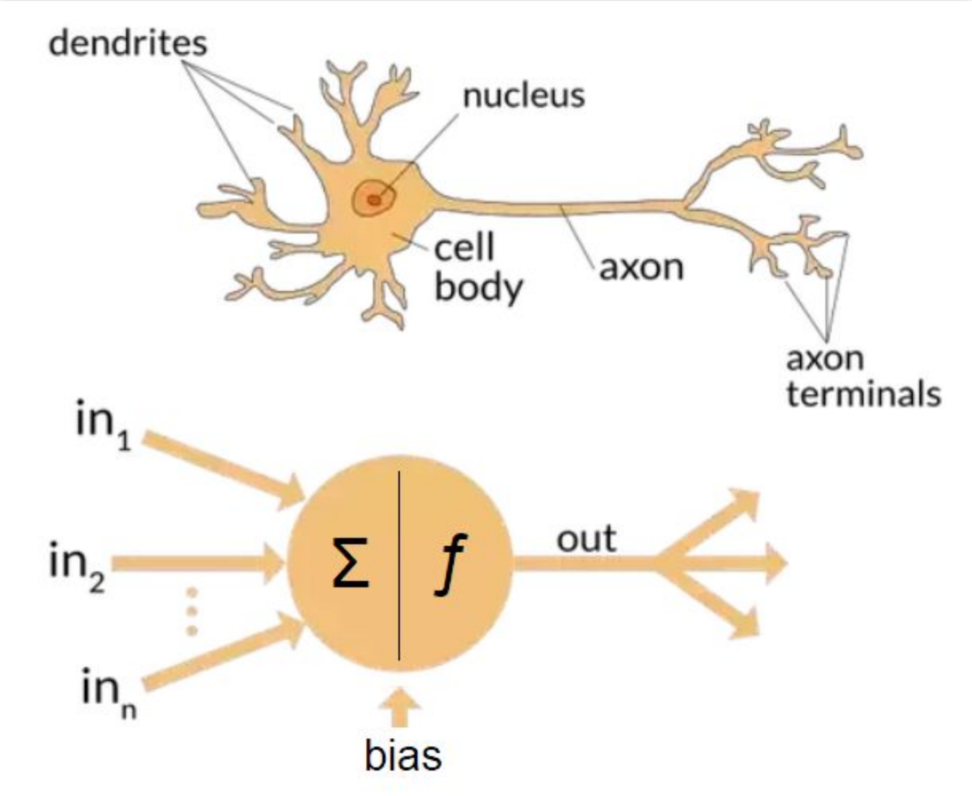
$$f(x) = \sum_d w_d e^{-(x-x_d)^2/L^2}$$

RBF-NN



$$f(x) = \sum_d w_d e^{-(x-x_d)^2/L^2}$$

Neuron Analogy



Plan for ICA

Think through how you would input:

- two centers
- two widths
- two weights

and plot the resulting RBF-NN on the data.

Think about how to do this with good choices for defaults.

Plan for Next Few Weeks

- ➔ this week: linear regression (me!)
- ➔ next week: probability and statistics (David Butts)
- ➔ next next week:
  - ◆ Fall Break!
  - ◆ projects are due!
  - ◆ presentations

Plan for Next Few Weeks

Today's lecture:
Linear Regression

Doc for webapp link is shared, provide link before our class on Wed

Lecture: Probability and Statistics (David Butts)

Project presentation

Presentation reports are due at 11:59 PM

ICA on previous topics

Project material due at 11:59 PM