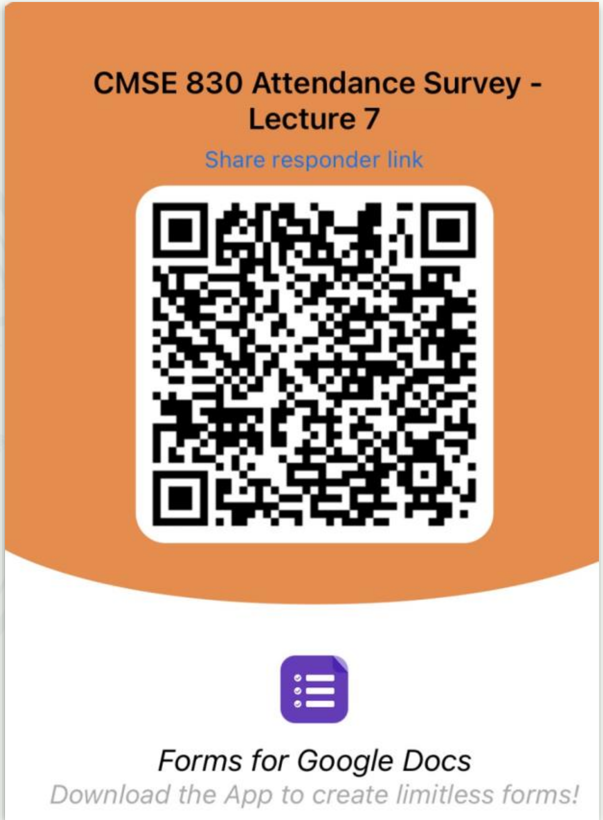
Linear Regression

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



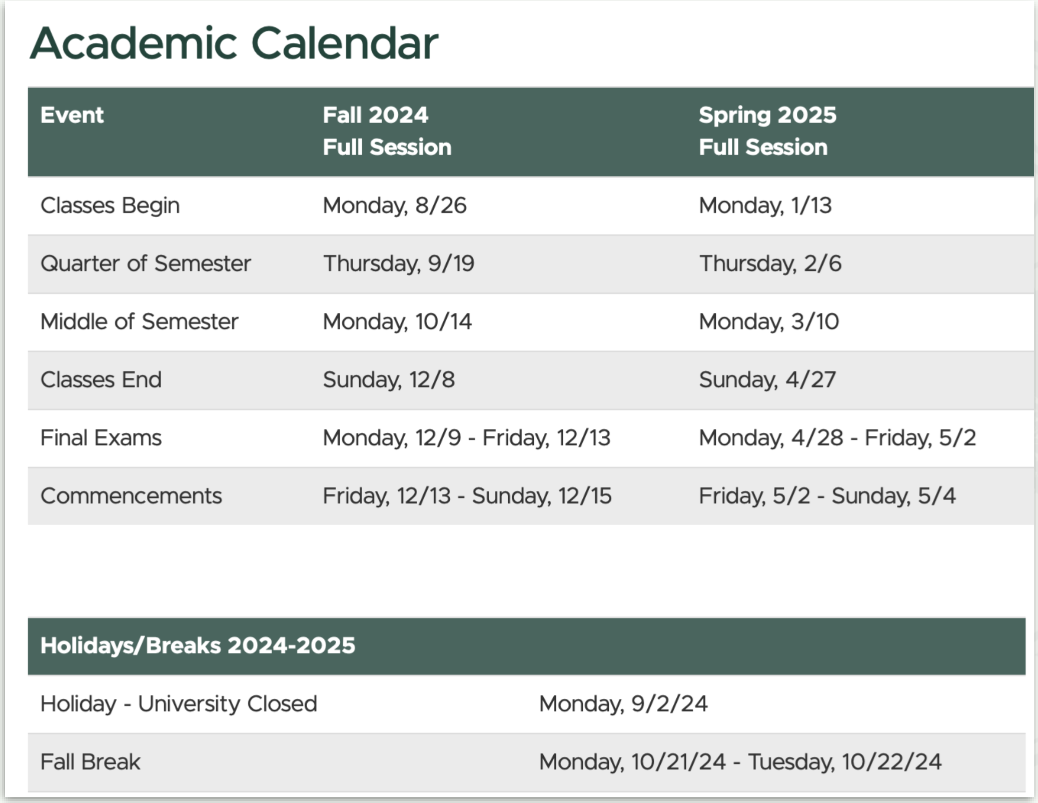
CMSE 830 Attendance Survey - Lecture 7

Share responder link

Forms for Google Docs

Download the App to create limitless forms!

Plan for Next Few Weeks



Academic Calendar

Event	Fall 2024 Full Session	Spring 2025 Full Session
Classes Begin	Monday, 8/26	Monday, 1/13
Quarter of Semester	Thursday, 9/19	Thursday, 2/6
Middle of Semester	Monday, 10/14	Monday, 3/10
Classes End	Sunday, 12/8	Sunday, 4/27
Final Exams	Monday, 12/9 - Friday, 12/13	Monday, 4/28 - Friday, 5/2
Commencements	Friday, 12/13 - Sunday, 12/15	Friday, 5/2 - Sunday, 5/4

Holidays/Breaks 2024-2025	
Holiday - University Closed	Monday, 9/2/24
Fall Break	Monday, 10/21/24 - Tuesday, 10/22/24

Your project will be done before Fall Break so that you can have a real break!

Final details on the midterm project are in this week’s homework.

Main goal: have your project ready to present in class the Thu before Fall Break (17th).

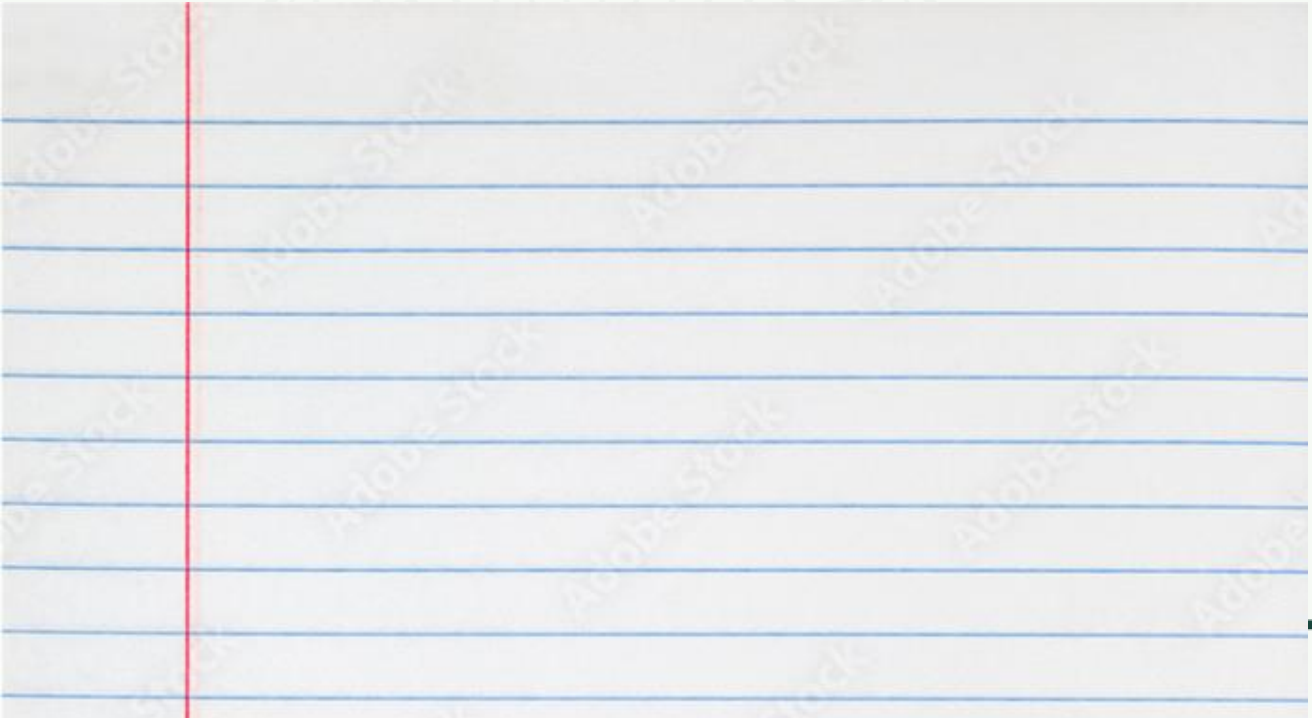
What is Linear Regression?

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?*

$$y = mx + b,$$

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

What is Linear Regression?

Quiz! *Which of these is linear regression?*

$$y = mx + b,$$

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

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

What is Linear Regression?

***Quiz!** Which of these is linear regression?*

$$y = mx + b,$$

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

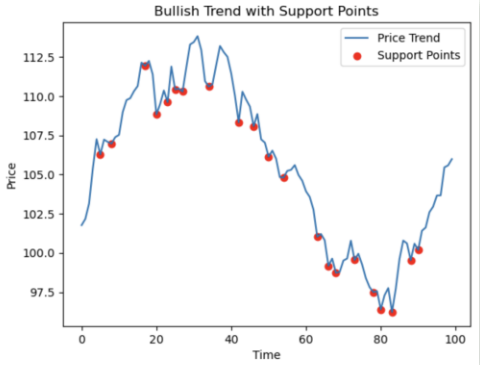
$$f(x) = \sum_d w_d e^{-(x-x_d)^2},$$

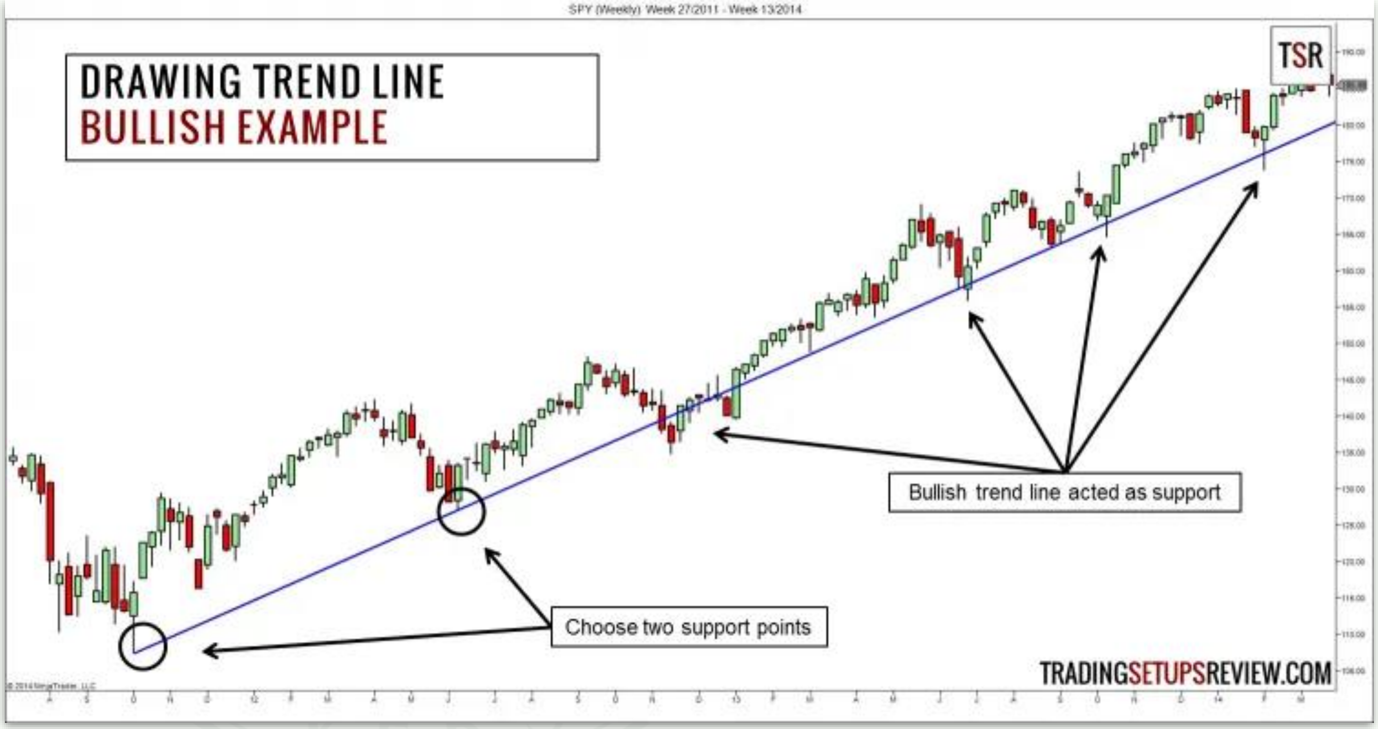
$$p(x_1, x_2) = c\sin(x_1) + d\cos(x_2)$$

Trend Lines, Smoothing and Regression

Trend lines indicate the general direction of a dataset.

You can also use support points (where a trend reverses direction) to define bullish and bearish trends.





DRAWING TREND LINE
BULLISH EXAMPLE

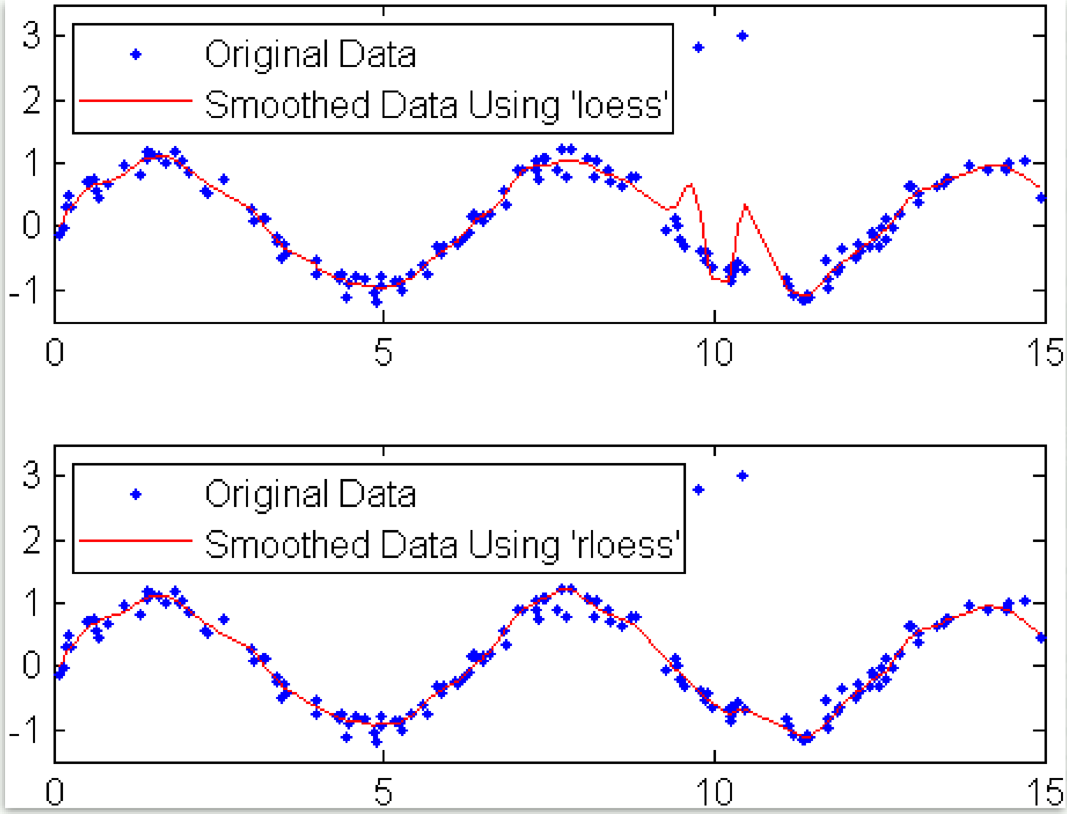
Bullish trend line acted as support

Choose two support points

TRADINGSETUPSREVIEW.COM

Trend lines are mainly qualitative.

Trend Lines, Smoothing and Regression



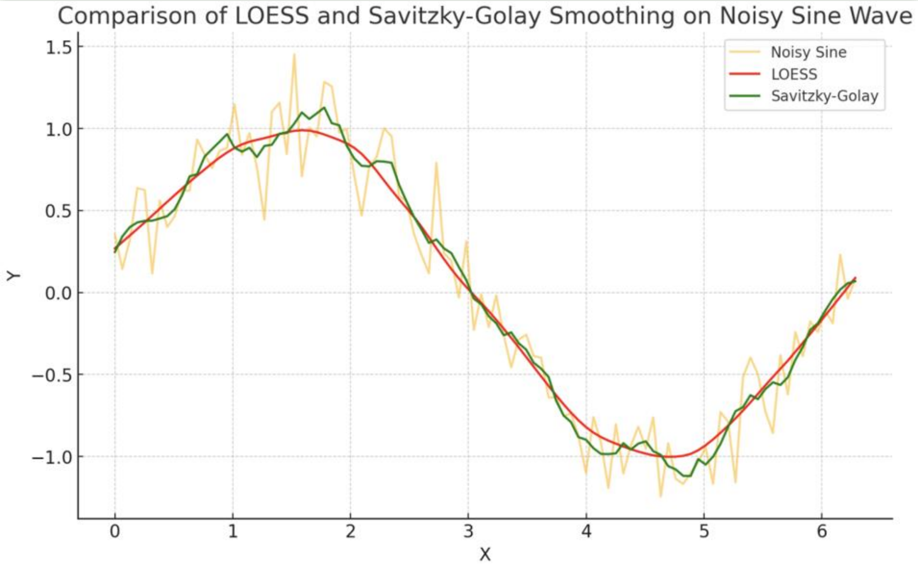
Often we wish to remove noise; that is, we want to *smooth* the data.

In this case, the smoothed data reveals the trend itself.

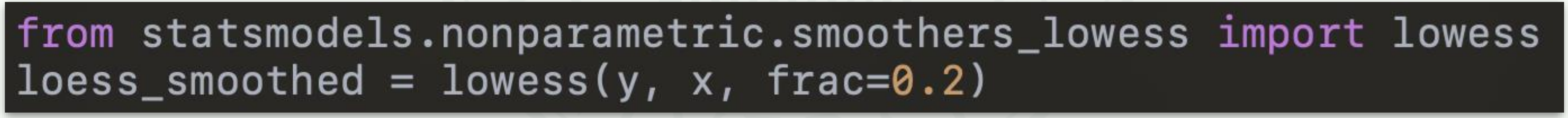
There are endless tools for smoothing (e.g., moving average).

A popular choice is LOESS = Locally Estimated Scatterplot Smoothing.

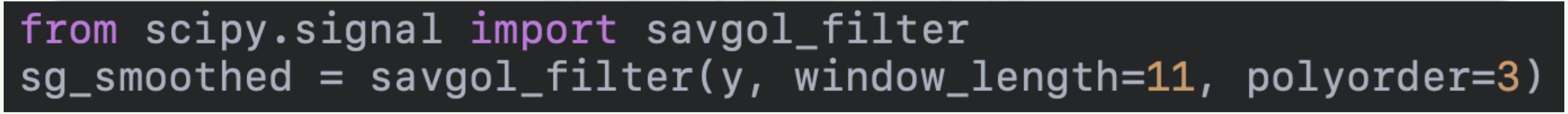
LOESS versus Savitsky-Golay



All of these algorithms have adjustable parameters that allow you to smooth as much as you like.

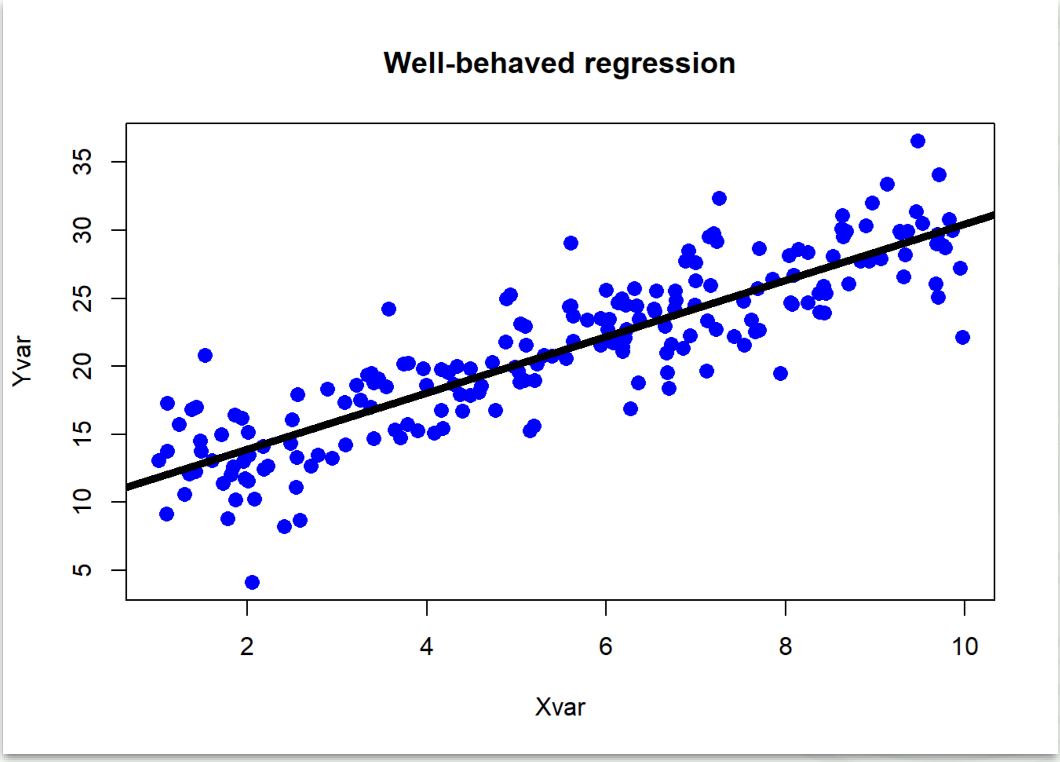


```python
from statsmodels.nonparametric.smoothers_lowess import lowess
loess_smoothed = lowess(y, x, frac=0.2)
```



```python
from scipy.signal import savgol_filter
sg_smoothed = savgol_filter(y, window_length=11, polyorder=3)
```

Trend Lines, Smoothing and Regression

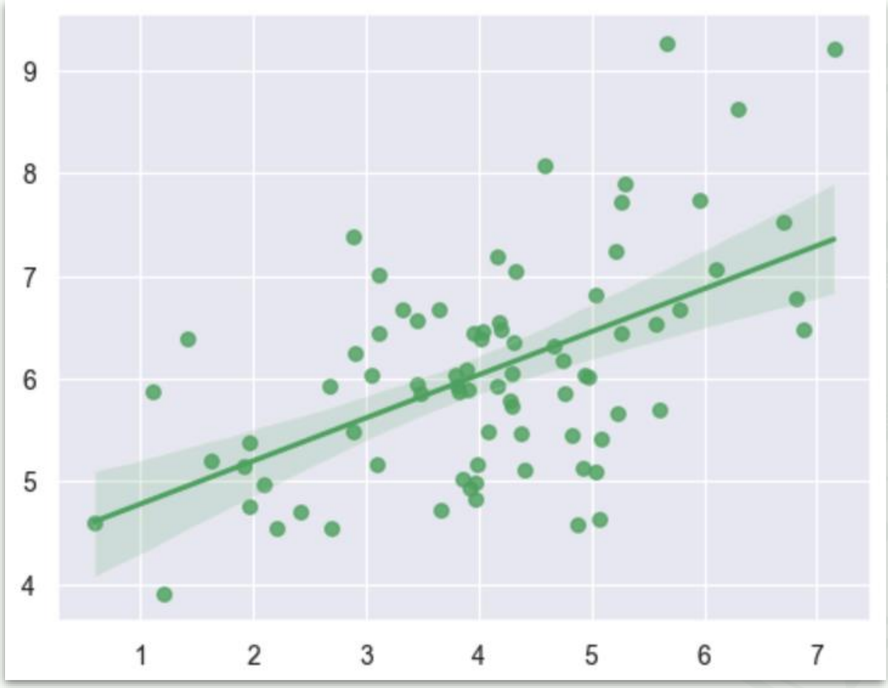


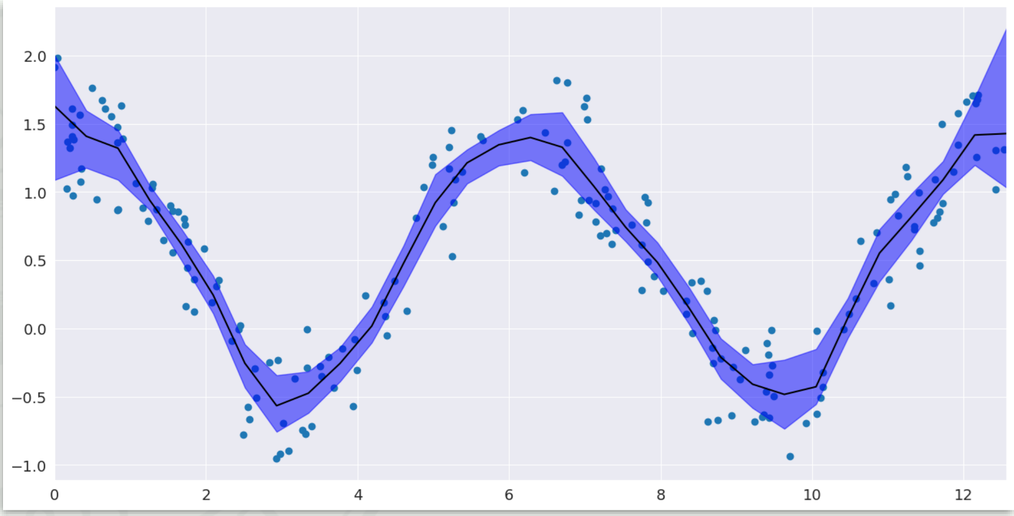
Regression aims to fit a well-defined mathematical model that finds relationships that allow for predictions.

Typically, these are global: all of the data is used to train the model. (LOESS and SavGol are *local*.)

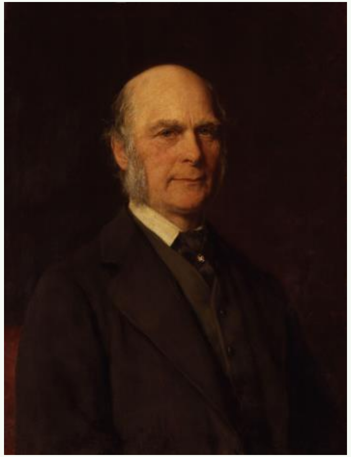
We want to **explain** and **predict**.

Examples: Seaborn and Statsmodels

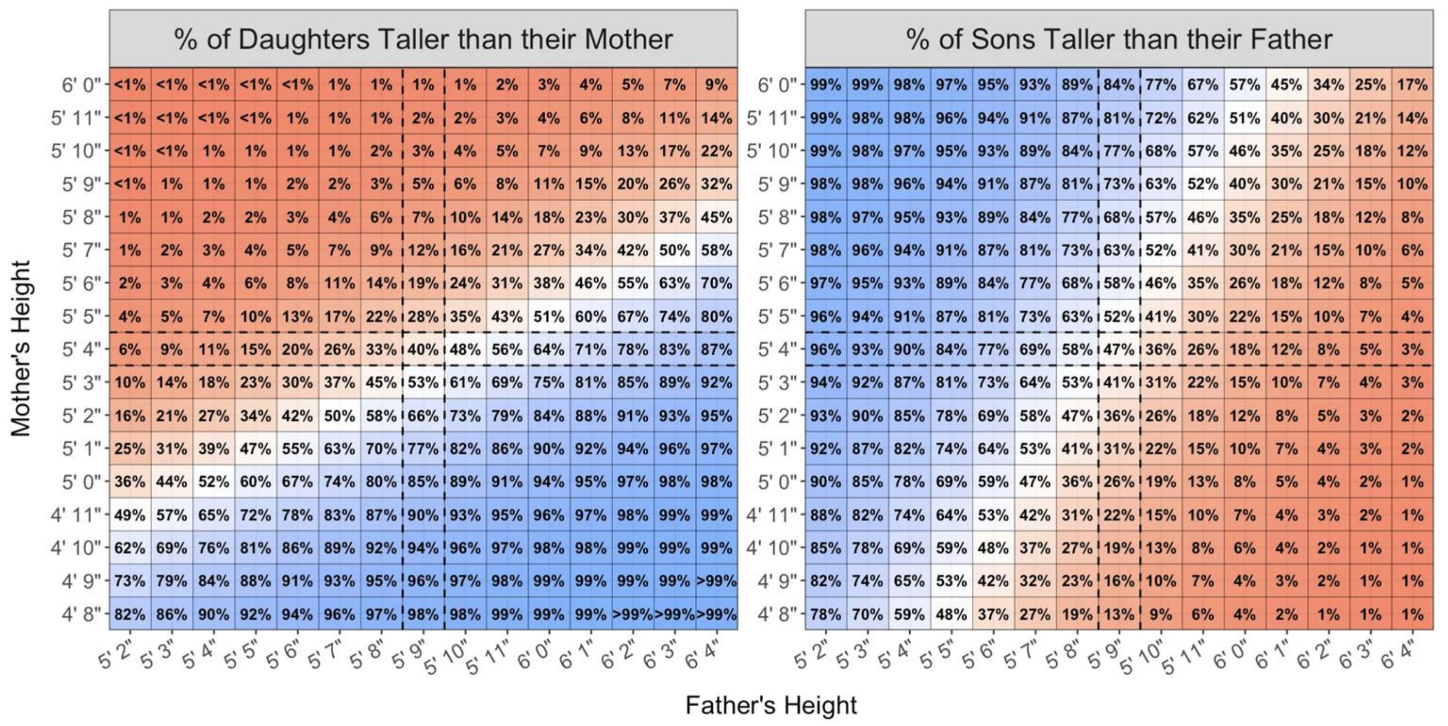




History



Sir Francis Galton introduced the concept of “*regression to the mean*” in the late 1800s.



% of Daughters Taller than their Mother

	5' 2"	5' 3"	5' 4"	5' 5"	5' 6"	5' 7"	5' 8"	5' 9"	5' 10"	5' 11"	6' 0"	6' 1"	6' 2"	6' 3"	6' 4"
6' 0"	<1%	<1%	<1%	<1%	<1%	1%	1%	1%	1%	2%	3%	4%	5%	7%	9%
5' 11"	<1%	<1%	<1%	<1%	1%	1%	1%	2%	2%	3%	4%	6%	8%	11%	14%
5' 10"	<1%	<1%	1%	1%	1%	1%	2%	3%	4%	5%	7%	9%	13%	17%	22%
5' 9"	<1%	1%	1%	1%	2%	2%	3%	5%	6%	8%	11%	15%	20%	26%	32%
5' 8"	1%	1%	2%	2%	3%	4%	6%	7%	10%	14%	18%	23%	30%	37%	45%
5' 7"	1%	2%	3%	4%	5%	7%	9%	12%	16%	21%	27%	34%	42%	50%	58%
5' 6"	2%	3%	4%	6%	8%	11%	14%	19%	24%	31%	38%	46%	55%	63%	70%
5' 5"	4%	5%	7%	10%	13%	17%	22%	28%	35%	43%	51%	60%	67%	74%	80%
5' 4"	6%	9%	11%	15%	20%	26%	33%	40%	48%	56%	64%	71%	78%	83%	87%
5' 3"	10%	14%	18%	23%	30%	37%	45%	53%	61%	69%	75%	81%	85%	89%	92%
5' 2"	16%	21%	27%	34%	42%	50%	58%	66%	73%	79%	84%	88%	91%	93%	95%
5' 1"	25%	31%	39%	47%	55%	63%	70%	77%	82%	86%	90%	92%	94%	96%	97%
5' 0"	36%	44%	52%	60%	67%	74%	80%	85%	89%	91%	94%	95%	97%	98%	98%
4' 11"	49%	57%	65%	72%	78%	83%	87%	90%	93%	95%	96%	97%	98%	99%	99%
4' 10"	62%	69%	76%	81%	86%	89%	92%	94%	96%	97%	98%	98%	99%	99%	99%
4' 9"	73%	79%	84%	88%	91%	93%	95%	96%	97%	98%	99%	99%	99%	99%	>99%
4' 8"	82%	86%	90%	92%	94%	96%	97%	98%	98%	99%	99%	99%	>99%	>99%	>99%

% of Sons Taller than their Father

	5' 2"	5' 3"	5' 4"	5' 5"	5' 6"	5' 7"	5' 8"	5' 9"	5' 10"	5' 11"	6' 0"	6' 1"	6' 2"	6' 3"	6' 4"
6' 0"	99%	99%	98%	97%	95%	93%	89%	84%	77%	67%	57%	45%	34%	25%	17%
5' 11"	99%	98%	98%	96%	94%	91%	87%	81%	72%	62%	51%	40%	30%	21%	14%
5' 10"	99%	98%	97%	95%	93%	89%	84%	77%	68%	57%	46%	35%	25%	18%	12%
5' 9"	98%	98%	96%	94%	91%	87%	81%	73%	63%	52%	40%	30%	21%	15%	10%
5' 8"	98%	97%	95%	93%	89%	84%	77%	68%	57%	46%	35%	25%	18%	12%	8%
5' 7"	98%	96%	94%	91%	87%	81%	73%	63%	52%	41%	30%	21%	15%	10%	6%
5' 6"	97%	95%	93%	89%	84%	77%	68%	58%	46%	35%	26%	18%	12%	8%	5%
5' 5"	96%	94%	91%	87%	81%	73%	63%	52%	41%	30%	22%	15%	10%	7%	4%
5' 4"	96%	93%	90%	84%	77%	69%	58%	47%	36%	26%	18%	12%	8%	5%	3%
5' 3"	94%	92%	87%	81%	73%	64%	53%	41%	31%	22%	15%	10%	7%	4%	3%
5' 2"	93%	90%	85%	78%	69%	58%	47%	36%	26%	18%	12%	8%	5%	3%	2%
5' 1"	92%	87%	82%	74%	64%	53%	41%	31%	22%	15%	10%	7%	4%	3%	2%
5' 0"	90%	85%	78%	69%	59%	47%	36%	26%	19%	13%	8%	5%	4%	2%	1%
4' 11"	88%	82%	74%	64%	53%	42%	31%	22%	15%	10%	7%	4%	3%	2%	1%
4' 10"	85%	78%	69%	59%	48%	37%	27%	19%	13%	8%	6%	4%	2%	1%	1%
4' 9"	82%	74%	65%	53%	42%	32%	23%	16%	10%	7%	4%	3%	2%	1%	1%
4' 8"	78%	70%	59%	48%	37%	27%	19%	13%	9%	6%	4%	2%	1%	1%	1%

Definitions

Simple Linear Regression

$$y = w_0 + w_1 x + \epsilon$$

Ordinary Least Squares (OLS)

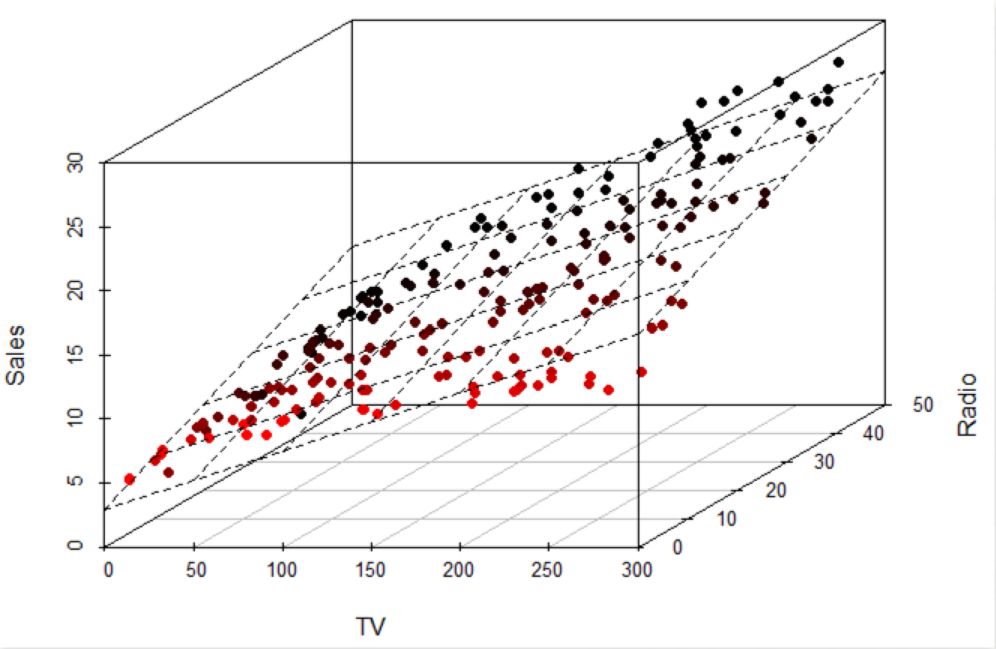
sum of **s**quared **e**rrors

$$SSE = \sum \Big(\overset{\text{observed}}{y_i} - (w_0 + \overset{\text{predicted}}{w_1 x_i})\Big)^2$$

Multiple Linear Regression

$$y = w_0 + w_1 x + w_2 x_2 + \ldots + \epsilon$$

Single Regression Generalizes to Multiple Regression



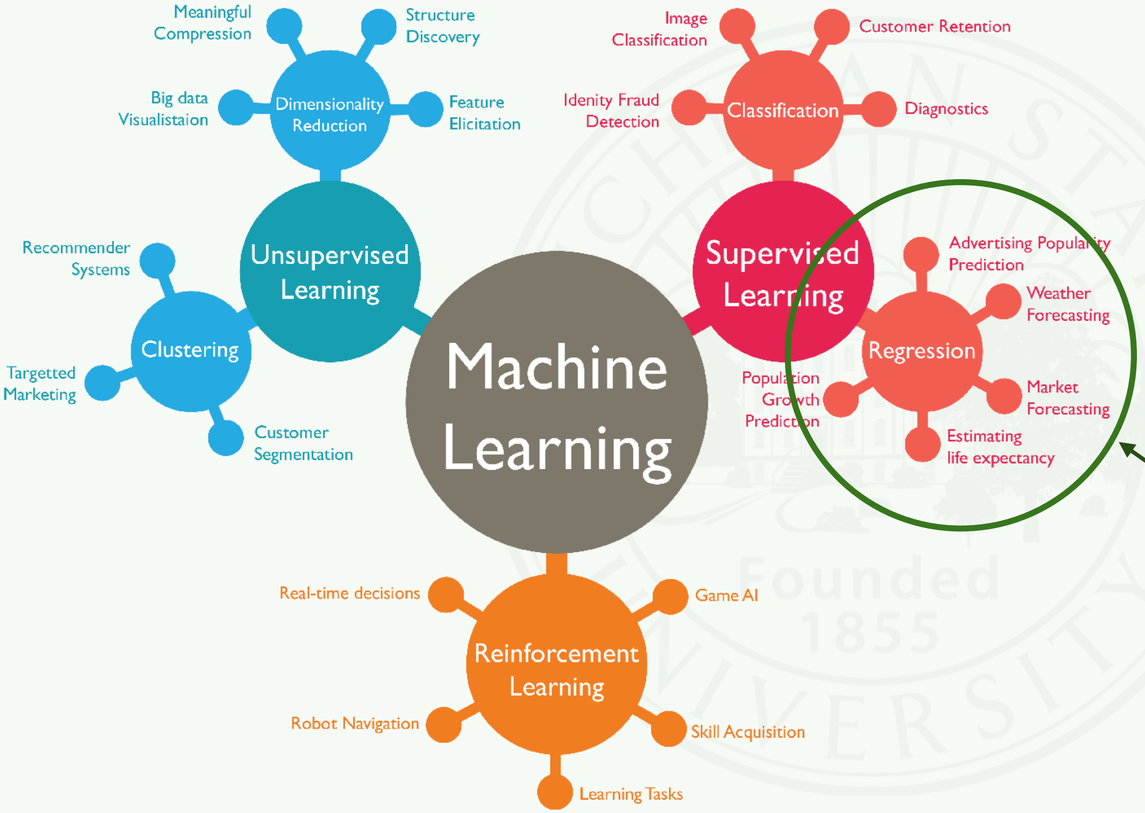
In general, you will **not** be able to see the regression.

Build good habits in 2D and 3D.

Dimensionality reduction can be used to visualize very high dimensional data.

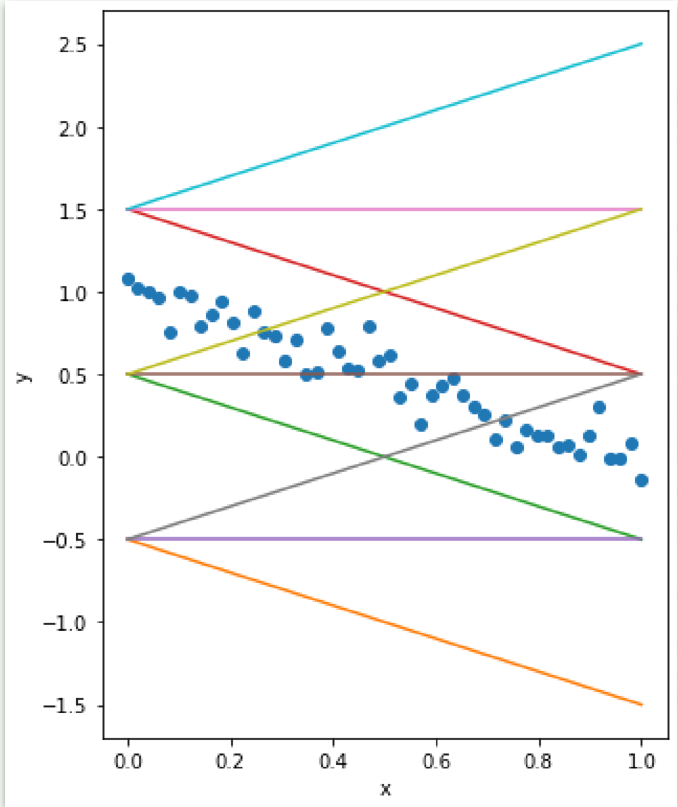
$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

Regression for Prediction: Machine Learning



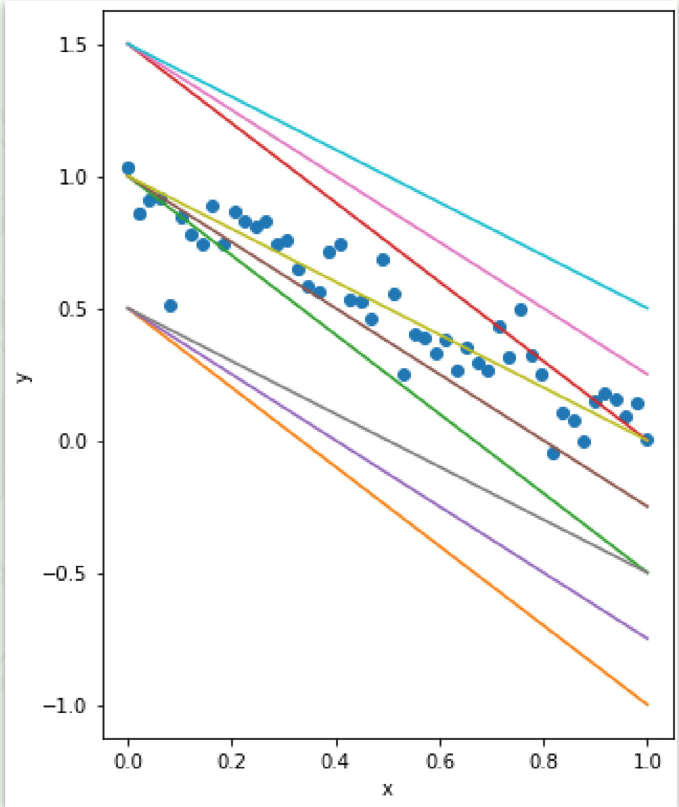
regression branch

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



Which is these lines is “best”?

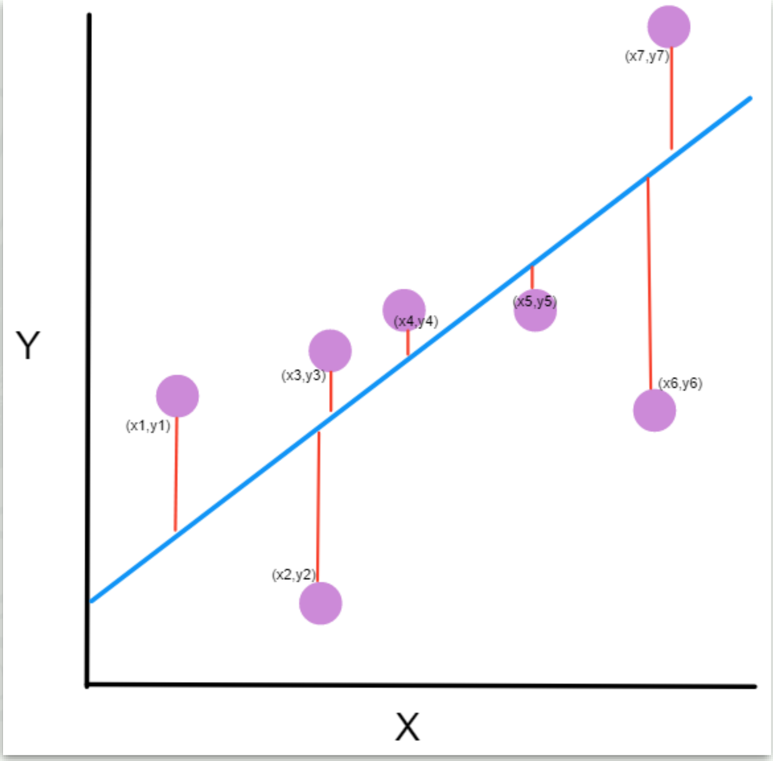
We could iterate until we all agree?



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.



We Can Automate This With a “Grid Search”

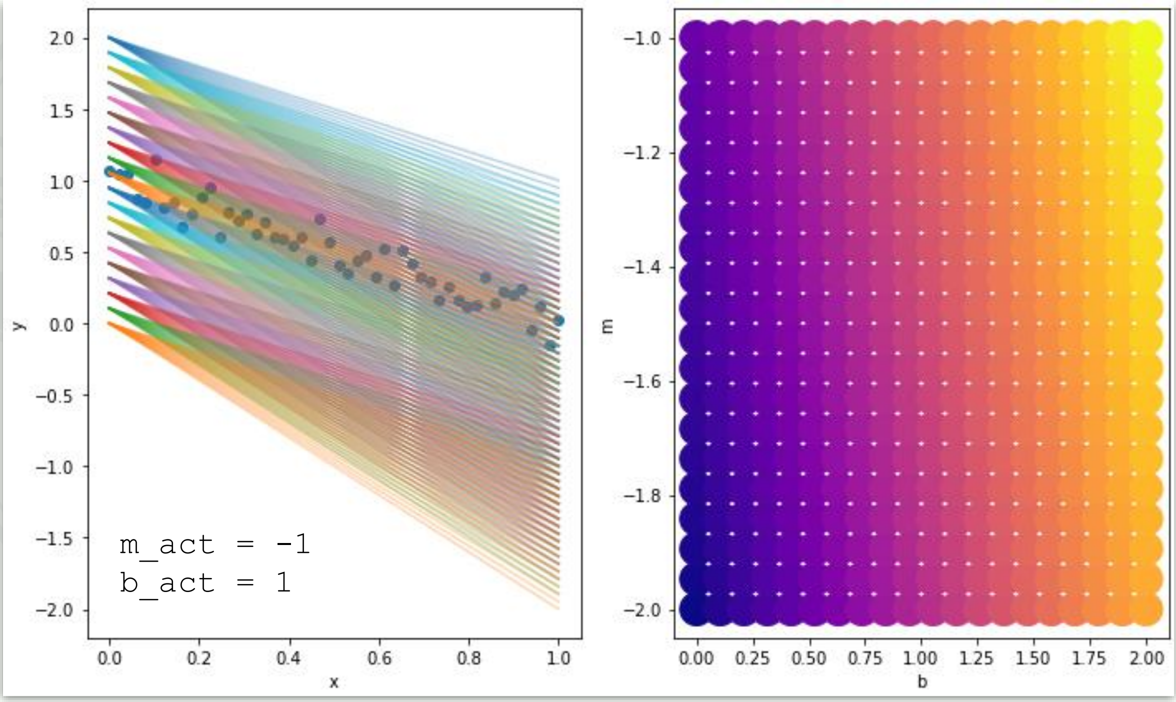
$$\mathcal{L}(m,b) = \sum_d (y_d - [mx_d + b])$$

In general, $\mathcal{L}$ is called a “loss function”.

there is no bottom to the error!

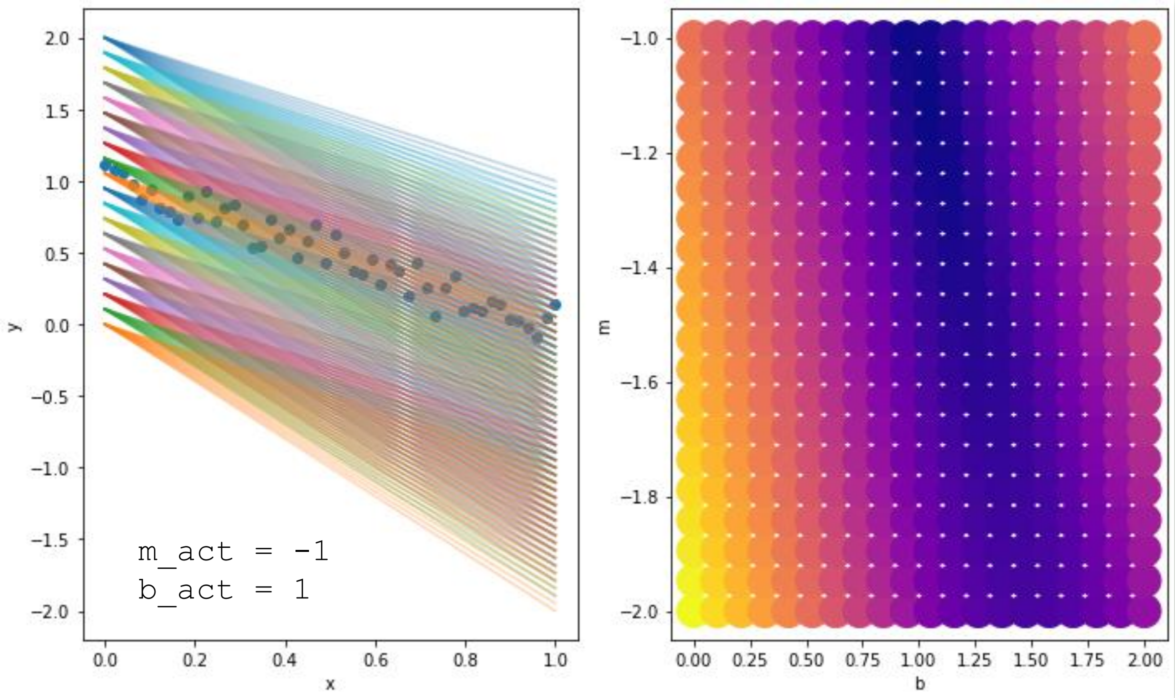
grid search:

- 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

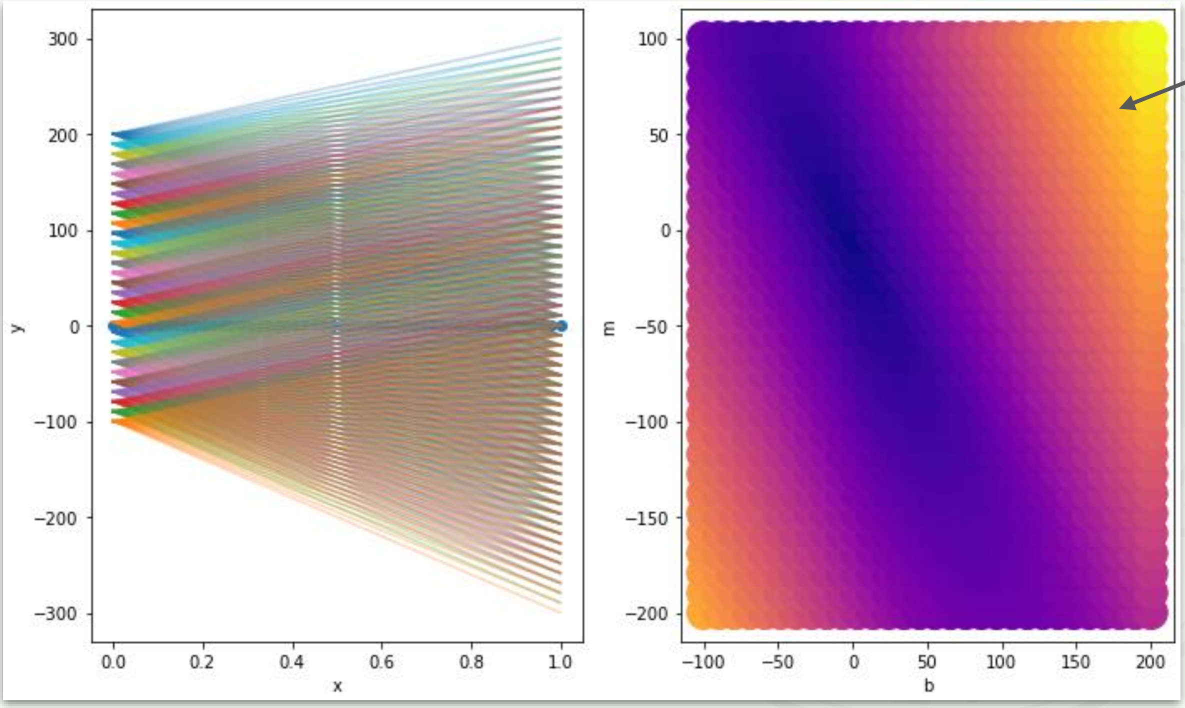


Use MAE (Mean Absolute Error)

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



Use MAE (Mean Absolute Error): Wider Range



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

Finding the best regression line is an optimization problem.

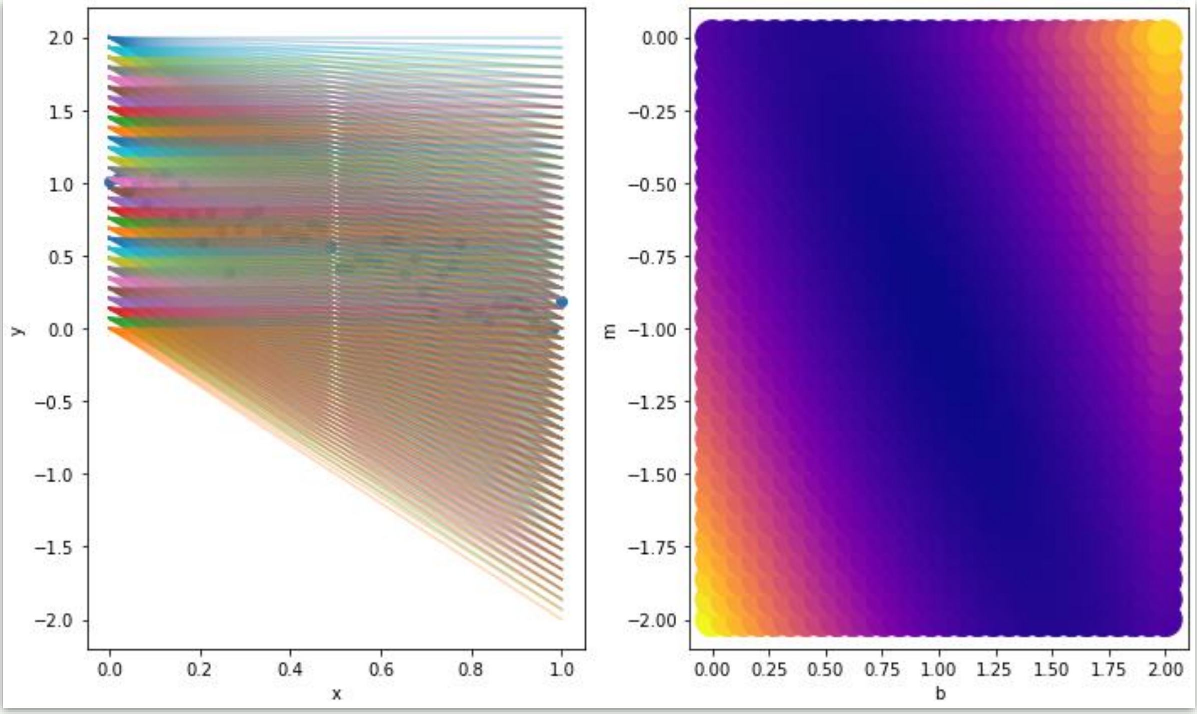
We have defined $L(m,b)$ and we are minimizing it.

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

However, this is tricky for the MAE.

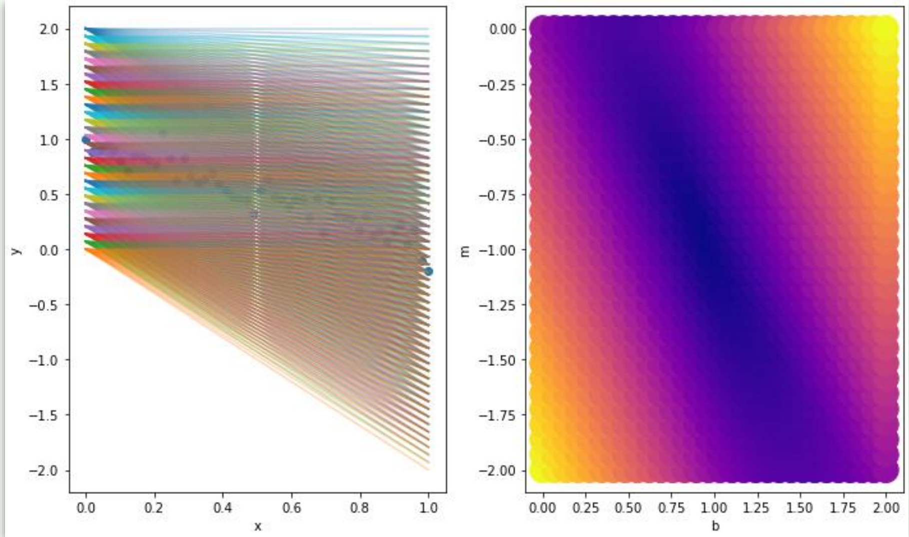
Use MSE (Mean Squared Error)

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

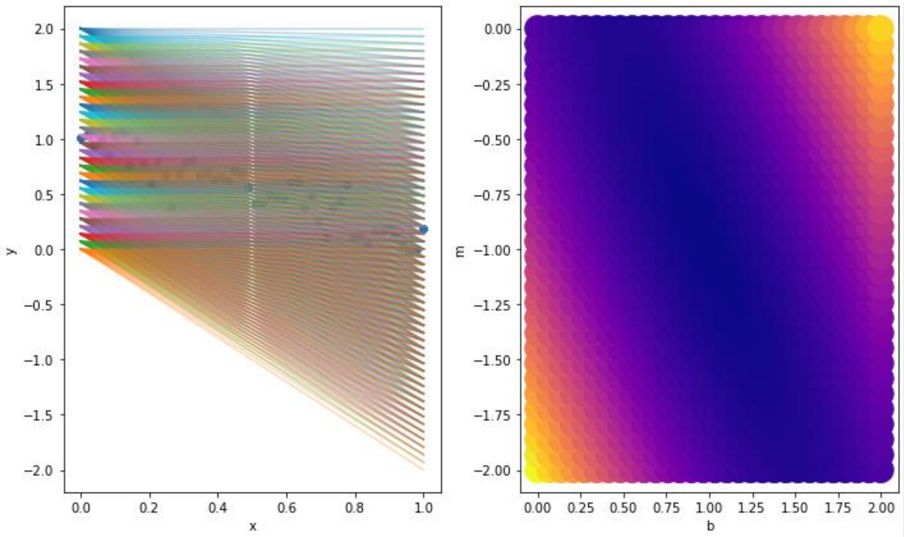


MAE-MSE Comparison

MAE



MSE



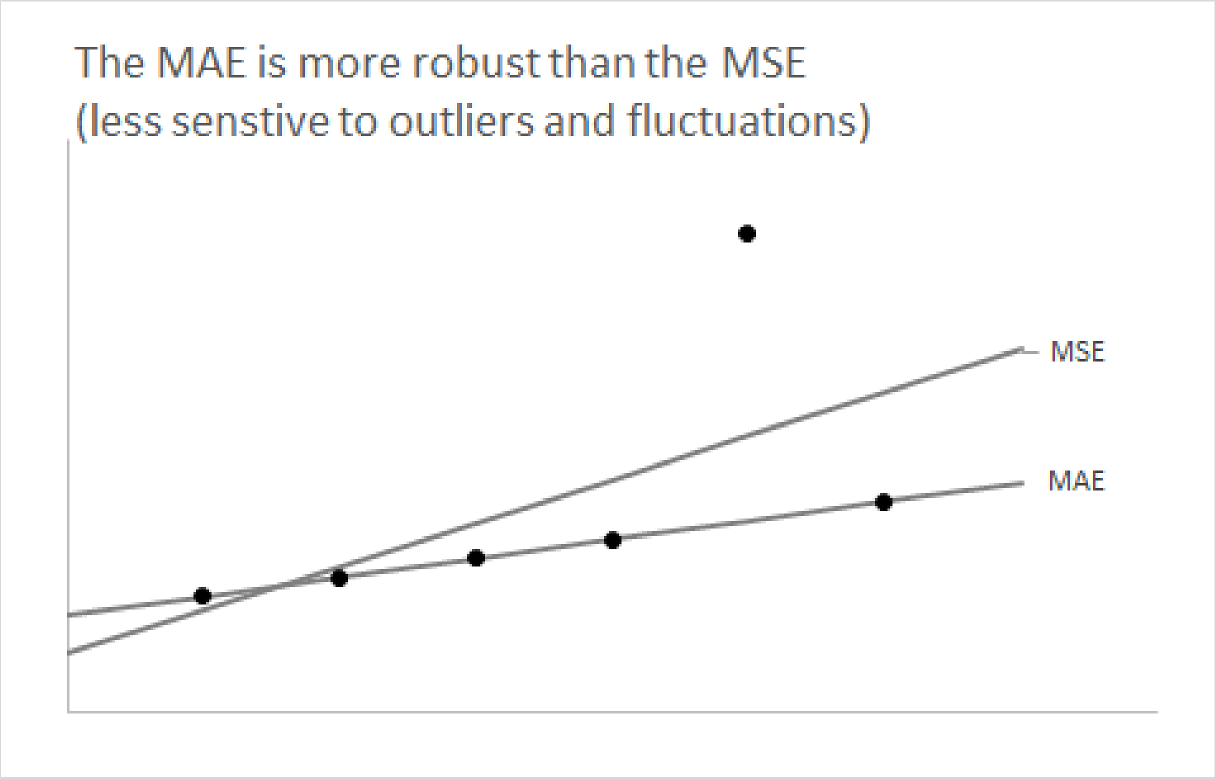
MAE-MSE Comparison

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.



The MAE is more robust than the MSE
(less senstive to outliers and fluctuations)

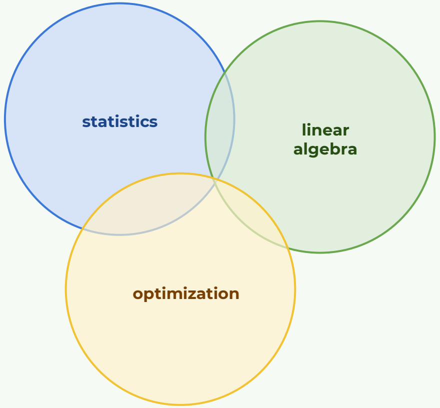
Using **Algebra**, Solve for the MSE Once and For All

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

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

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

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.

Writing the Linear Regression solution in terms of statistics: HW

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

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

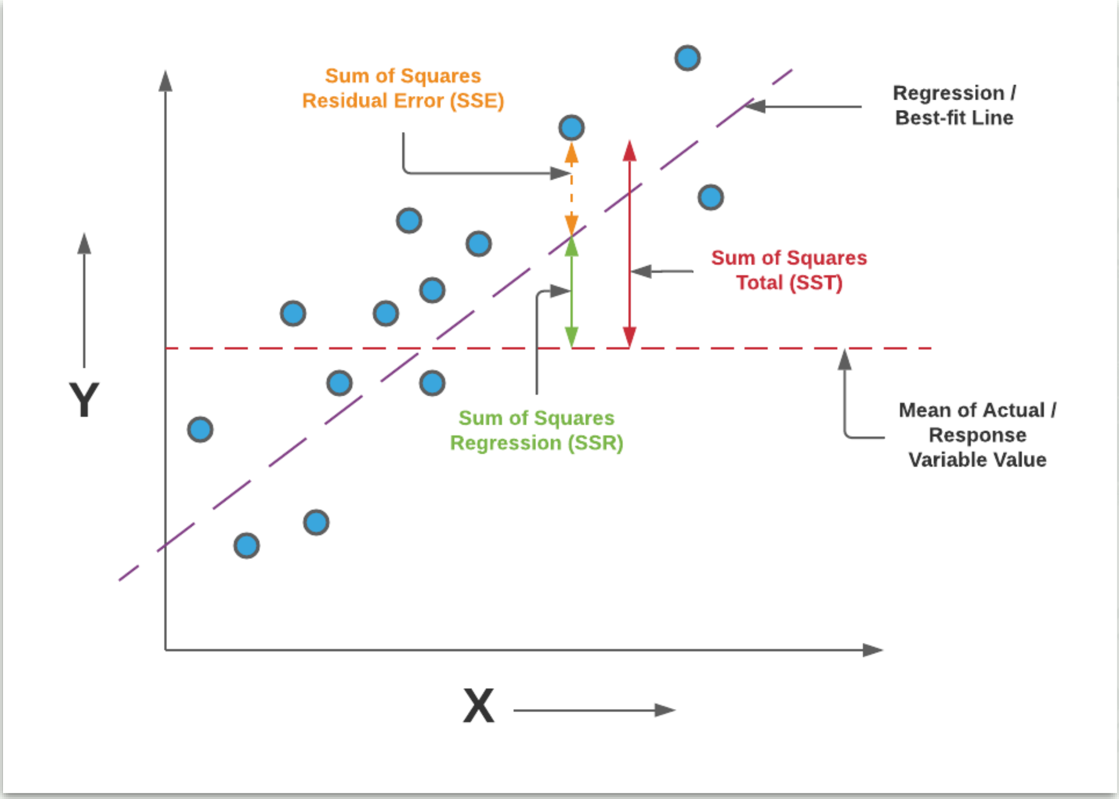
$$= \frac{\mathrm{E}[X,Y] - \mathrm{E}[X]\mathrm{E}[Y]}{\mathrm{E}[X,X] - \mathrm{E}[X]^2},$$

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

You will derive these in the HW this week.

Lucky you!

How well did we do? R^2



- **Total Variability** (SST):

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

- **Explained Variability** (SSR):

$$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

- **Unexplained Variability** (SSE):

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Relationship Between SST, SSR, and SSE:**

$SST = SSR + SSE$

- **R²:**

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

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.

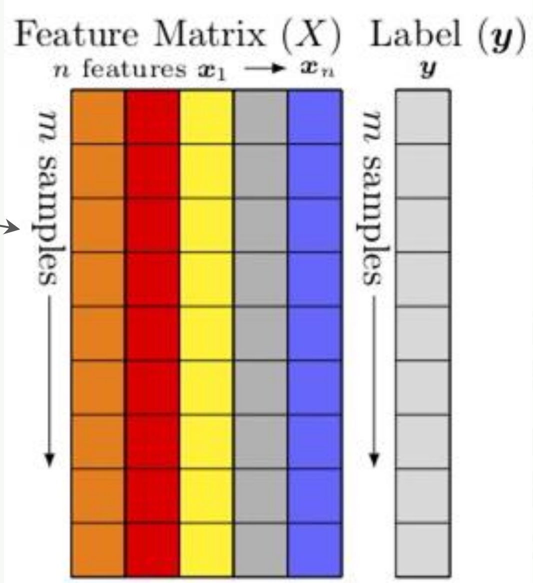
Feature Engineering

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

Vary parameters to minimize the loss. In general, we use multiple linear regression.

$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

This sum is over these rows in the data matrix.



The math doesn’t know or care where the numbers in the columns came from.

We could have, for example:

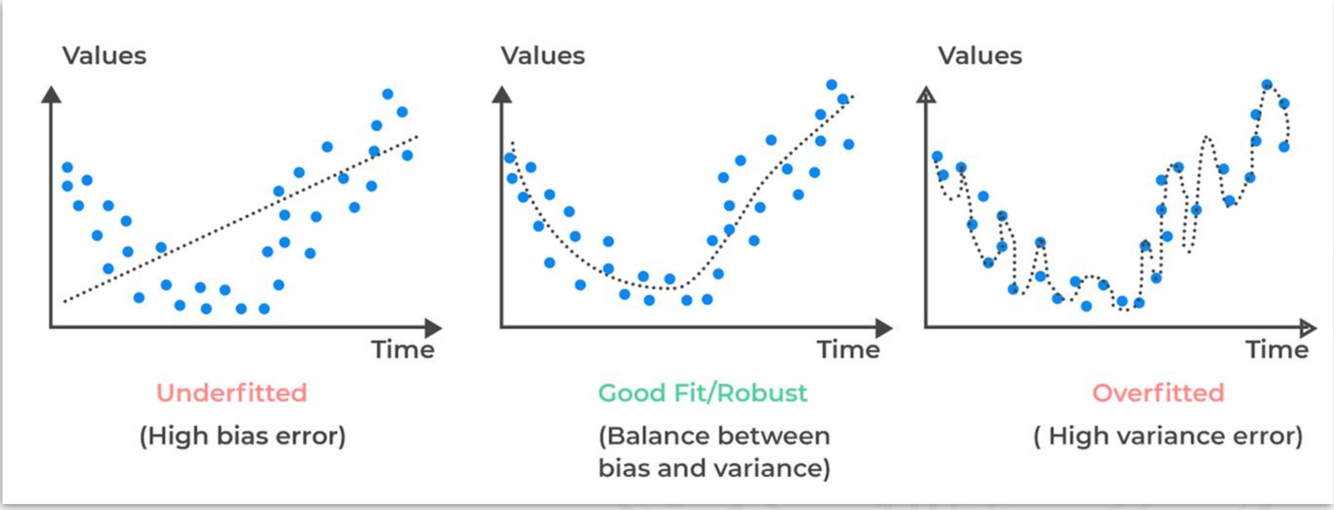
$$x_2 = x_1^2$$

$$x_n = x_1^n$$

There are libraries for this. For example, `sklearn` has `PolynomialFeatures`.

This is why linear regression allows for non-linear-function fits.

Overfitting: Bias-Variance Tradeoff



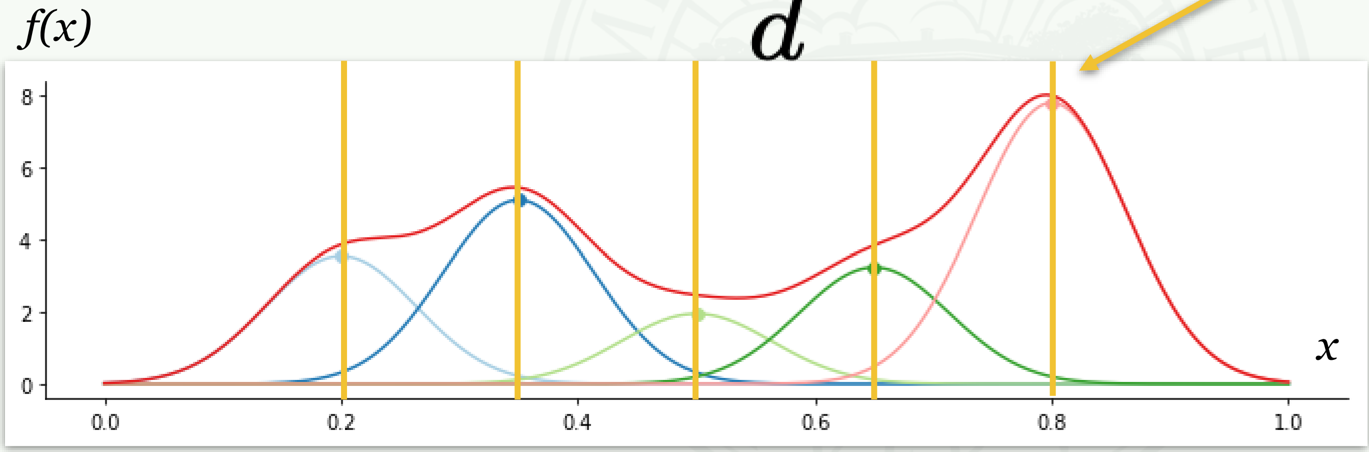
There are methods beyond the scope of this course that prevent overfitting.

For now, use the simplest model that EDA suggests is reasonable.

We’ll explore this a bit in the HW.

Radial Basis Function Neural Networks

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

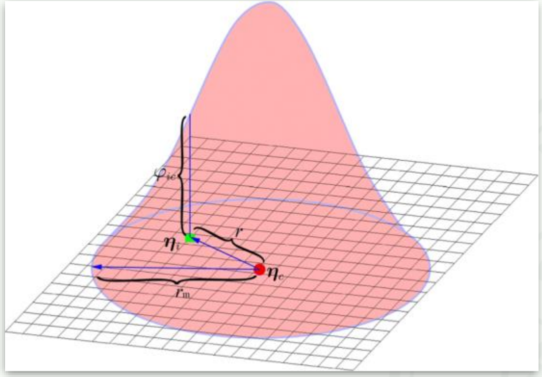


If we naively put each RBF through the data, the model will be a poor fit.

The x_d are literally the positions of the data points. *This makes the problem linear in the coefficients!*

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

RBF-NN: What Does That Mean?



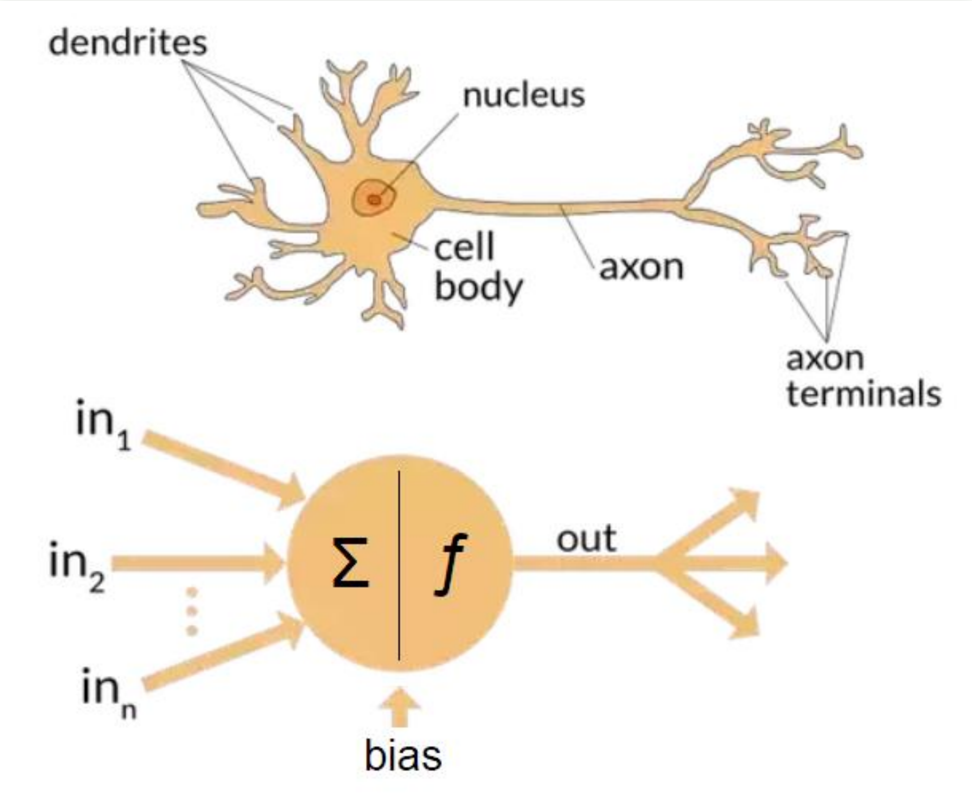
“Radial” functions are functions that vary away from a point the same way in all directions.

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

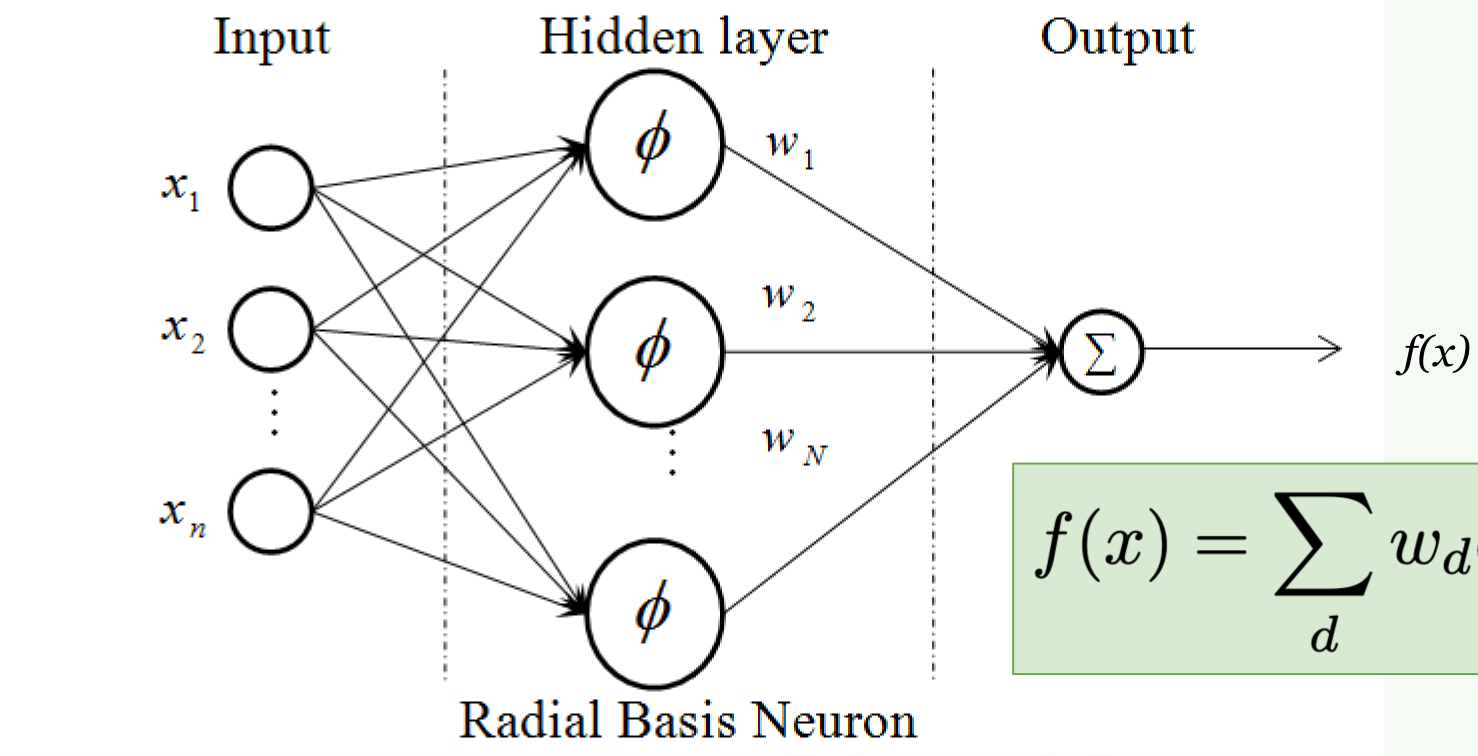
$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X),$$

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

Neuron Analogy



RBF-NN Diagrams

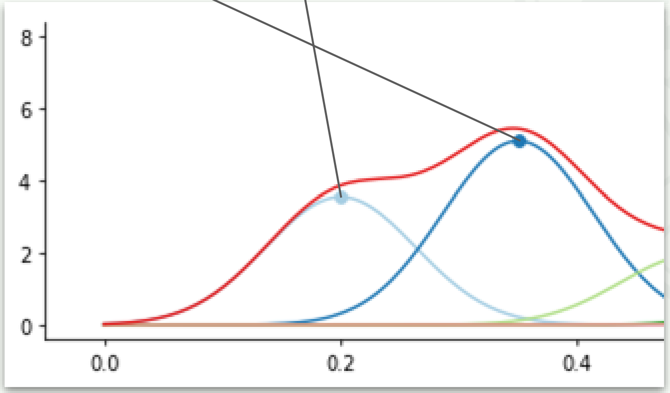


$$f(x) = \sum_d w_d e^{-(x-x_d)^2/L^2}$$

As Written, RBF-NNs is a Linear Regression Problem

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

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

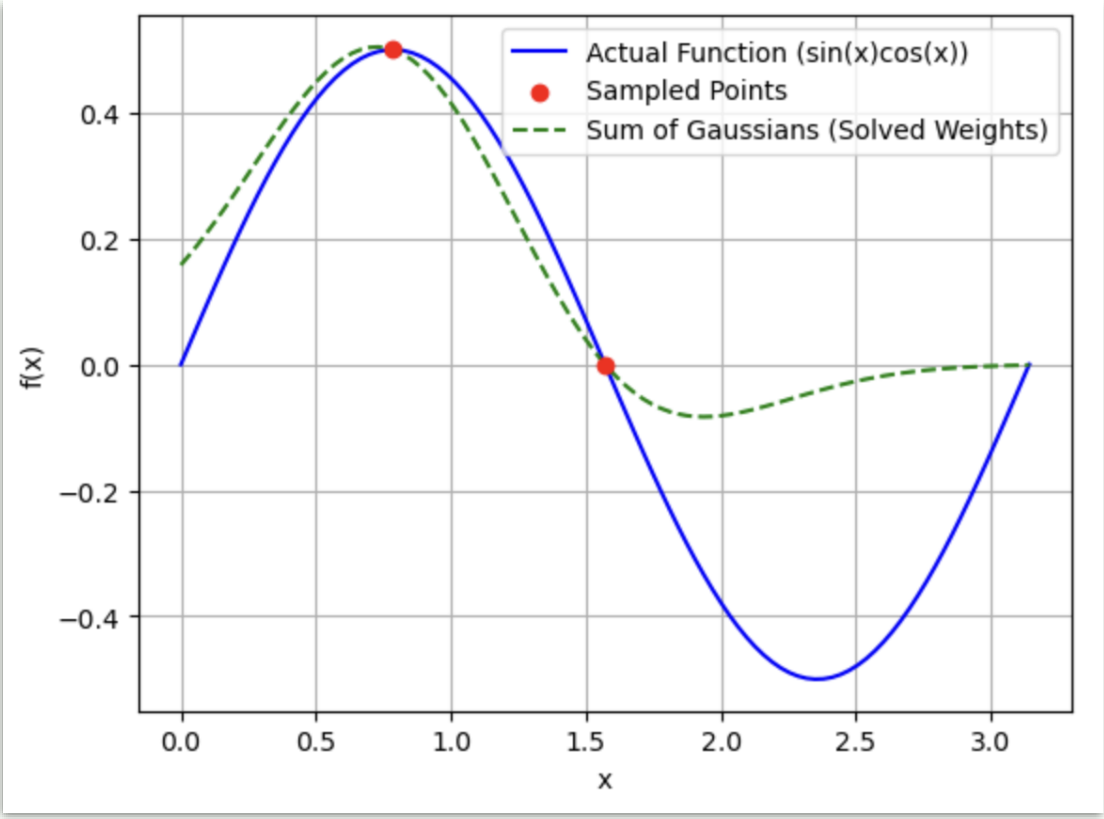


We have two equations in two unknowns.

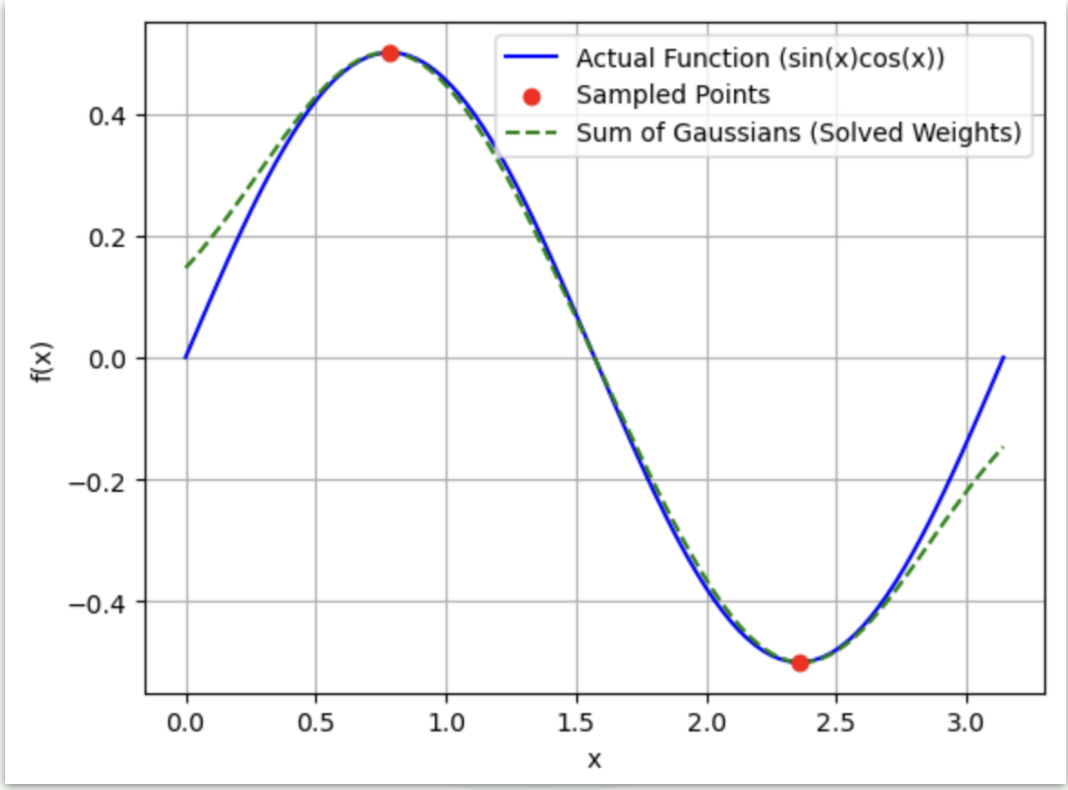
Everything else is known.

Just some simple algebra!

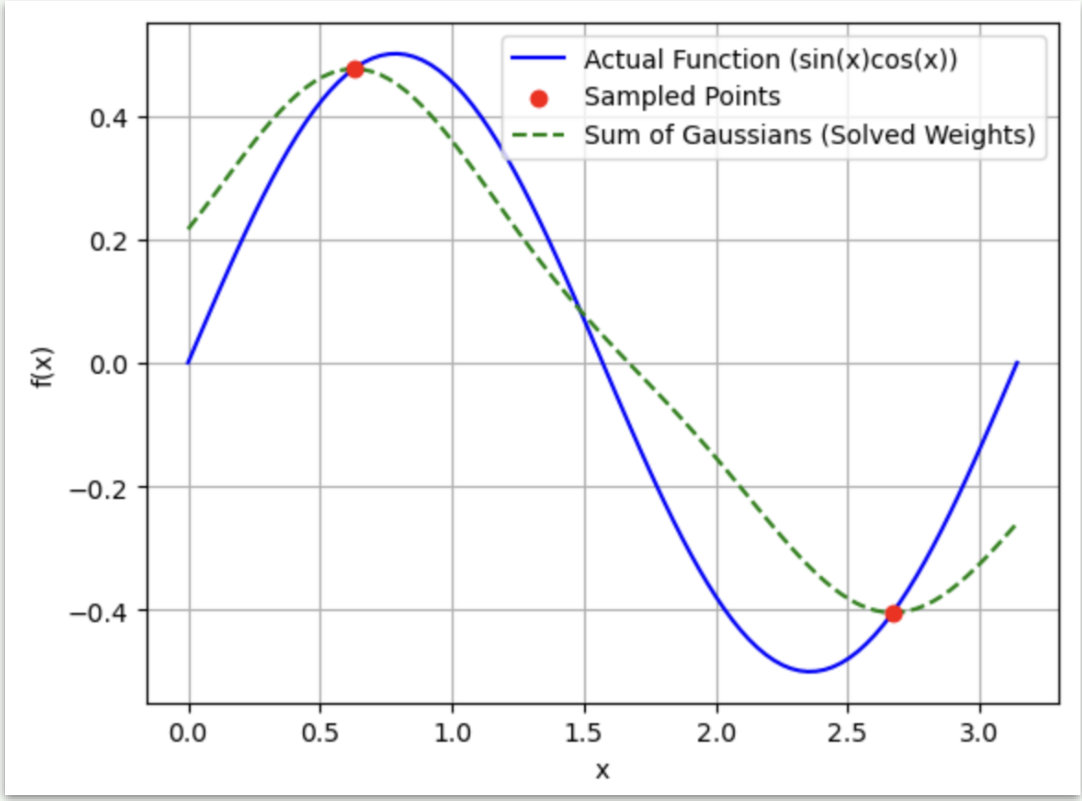
As Written, RBF-NNs is a Linear Regression Problem



As Written, RBF-NNs is a Linear Regression Problem

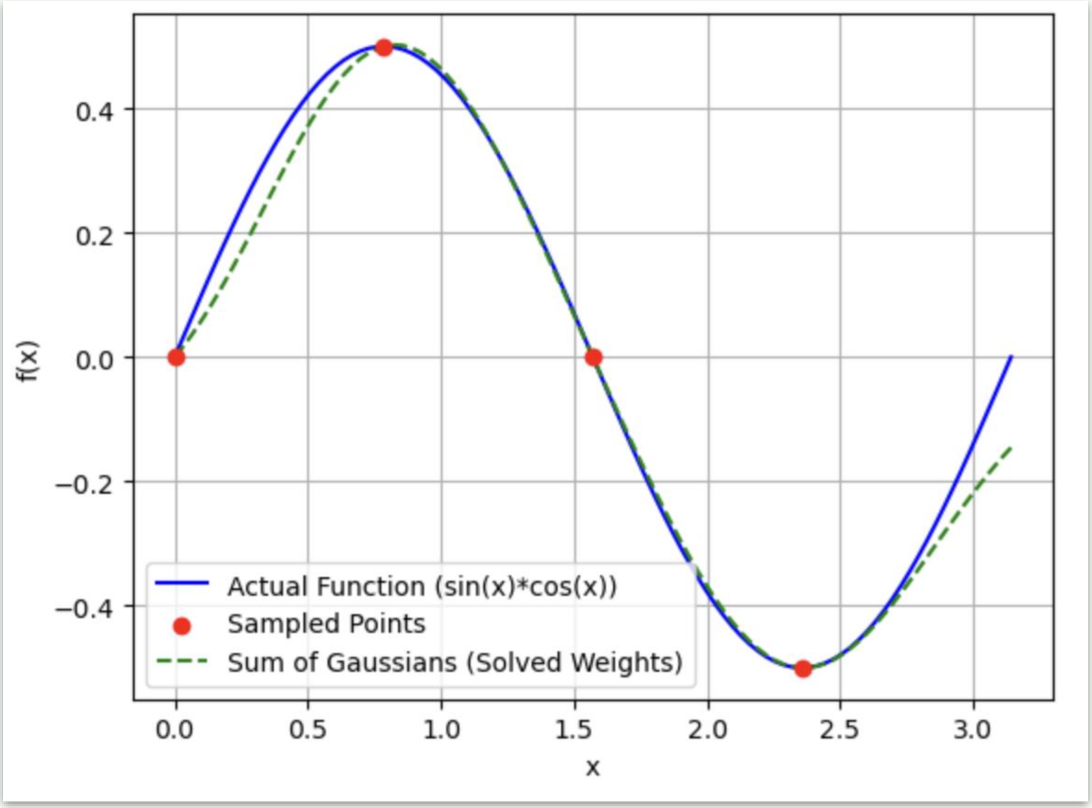


As Written, RBF-NNs is a Linear Regression Problem



More Points is Better!

Here, the algebra is getting less simple.

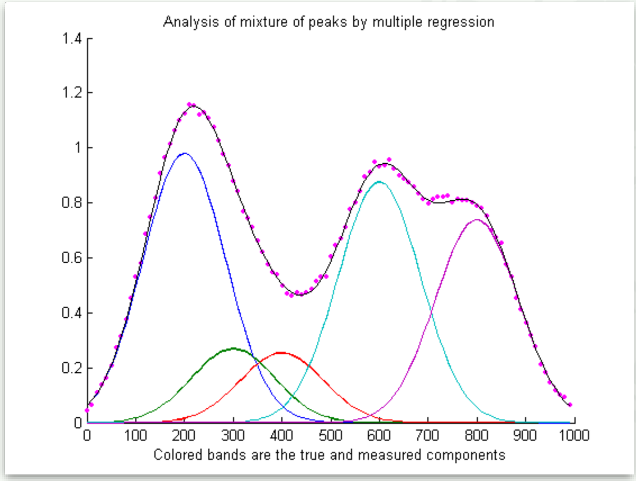


Noise, Overfitting, “Wrong” Number of Parameters

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

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

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

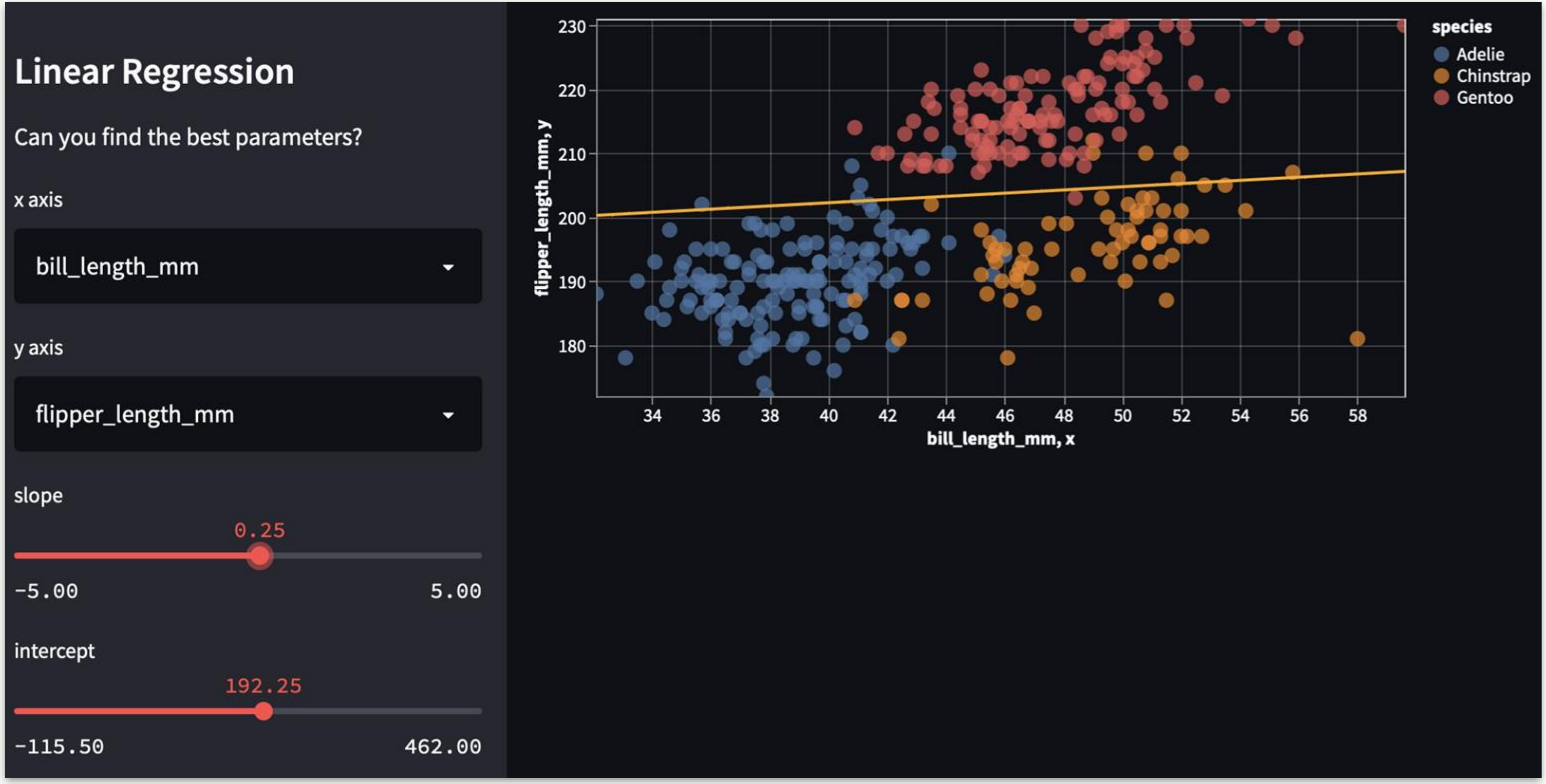


Usually, we have have far fewer parameters than data points.

This causes issues in the algebra.

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

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



Linear Regression

Can you find the best parameters?

x axis: bill_length_mm

y axis: flipper_length_mm

slope: 0.25 (range -5.00 to 5.00)

intercept: 192.25 (range -115.50 to 462.00)