# MATLAB

Class 7: Computational models & model selection

# What is a model?


VOGUE
NOV

# What is a model?

A way to describee phenomena in abstract terms that is useful for us to understand, define, quantify or predict

abstract-- removed from referent

# What is a model?

“the aim of which is to make a particular part or feature of the world easier to understand, define, quantify, visualize, or simulate “ --Wikipedia

# Models in Science

**Descriptive Model**:

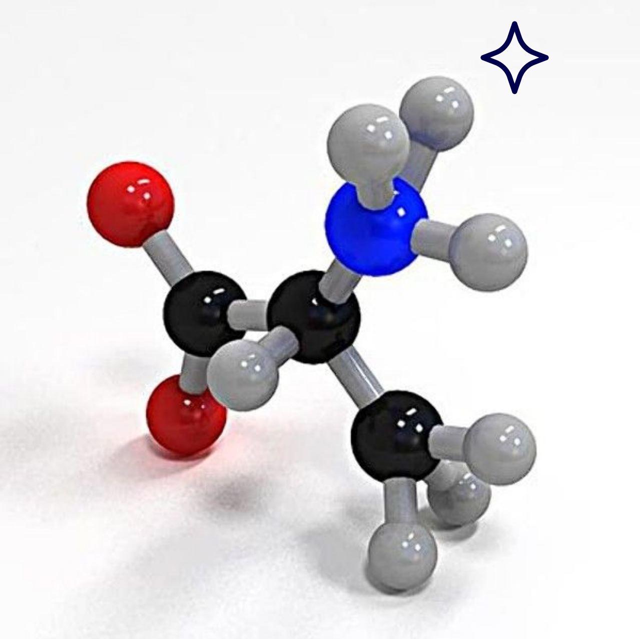
Abstraction that represents reality in nonliteral terms

E.g. Bohr-Rutherford model

**Computational Model**:

A mathematical description of a phenomena that requires computation

Nothing special about computers... you could use toilet paper and stones (Searle, 1980)

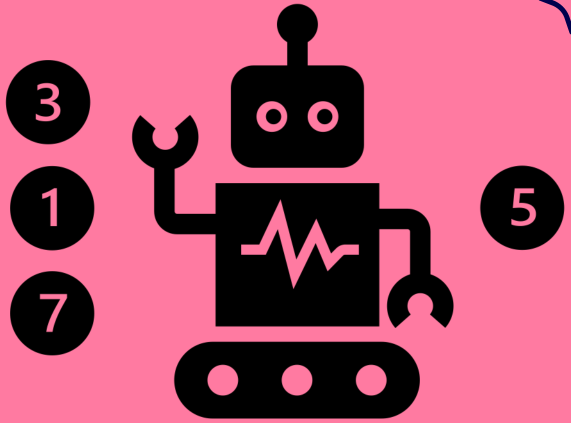


*** not mutually exclusive

# Why use computational models

If you wanted to build a machine, you would need to know how that machine takes its inputs and processes information to produce output

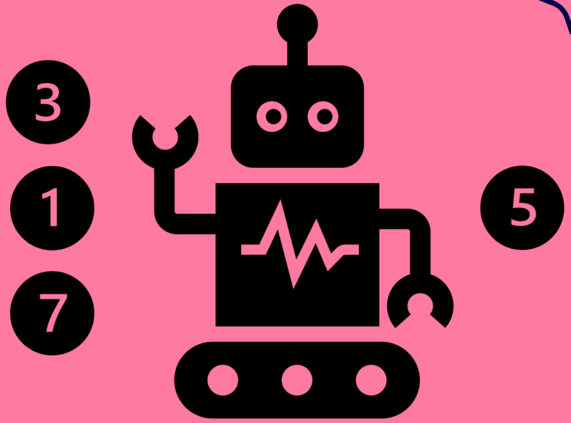
This is a ***mechanistic understanding***



# Why use computational models

With mechanistic explanations of processes, you can understand in a general sense what would happen if you gave the machine input A, B, or C

Similarly, you would know how to fix it if it isn't working



# Why use computational models

When studying the brain, we can try to understand how it works at different levels :

- Knowing the gist of what a machine is doing/ purpose
- Knowing the operations and procedures of the machine
- Knowing how the machine implements said procedures

# Why use computational models

In the case of a calculator for example:

- We know that it adds numbers together
- It does so through binary codes, circuits, and wires
- It does this by passing electrical current and opening/closing transistors

# Marr's Three levels of analysis

- **Computational**

What is the goal of the computation, why is it appropriate, and what strategy is used to carry it out?

- **Representational / Algorithmic**

How can this computational theory be implemented? What is the representation of the inputs and outputs, and what is the algorithm for the transformation?

- **Hardware Implementation**

How can the representation and algorithm be realized physically?

# Steps to making a Computational Model

- **Step 1:** Reformulate the problem in terms of a rigorous mathematical equation
- **Step 2:** Make any additional assumptions necessary not included in the conceptual framework (ad hoc assumptions)
- **Step 3:** Estimate values for the model parameters
- **Step 4:** Compare the predictions of your model to competing models. Which one produces results closets to the empirical data?
- **Step 5:** Reformulate framework and start all over again!

# Model fitting algorithms

All model fitting algorithms work by trying to find the global minima / maxima of your parameter space

If you think of all possible values of your model parameters plotted against error, fitting a model essentially tries to find the minima in this space

# Model fitting algorithms

There are many different algorithms used all with their advantages and disadvantages

One example is Least-squares which tried to minimize the sum of squared residuals (see class 8)

This goes beyond the scope of this class

# Curve Fitting

One way you can fit a model to your data is by attempting to fit polynomial curves to it with **fit()**

This function takes in your x and y data as well as the desired poly function you wish to fit

# Polynomial Functions

If you remember from high school math, these are functions of the form:

$$y = intercept + B_1 X_1 + B_2 X_2^2 + B_3 X_3^3 \ldots + B_n X_n^n$$

Where you describe y as a function of several orders of x

Note that a *first order* polynomial has *2 parameters*

# Polynomial Functions

fit(x, y, 'poly3') would fit the following to your data:

$$y = intercept + B_1X_1 + B_2X_2^2 + B_3X_3^3$$

# Model Error

- Making predictions comes with some form of error, models are never perfect just like humans aren't (f.t. Hannah Montana)
- But there are many methods to compute how wrong your predictions are from your data

# Model Error

It is important to note, however, that when you want to assess how good your model is predicting something you need to use data that your model has not seen (i.e., not been trained on)

This is done to avoid OVERFITTING

# Error

## Error

Observed minus predicted

## Squared Error

Square the difference between observed and predicted

## Root MSE

Square root of mean square error

# Error Formulas

$$Error = X_{obs} - X_{pred}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_{obs} - X_{pred})^2$$

$$SqErr = (X_{obs} - X_{pred})^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{obs} - X_{pred})^2}$$

# Picking the best models based on error

Generally speaking, the best fitting model is the one that predicts the data the ***best*** → this ultimately entails minimizing error (regardless of which kind of error you are measuring)

It is important to not that you want a model that minimizes error within your sample and across all other possible samples (i.e., generalizes)

But there are other considerations beyond minimizing error

# Picking the best model based on observations

Instead of picking the best model by minimizing error,

you can alternatively pick the best model by trying

to maximize the likelihood

Said another way, picking a model that maximally

produces data like yours

# Likelihood is NOT probability

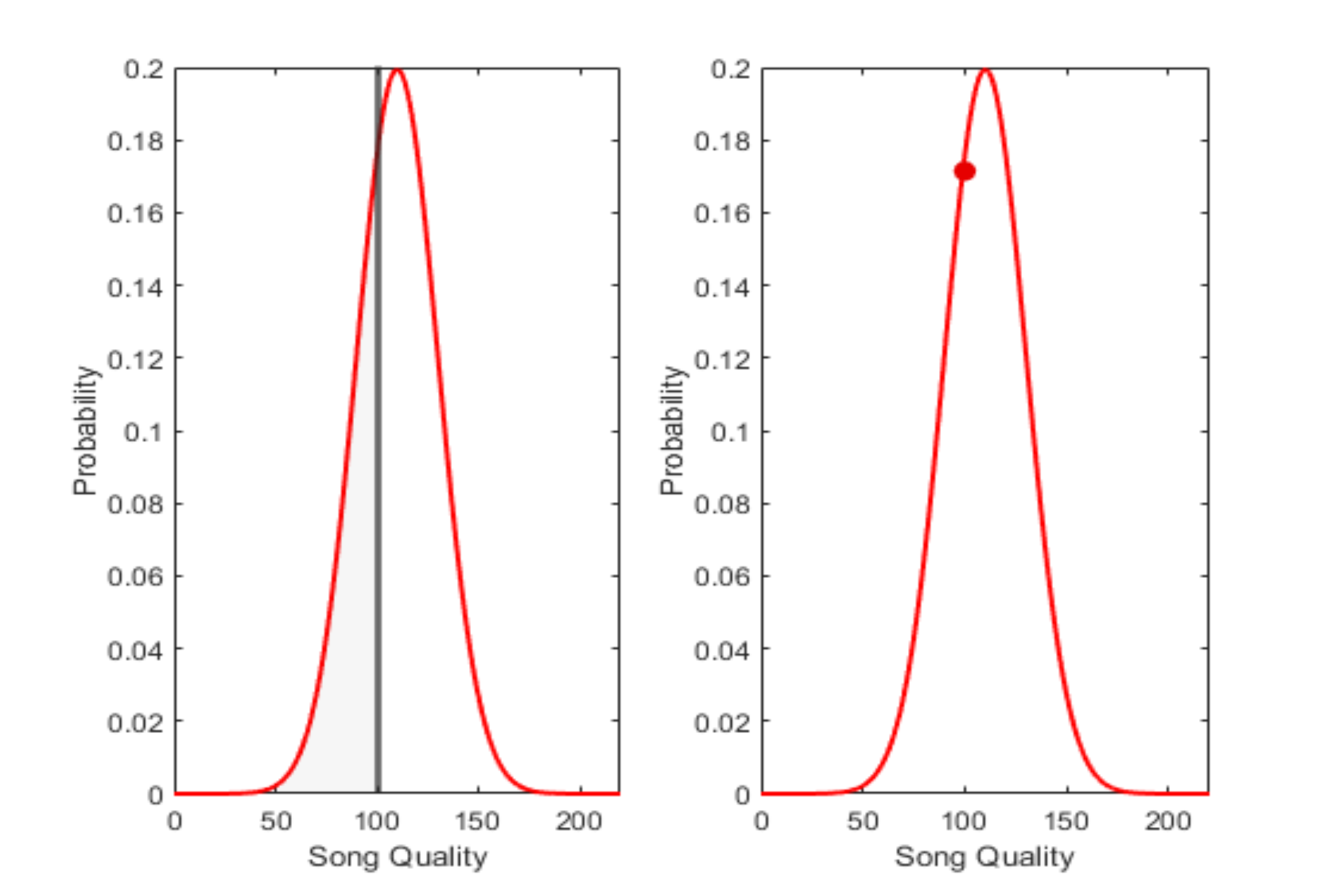
- In stats probability represents possibilities that are **exhaustive** and **mutually exclusive** (e.g. a coin toss), they always add up to 1
- Likelihoods do not need to be exhaustive or exclusive

# What is likelihood

Looking at a probability density function we can understand the differences between probability and likelihood

Probability is the area under the curve up to a value xi

Likelihood is the value of the function at xi

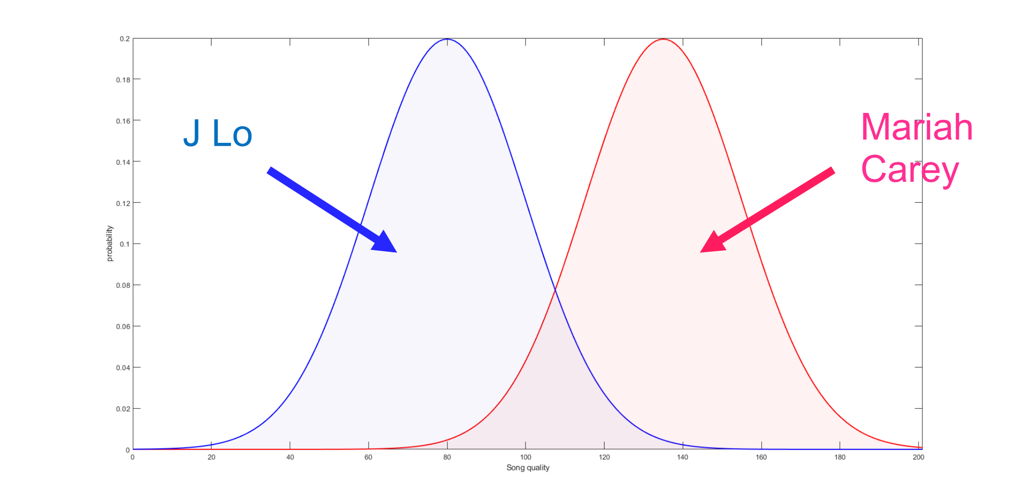

0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Probability
0
50
100
150
200
Song Quality
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Probability
0
50
100
150
200
Song Quality

# Likelihood Example

Let us assume we are a data scientists at Vogue and want to find out which musician is the most likely to have sung the latest hit song

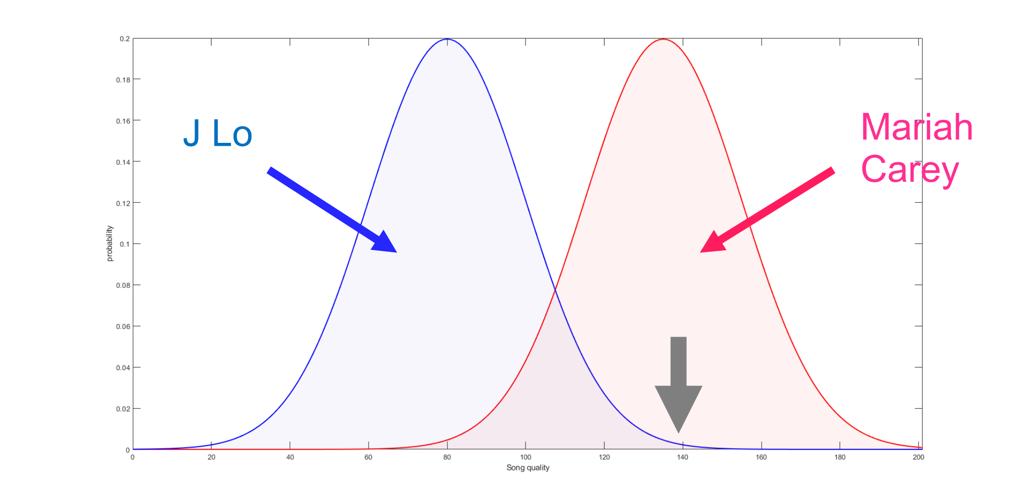
# Likelihood Example

We know the quality of the song and the PDF of different artists' song quality



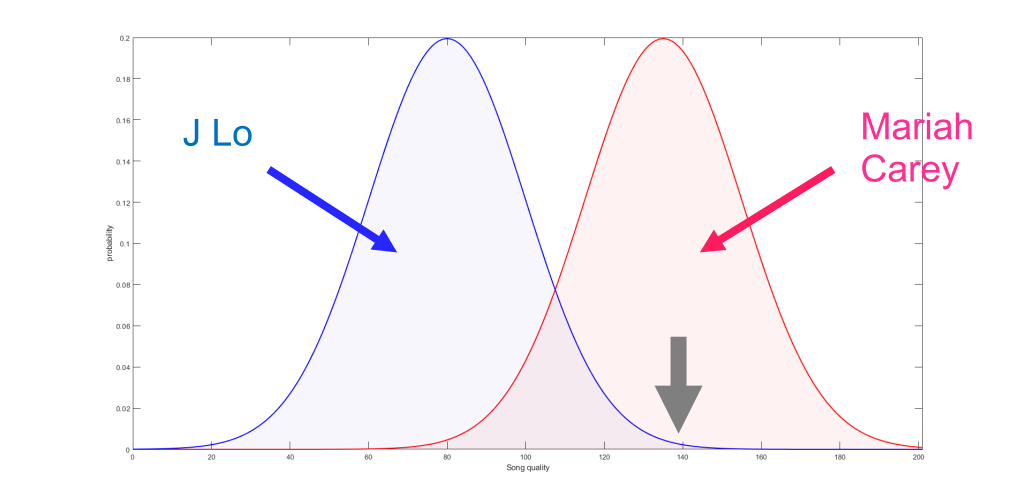
# Likelihood Example

We know the quality of the song and the PDF of different artists' song quality



# Likelihood Example

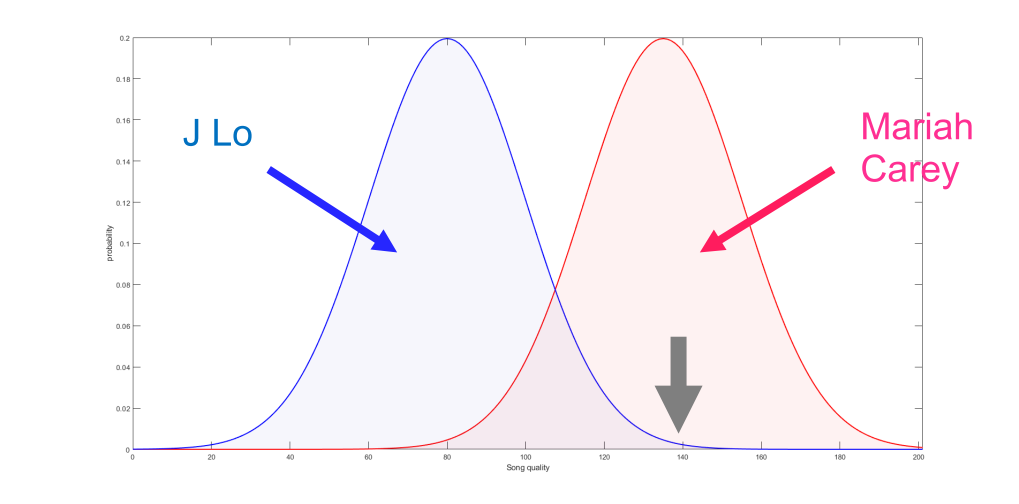
Given the song quality what parameter (i.e., artist) is likely to generate the data (i.e., song) ?



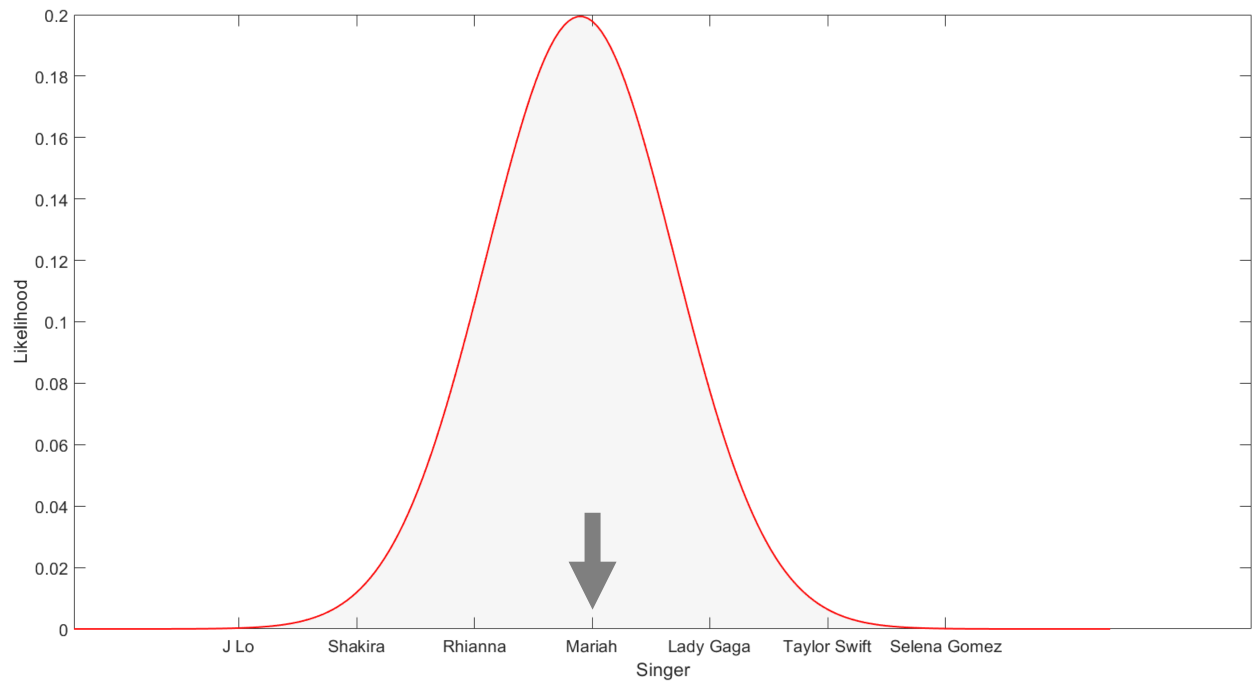
# Likelihood Example

This is a problem of maximizing the likelihood of a

parameter given data



# Likelihood Example



# Likelihood

Therefore, we can see how maximizing the likelihood can generate a model that best explains the distribution of our data based on an unknown parameter (random variable)

# Likelihood

The goodness-of-fit of models will be described in Log likelihoods. These values by themselves are meaningless. They can only tell you how well a model fits RELATIVE to another model fit on the SAME data.

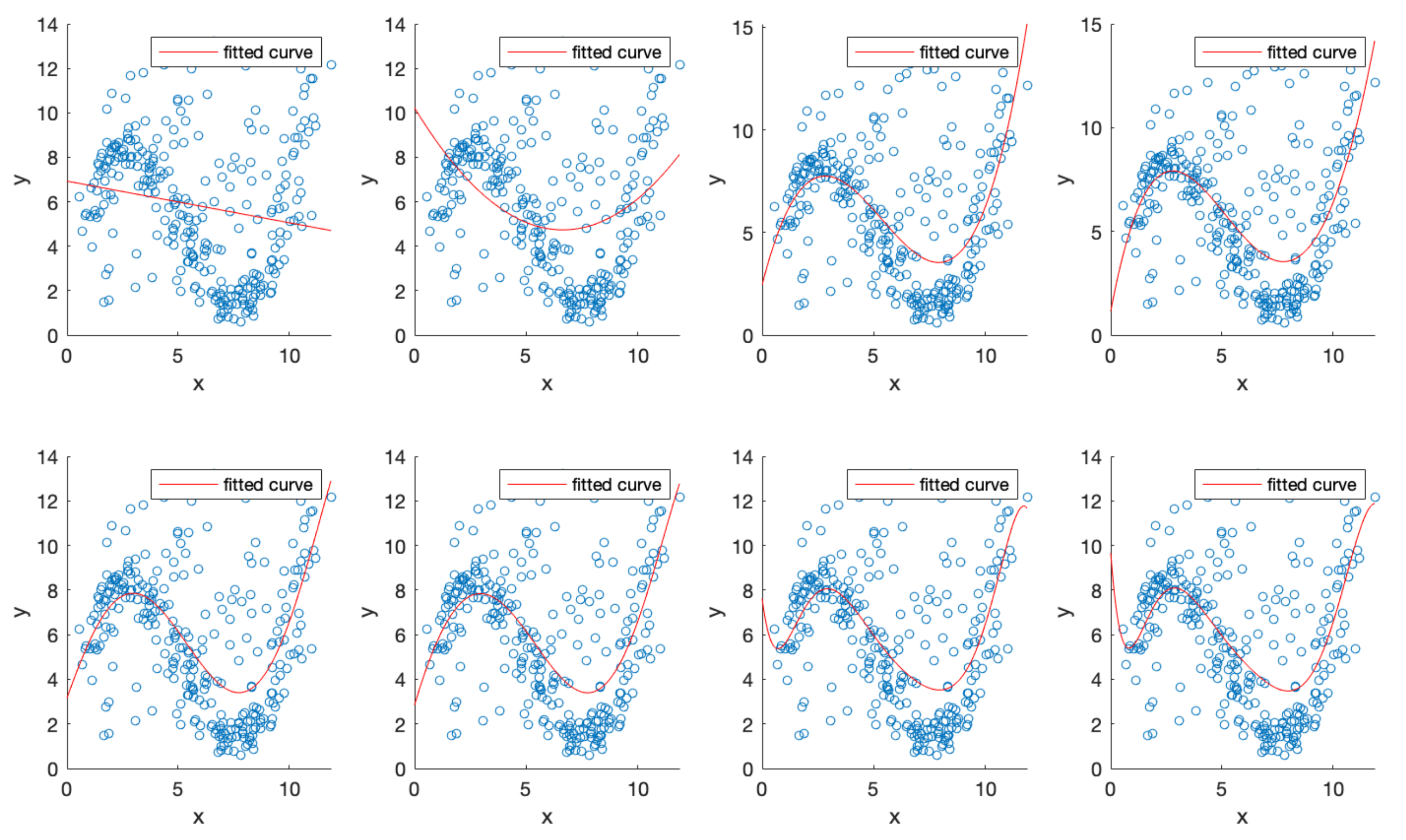
# Degrees of freedom

Each model has a different number of 'free' variables / values that can vary

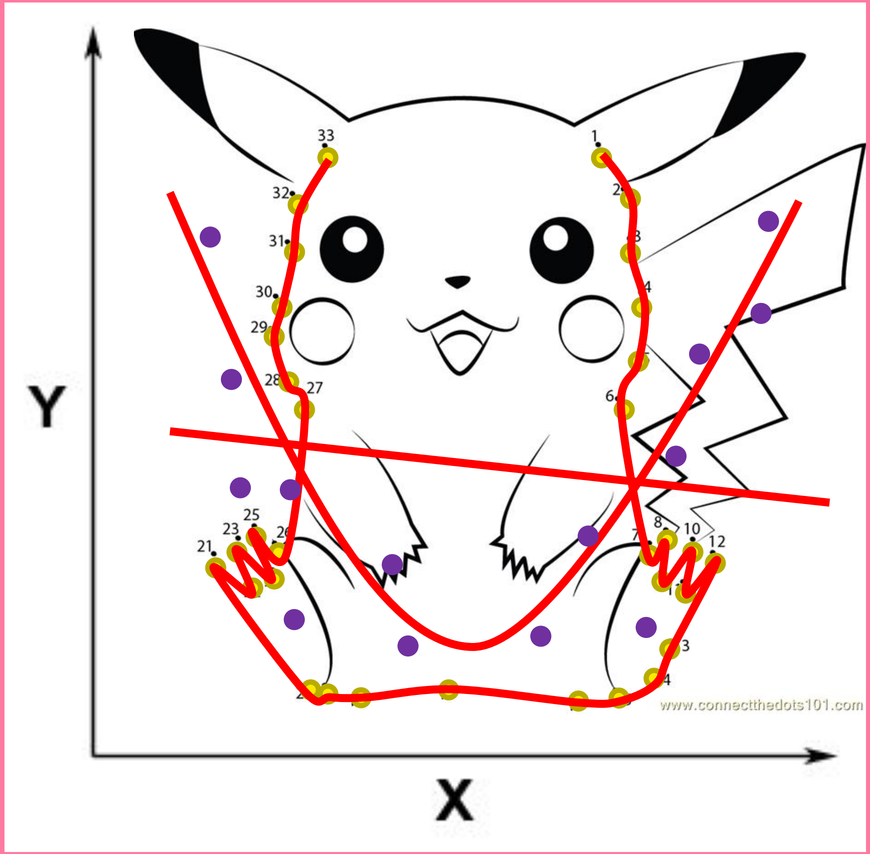
The more your model is free to vary, the better it can fit data

Visually, this is the equivalent of adding more bends to your data

While we love flexibility, models that are too flexible are hard to interpret and tend not to generalize well


fitted curve
x
y

# Overfitting




$$Y = \beta X + \alpha$$

$$Y = \beta X^2 + \gamma X + \alpha$$

$$Y = \beta X^2 + \gamma X + \frac{\xi \sin\left(\tau\left(\sqrt{X} - \varphi\right)\right)}{\sqrt[3]{e^{-\omega X}}} + \alpha$$

# Occam's Razor

"Plurality should not be posited without necessity."

Heuristic for the development of scientific models

- Parsimonious
- Falsifiable
- Related to Overfitting

# Model selection

We can try and quantify a measure that balances both the **fit** of the model as well as **parsimony** (i.e., the number of parameters)

**Akaike information criterion (AIC).**

$-2\log L(\theta) + 2k$

**Bayesian information criterion (BIC)**

$-2\log L(\theta) + k^{*}\log(T)$

Where k is the number of parameters fit to T observations