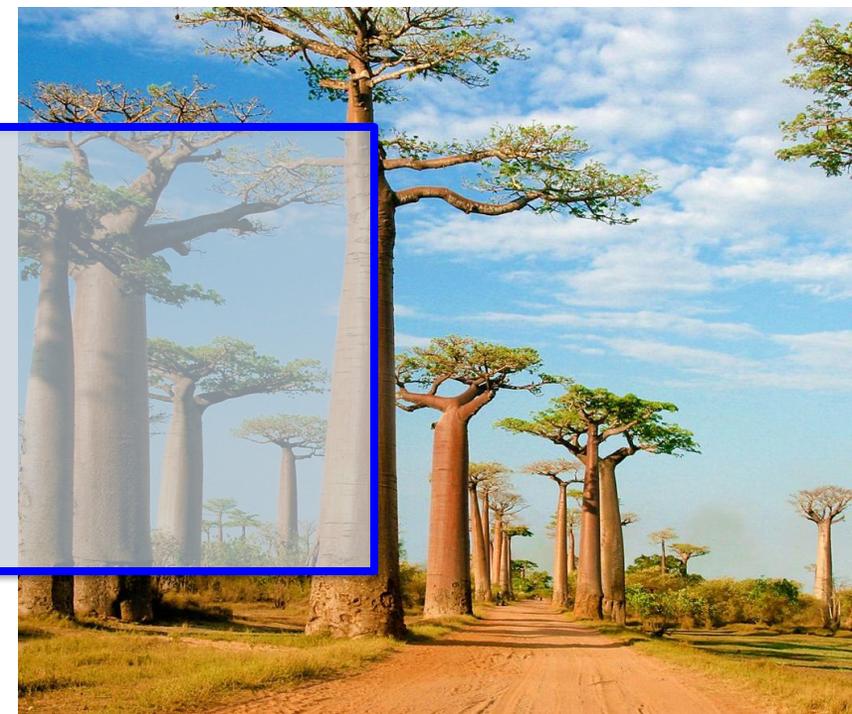
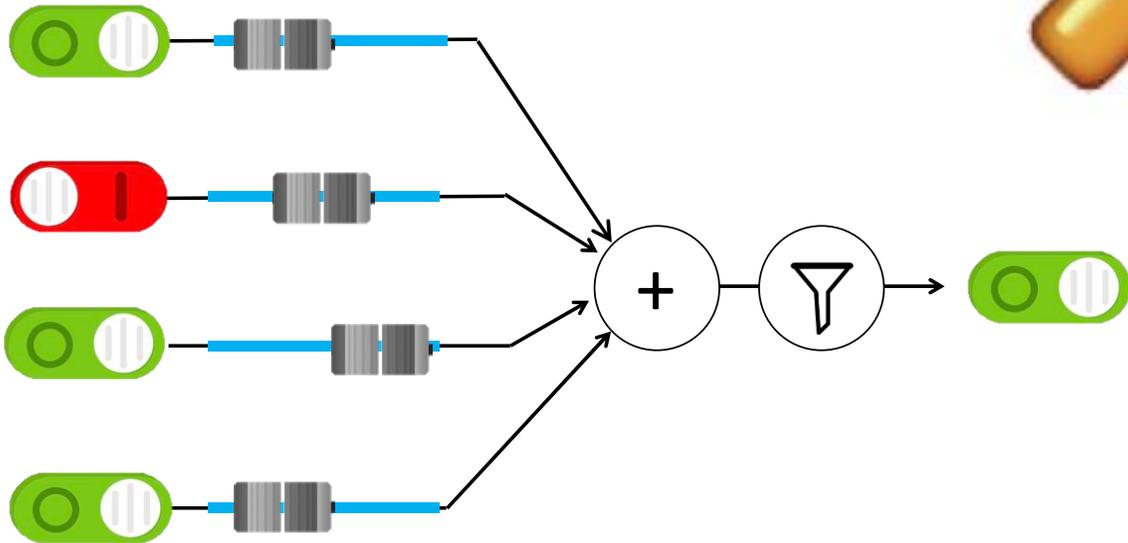
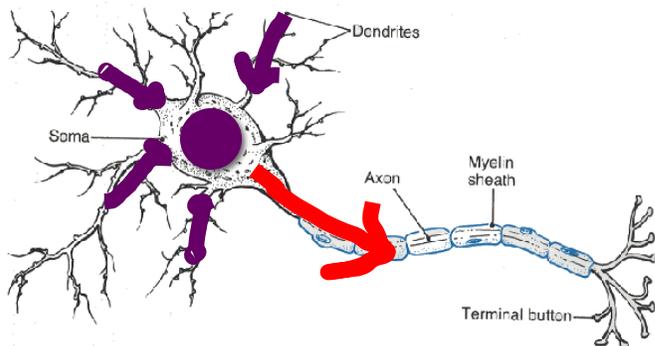


**Diffusion**  
Chris Piech  
CS109, Stanford University

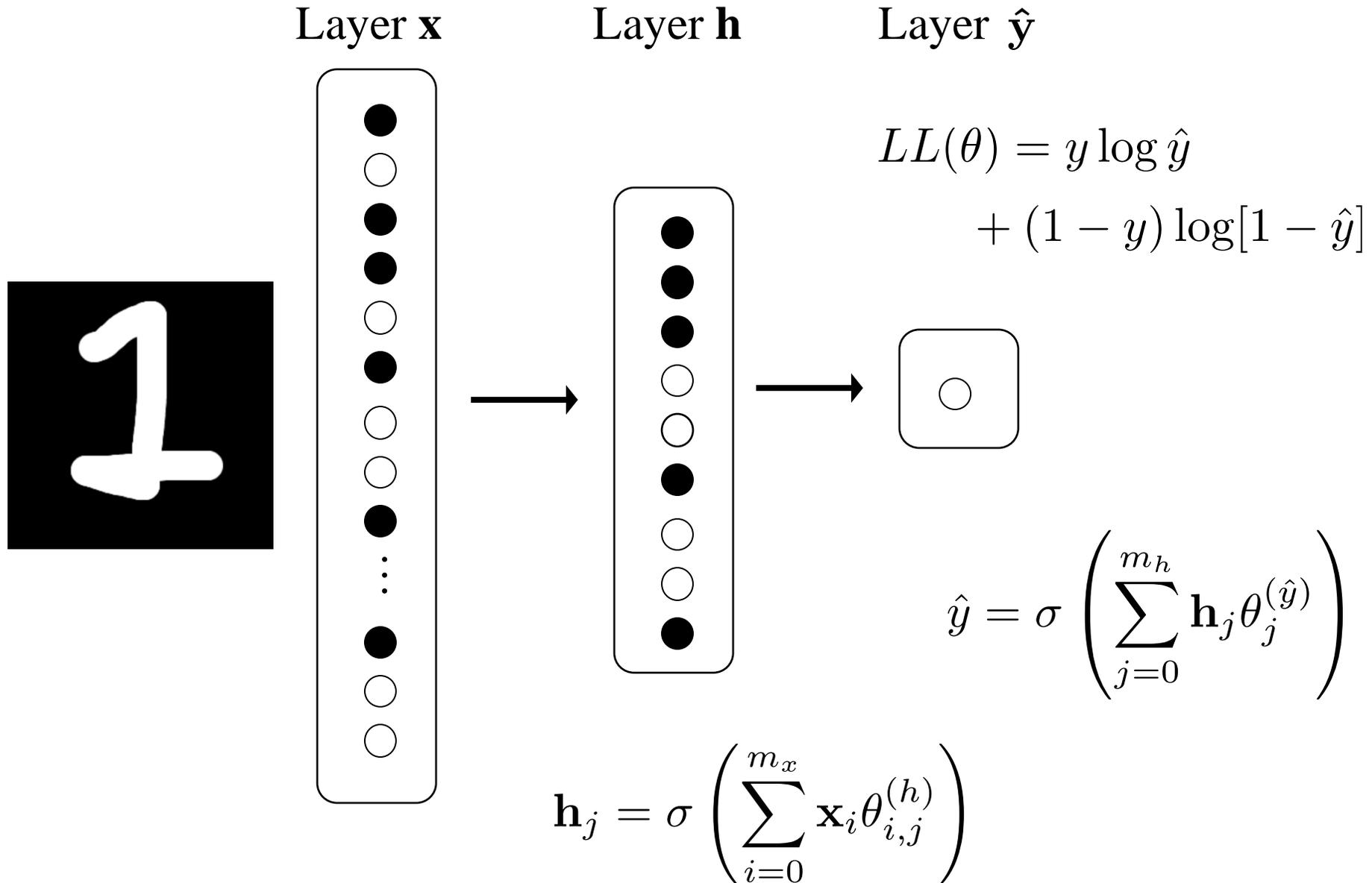


# Review

# Artificial Neurons

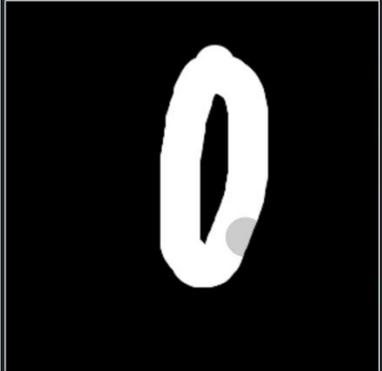


# Deep Learning



# Demonstration

Draw your number here



X  

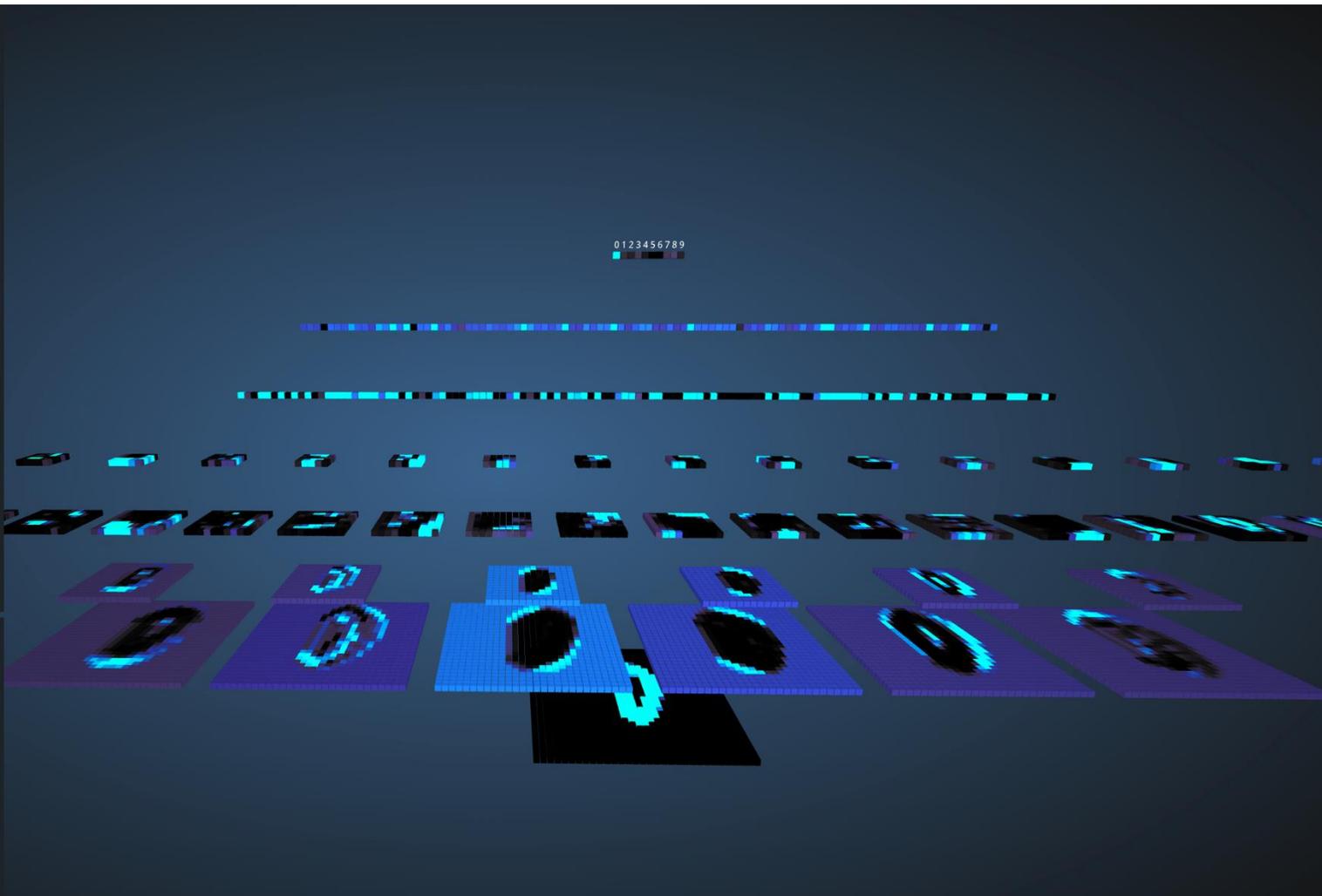
Downsampled drawing:

First guess:

Second guess:

**Layer visibility**

Input layer	Show
Convolution layer 1	Show
Downsampling layer 1	Show
Convolution layer 2	Show
Downsampling layer 2	Show



<https://web.archive.org/web/20211117115916/https://www.cs.ryerson.ca/~aharley/vis/conv/>

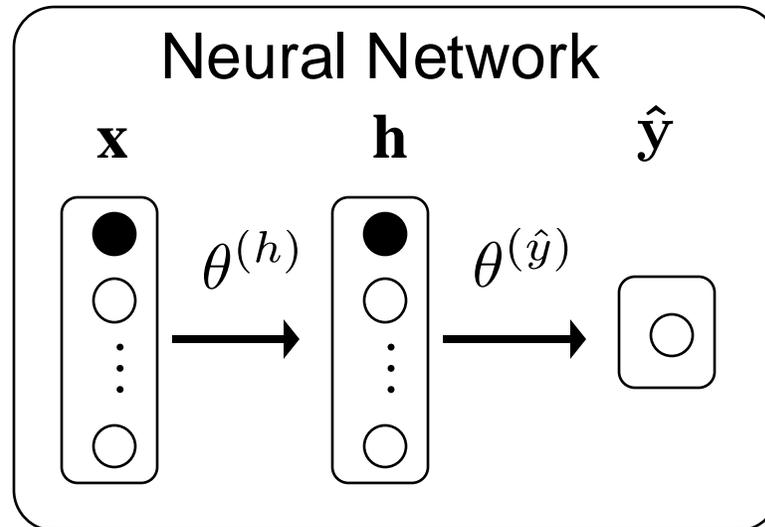
# Derivative Goals

LL with respect to  
output layer params

$$\frac{\partial LL(\theta)}{\partial \theta_i^{(\hat{y})}}$$

LL with respect to  
hidden layer params

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}}$$



# Back Propagation with Chain Rule

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \text{[Peashooter] [Tombstone] [Cactus]}$$

$$\text{[Peashooter]} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

$$\text{[Tombstone]} = \hat{y}[1-\hat{y}]\theta_j^{(\hat{y})}$$

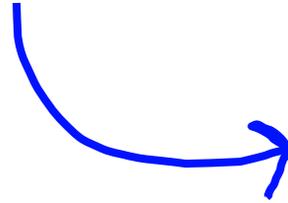
$$\text{[Cactus]} = \mathbf{h}_j[1-\mathbf{h}_j]\mathbf{x}_j$$

# Deep Learning Code

Train model



Build model



```
38 def run_train(train, model):
39     # this function can score how good a precision is
40     loss_function = nn.CrossEntropyLoss()
41     # this function can adjust the model to make better predict
42     optimizer = torch.optim.Adam(model.parameters(), lr=0.0001)
43
44     # get labelled examples to train on
45     for image, truth in train:
46         # predict the label
47         pred = model(image)
48         # score how well we did
49         loss = loss_function(pred, truth)
50         # learn from our mistakes
51         optimizer.zero_grad()
52         loss.backward()
53         optimizer.step()
```

```
8 def main():
9     # get the data
10    train, test = download_data()
11    print(f"Training examples: {len(train.dataset)}")
12    print(f"Test examples: {len(test.dataset)}")
13
14    # a very simple and fast nn
15    model = nn.Sequential(
16        nn.Flatten(),
17        nn.Linear(28*28, 1024),
18        nn.ReLU(),
19        nn.Linear(1024, 1024),
20        nn.ReLU(),
21        nn.Linear(1024, 10)
22    )
23
24    # see how bad it is before training
25    accuracy = run_test(test, model)
26    print(f"Untrained, Accuracy: {accuracy}")
27
28    # train the model
29    print("training...")
30    for epoch in range(100):
31        # learn from examples
32        run_train(train, model)
33        # measure how well we are doing
34        accuracy = run_test(test, model)
35        print(f"Epoch {epoch}, Test Accuracy: {accuracy}")
36
```

# Types of Machine Learning Tasks

---

Multi-Class  
Classification

Regression

Reinforcement  
Learning

Generation

Today: Diffusion

# How Does This Work?

create an image of a tree

another tree



Here's an image of a serene tree in a peaceful meadow.

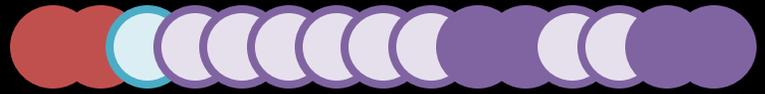


Here's another beautiful tree, this time showcasing the vibrant colors of autumn.

Probability gives you a new  
lens on the world

# Original deep learning for education paper

KHAN  
Student



1

10

Exercise index

Exercise Type:

Answer:



Solving for x-intercept



Correct



Solving for y-intercept

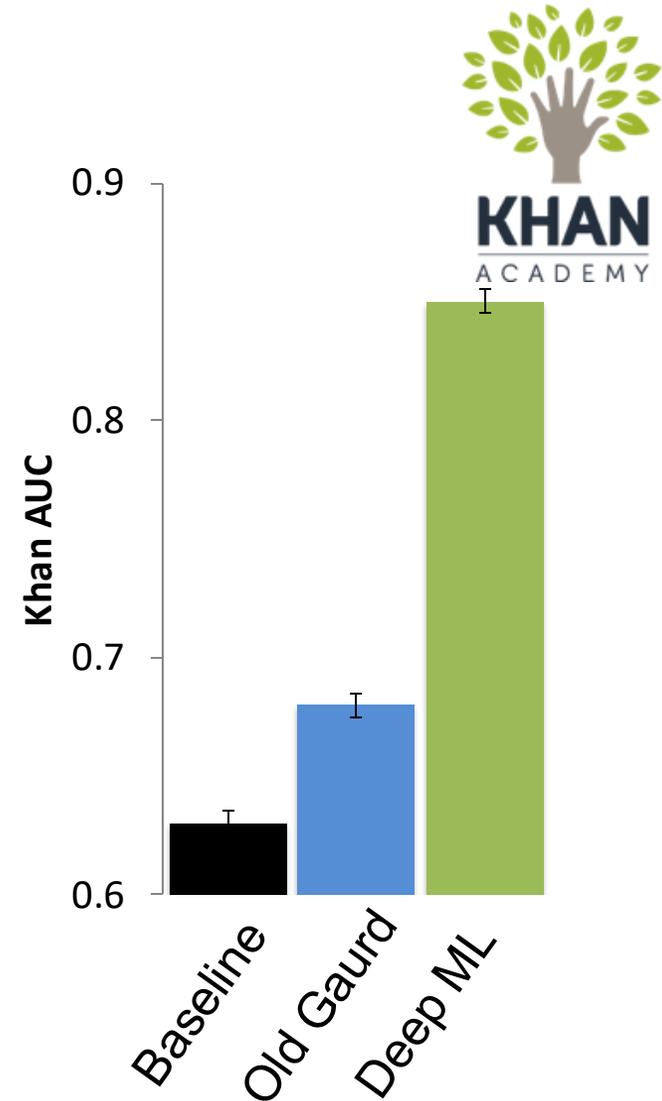


Incorrect

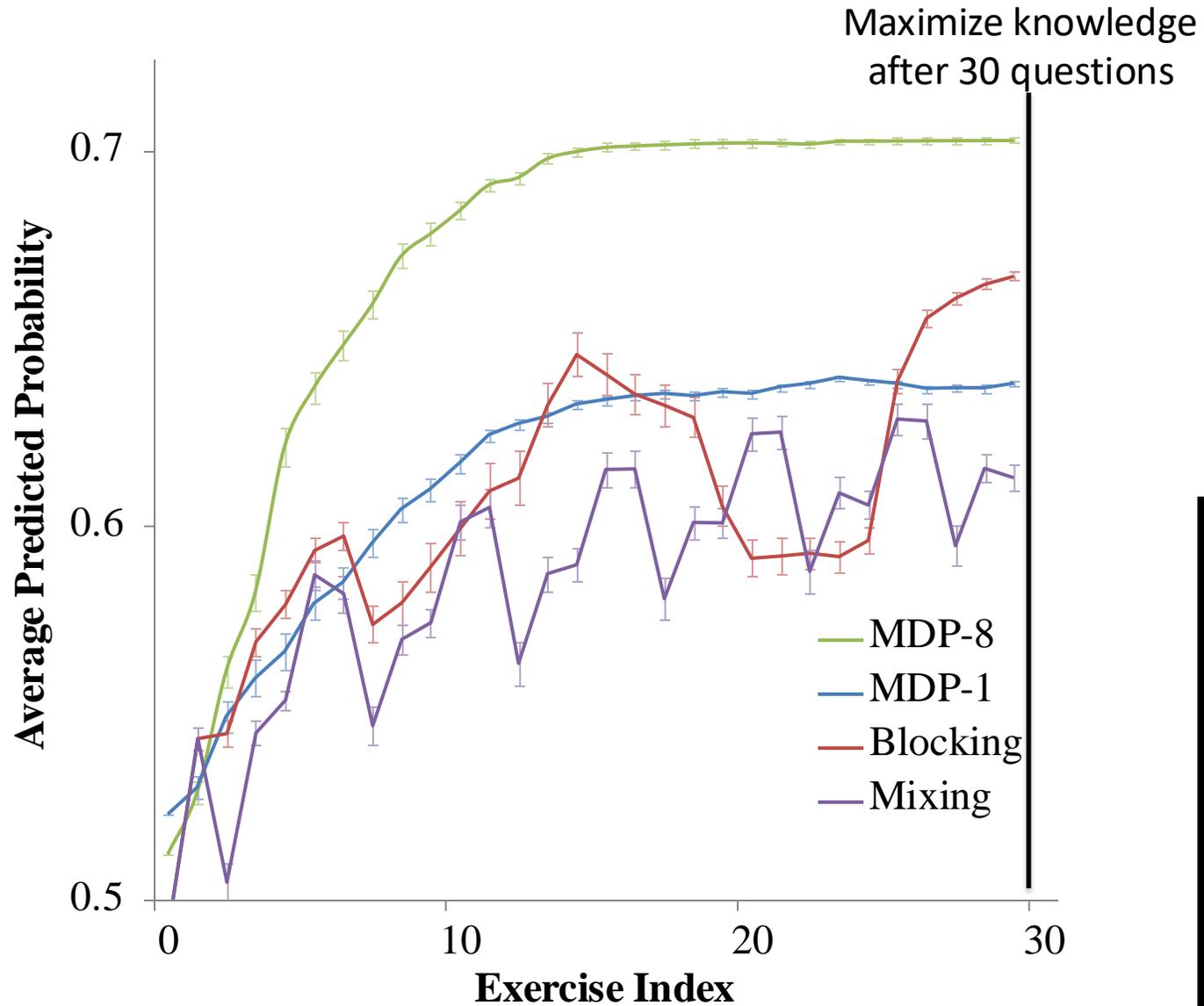


Square roots

Powers DuoLingo Bird Brain (as of 2023)

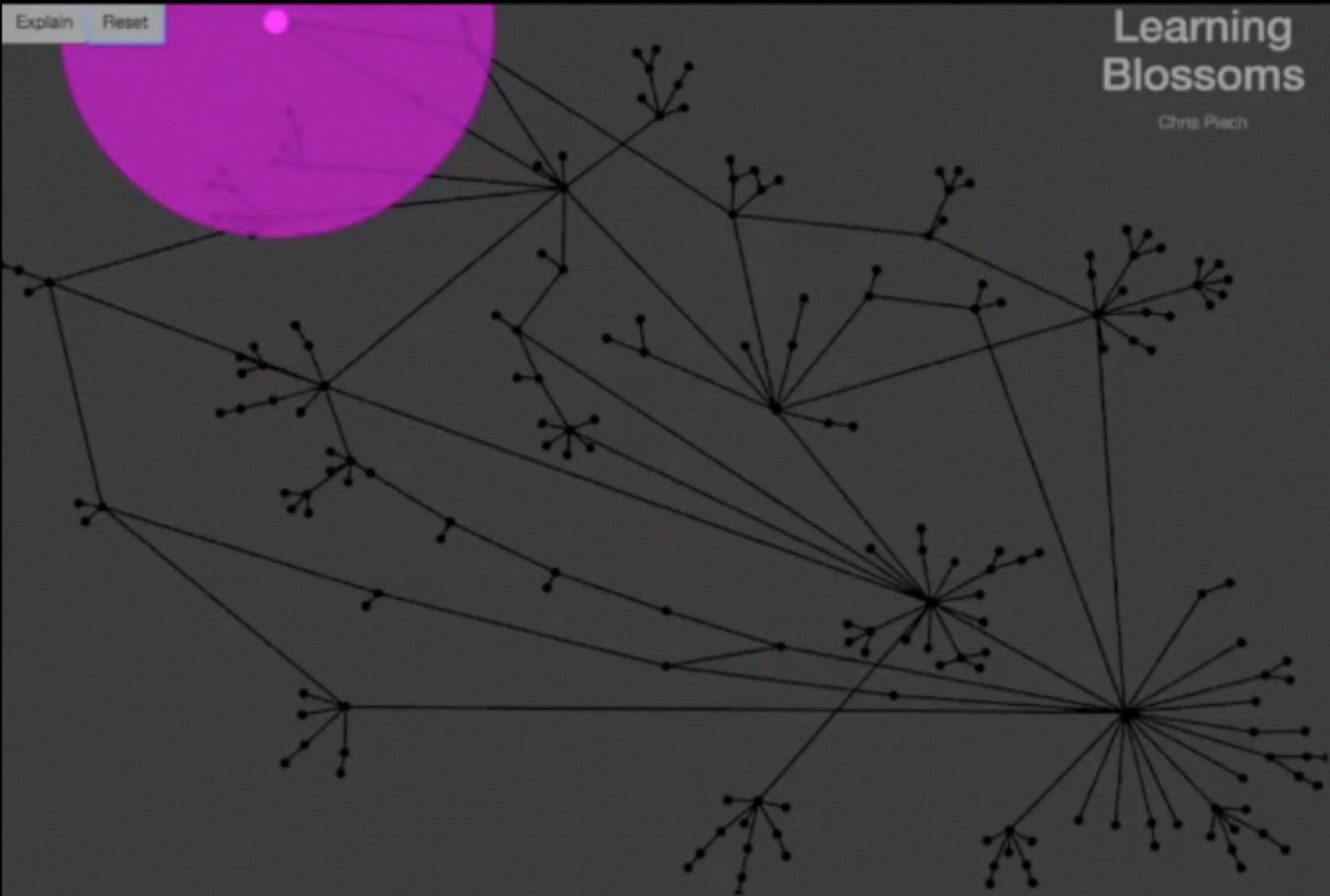


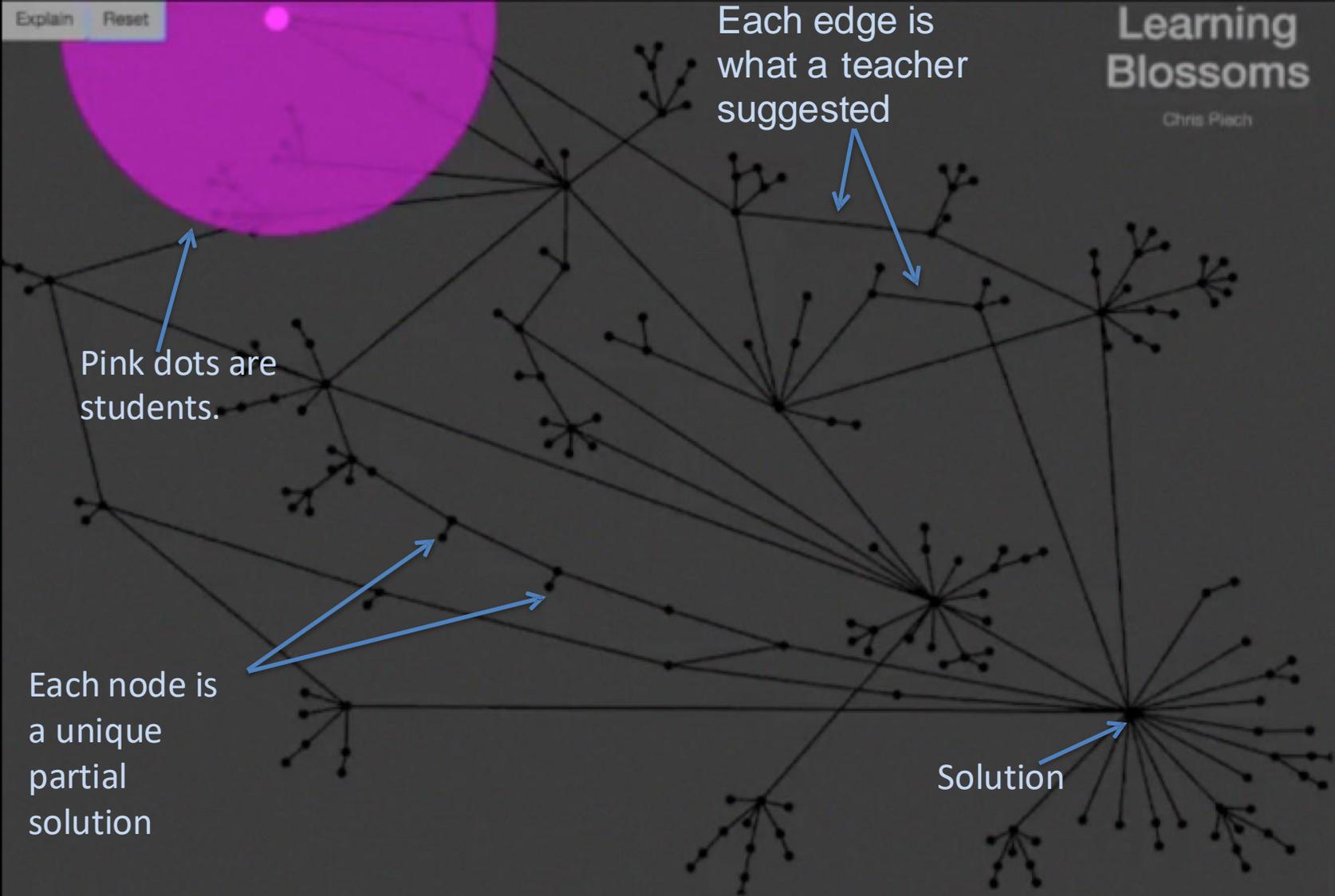
# Optimal Teaching



Used in DuoLingo  
Bird Brain (last  
checked in 2023)



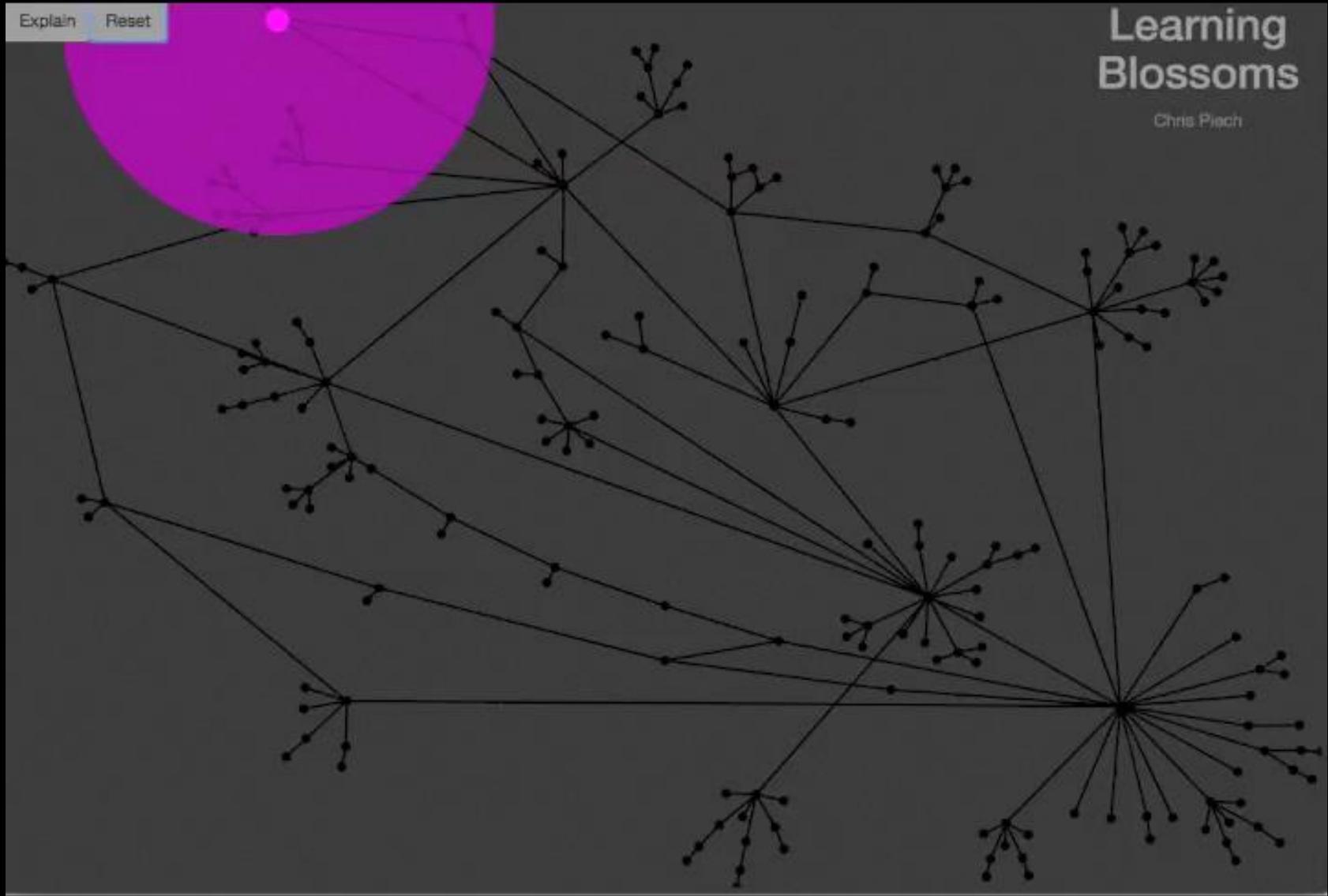




Explain Reset

# Learning Blossoms

Chris Piech



# The Crowd is Un-wise

Temporal methods tried:

Shortest path

Min Time

Expected Success

Reinforcement learning

Most Common Next

Most Popular Path



when run

move forward

move forward

turn left ↻ ▼

when run

move forward

turn left ↻ ▼

18%

when run

move forward

move forward

turn right ↻ ▼

45%

when run

move forward

move forward

turn left ↻ ▼

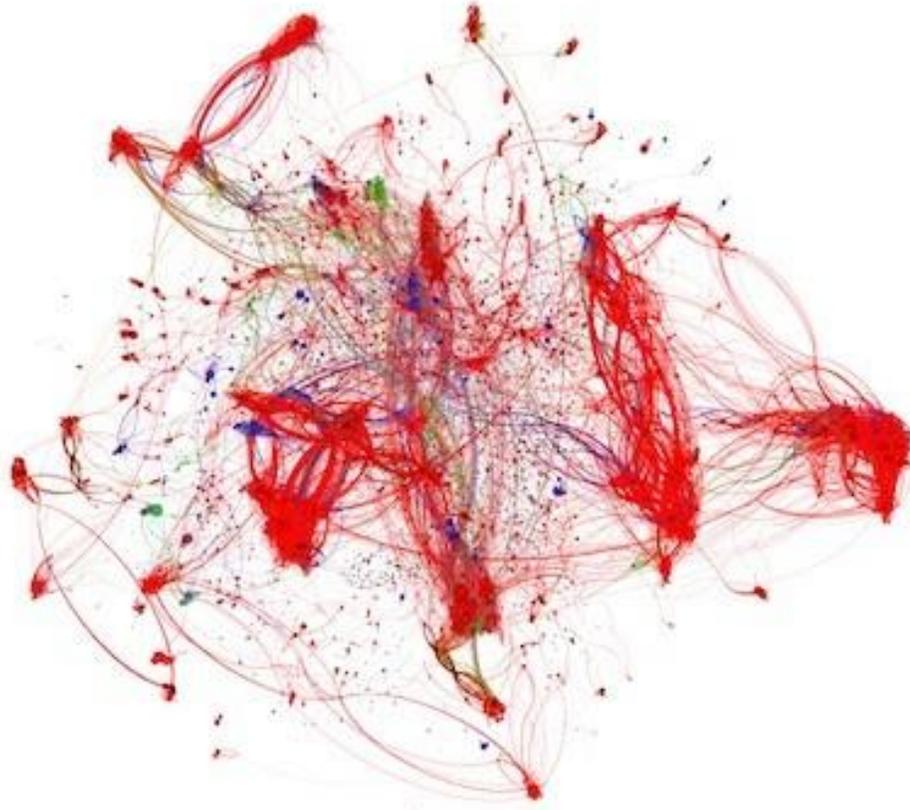
move forward

12%



# Hard Problem

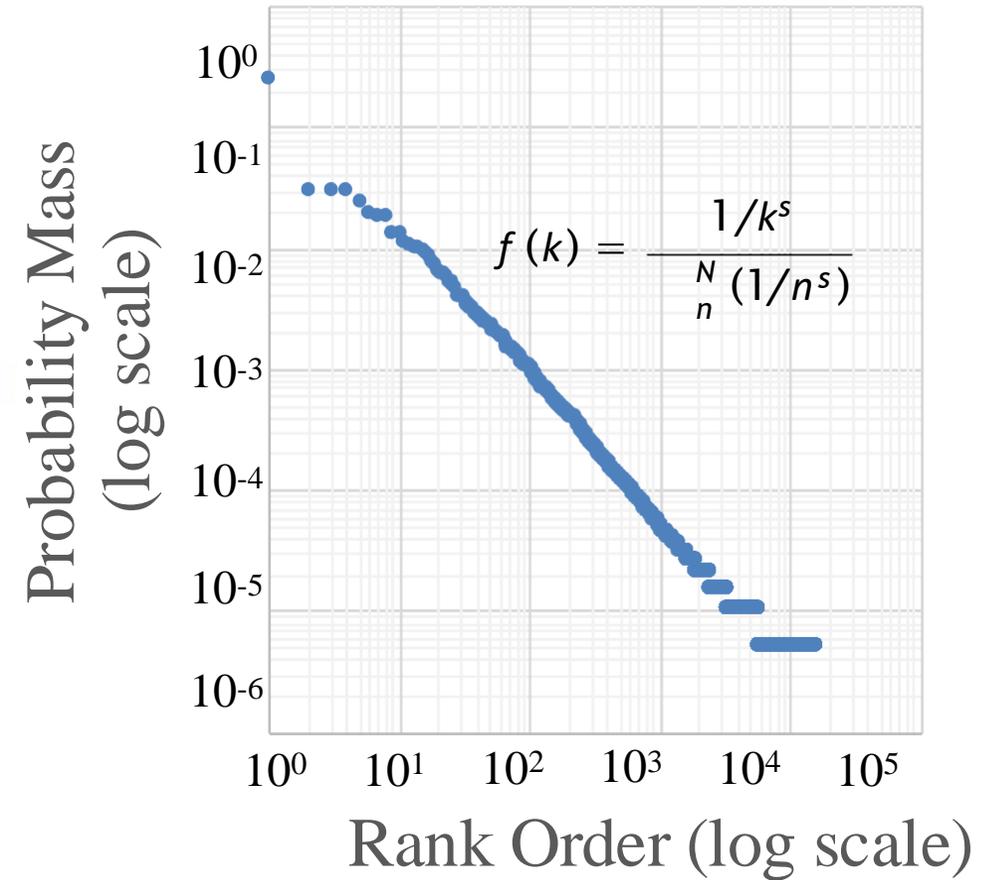
Brute force solution?



1 million unique solutions to  
programming Linear Regression

WWW 2014

## Code Zipf Plot



# They are all Zipf!

## (a) Datasets in Computational Education

Code.org Problem 8

Powergrading P13

What is one reason the original colonists came to America?

- Religious freedom
- For religious freedom
- Freedom

- declared our independence from england
- religeous freedom
- as a criminal punishment

- to create a new colony
- to find better economic prospects
- to break away from the church in great britain

CS1: Liftoff

Write a Java Program to print the numbers 10 down to 1 and then write liftoff. You must use a loop.

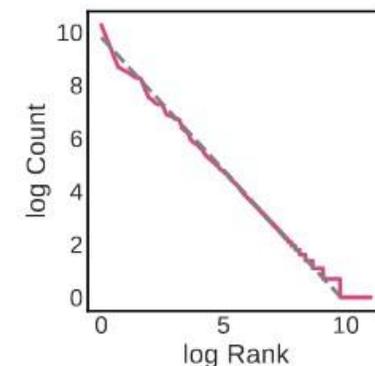
```
public void run() {
    for (int i=START; i>0; i -=1)
    {
        println(i);
    }
    println("Liftoff");
}
```

```
public void run() {
    int x = START;
    int y = 1;
    int z = 9;
    while (x>=1) {
        println(x);
        x=z;
        z=x-y;
    }
    println("Liftoff");
}
```

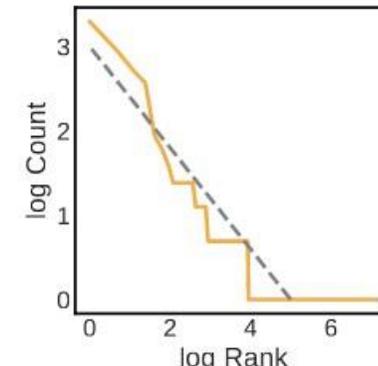
PyramidSnapshot

Use the graphics library to construct a symmetric and centered pyramid with a base width of 14 bricks.

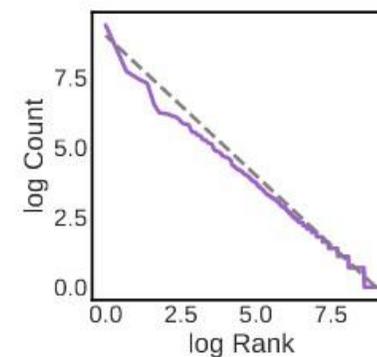
(b) Code.org P8



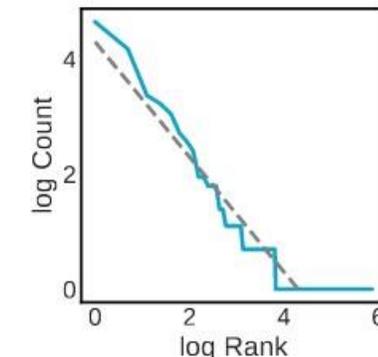
(c) CS1: Liftoff



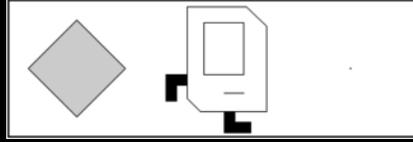
(d) Pyramid



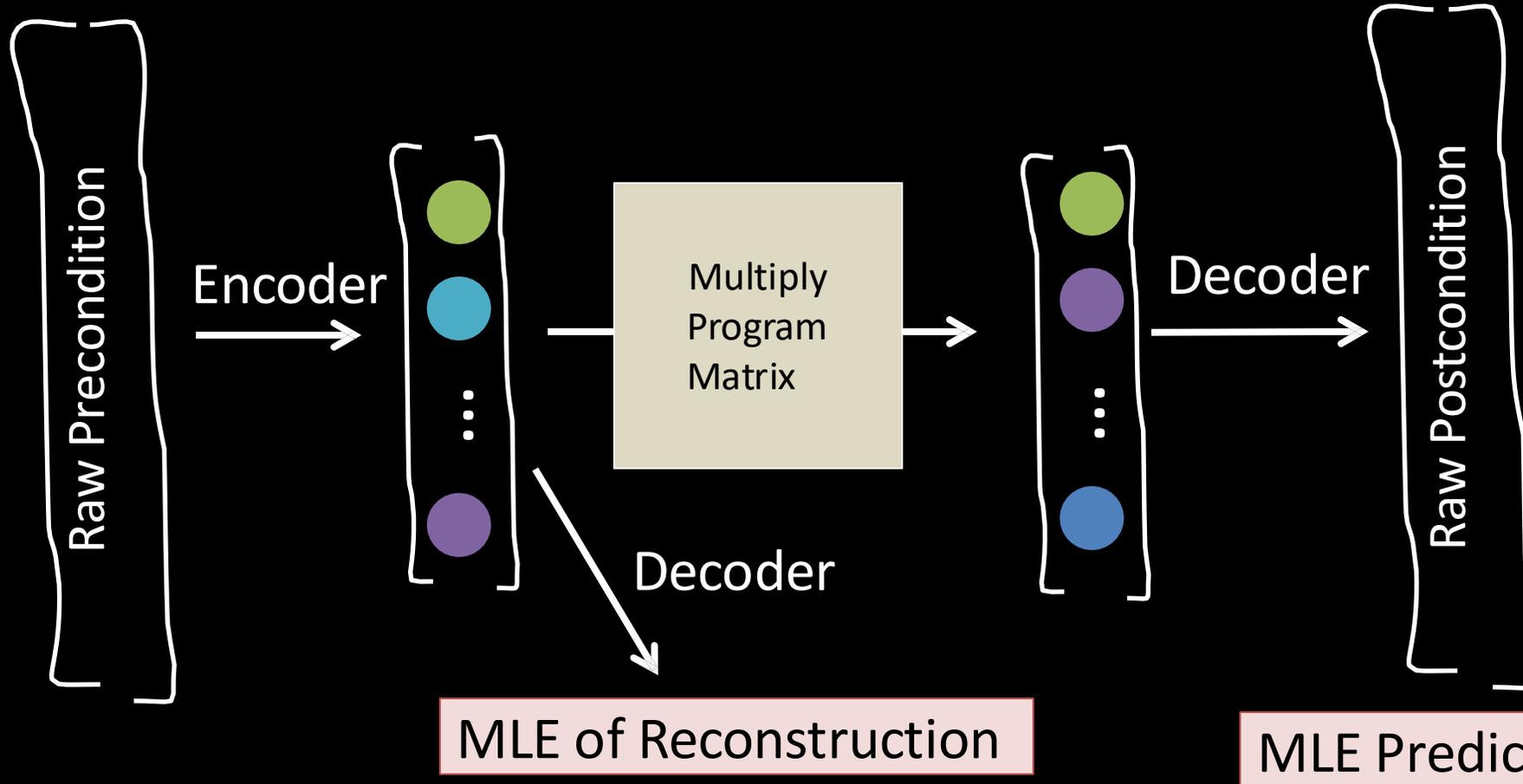
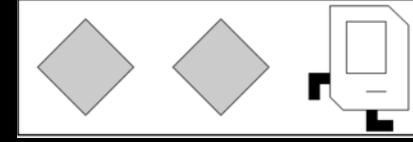
(e) Powergrading



# Neural Network for Programs



```
method step() {  
  putBeeper();  
  move();  
}
```

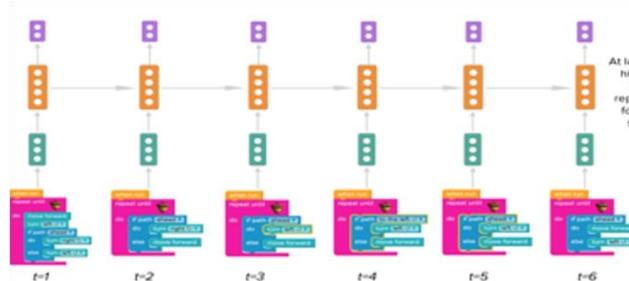
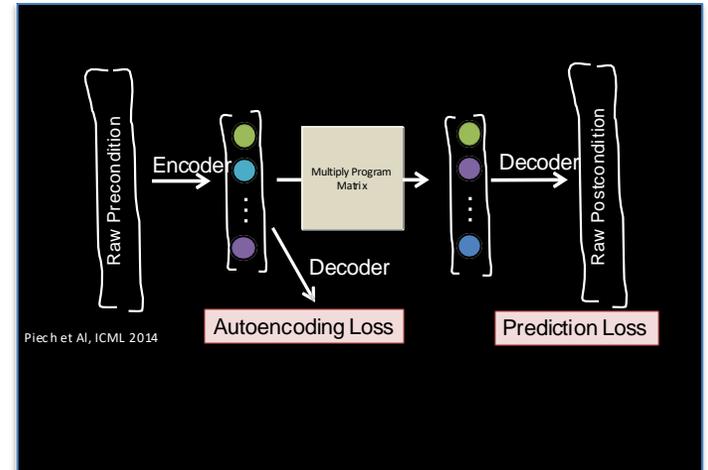
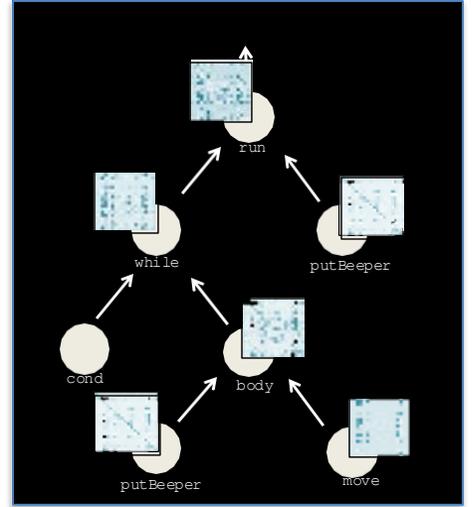
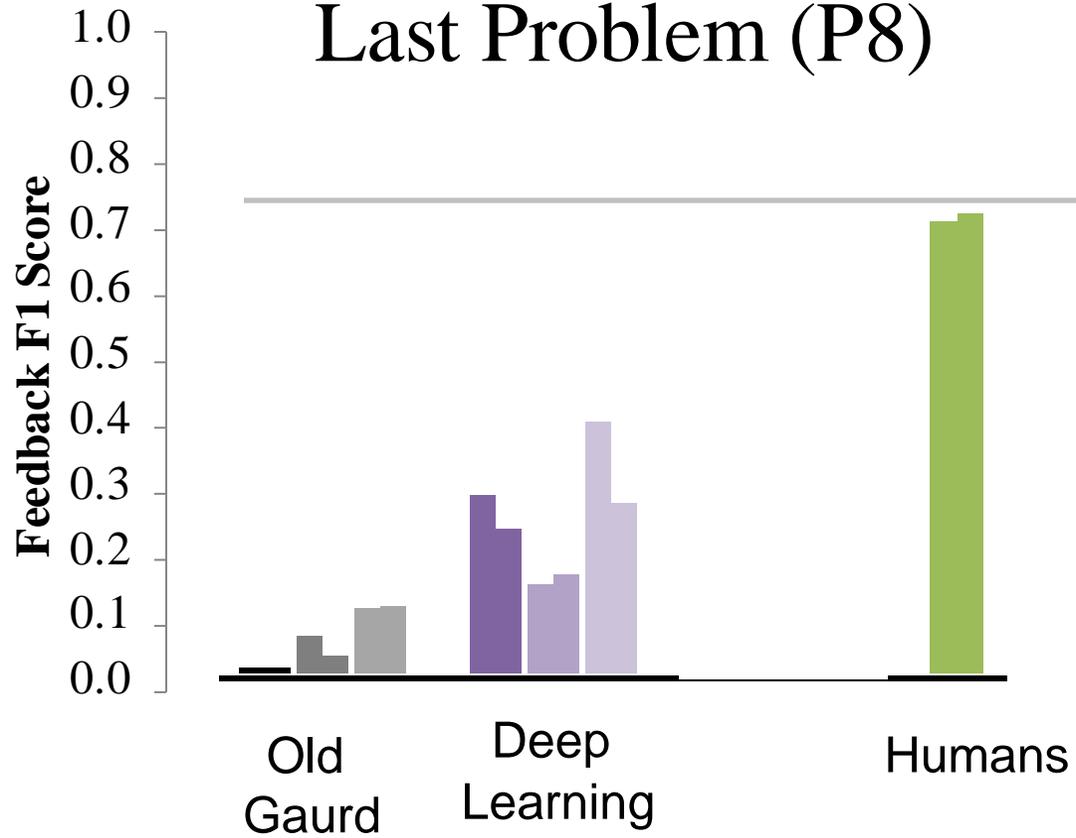


\*coded pre-tensor flow



# Inaccurate, Uninterpretable, and Data Hungry

Label student code



# Humans Don't Need Much Data

Single training example:

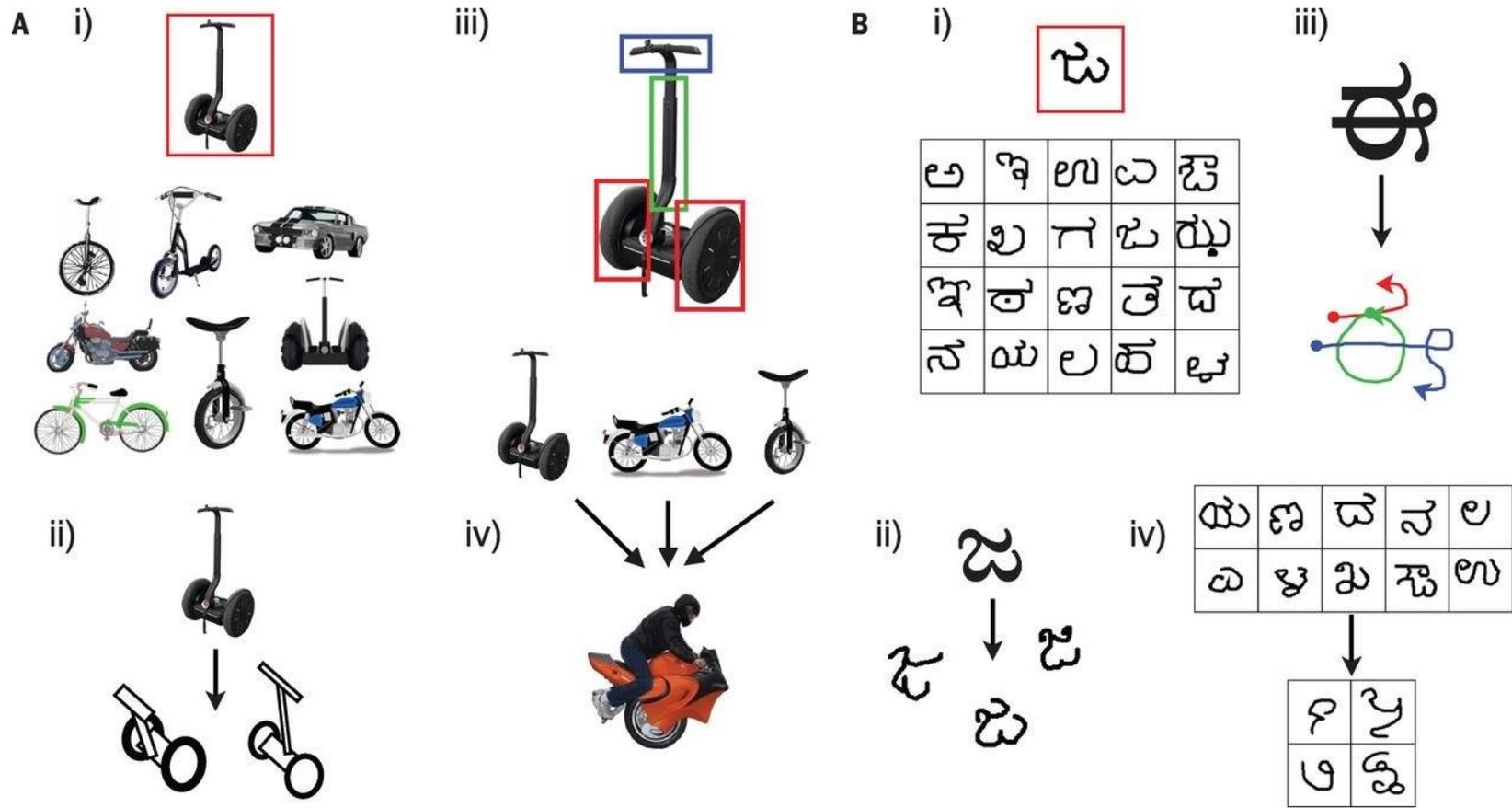
3

Test set:

3 3 3  
3 3 3  
3 3 3



Fig. 1 People can learn rich concepts from limited data.



Brenden M. Lake et al. Science 2015;350:1332-1338



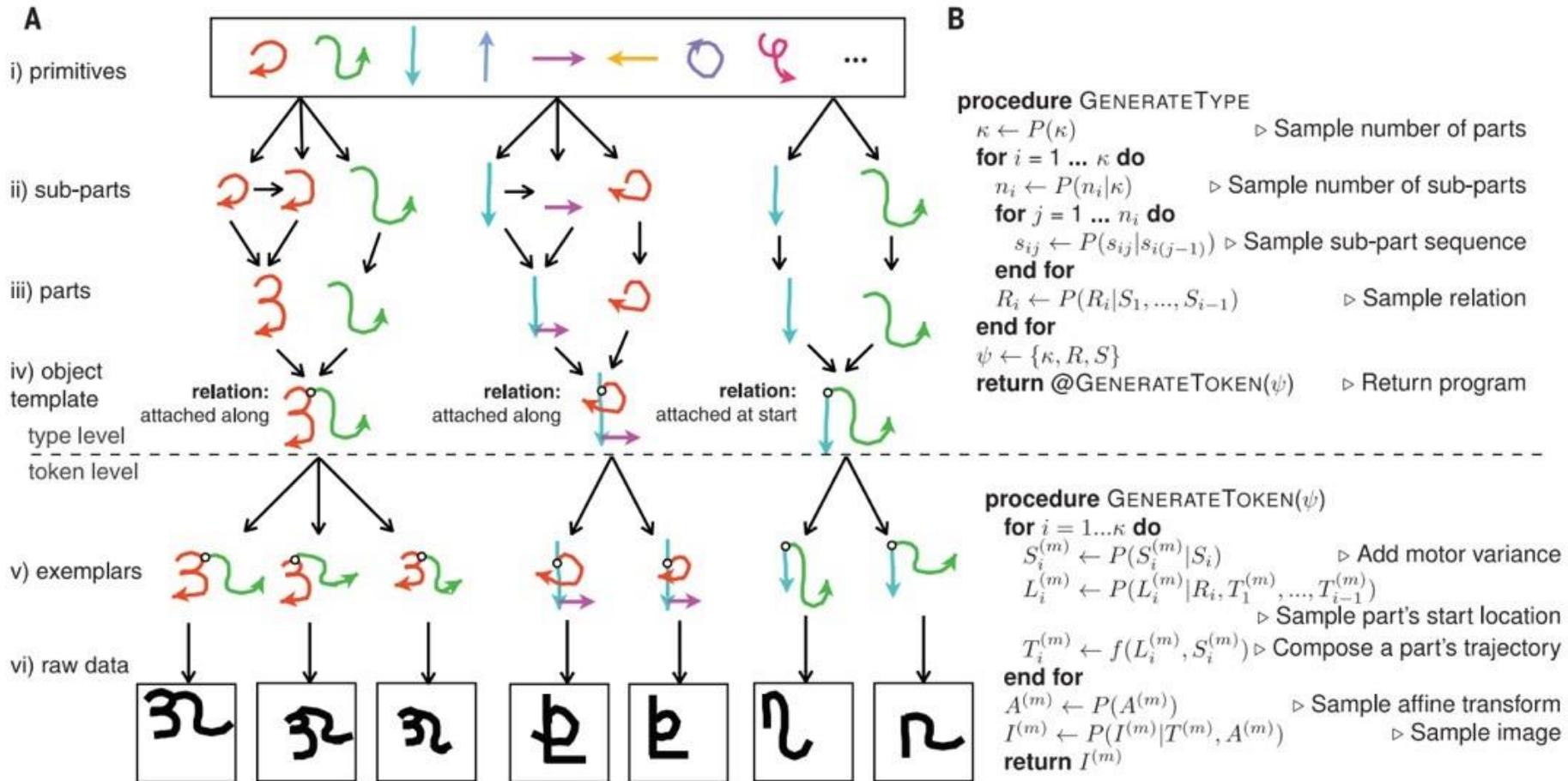
Fig. 2 Simple visual concepts for comparing human and machine learning.



Brenden M. Lake et al. *Science* 2015;350:1332-1338

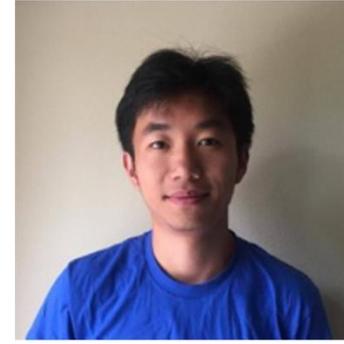
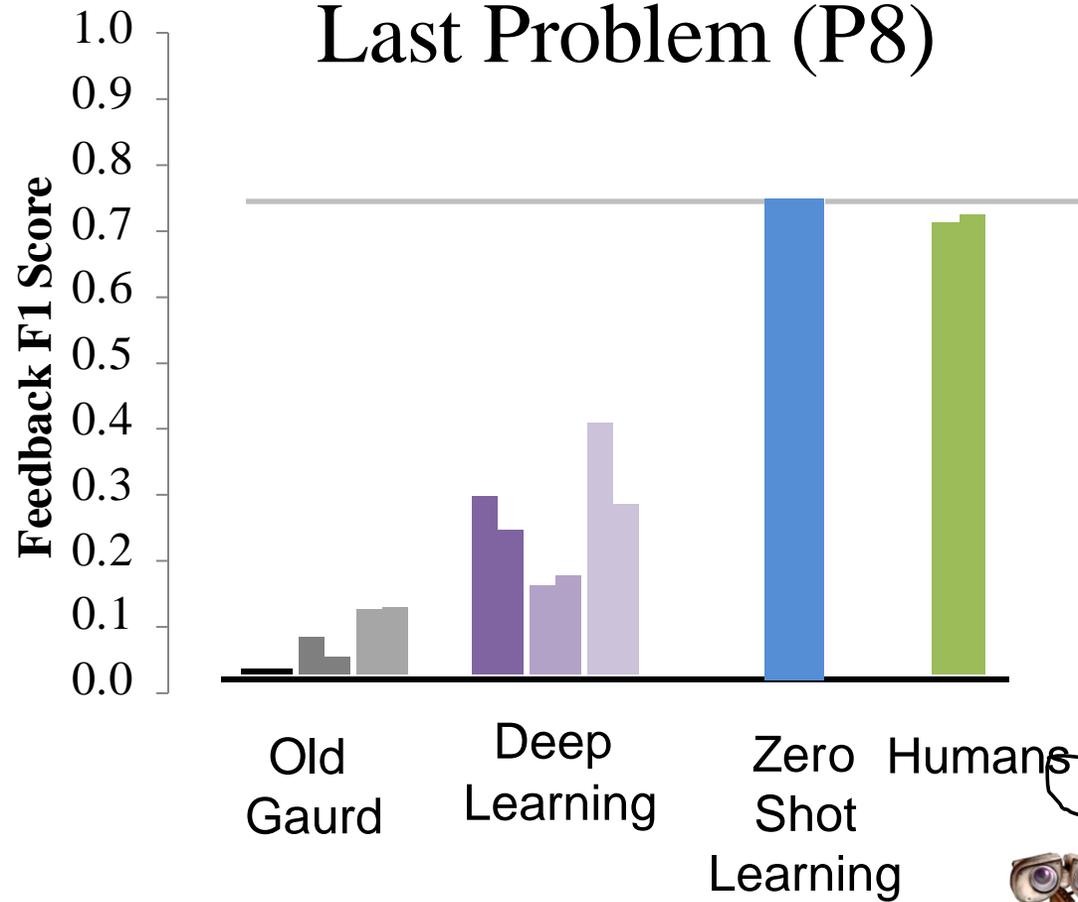


# Bayesian Program Learning

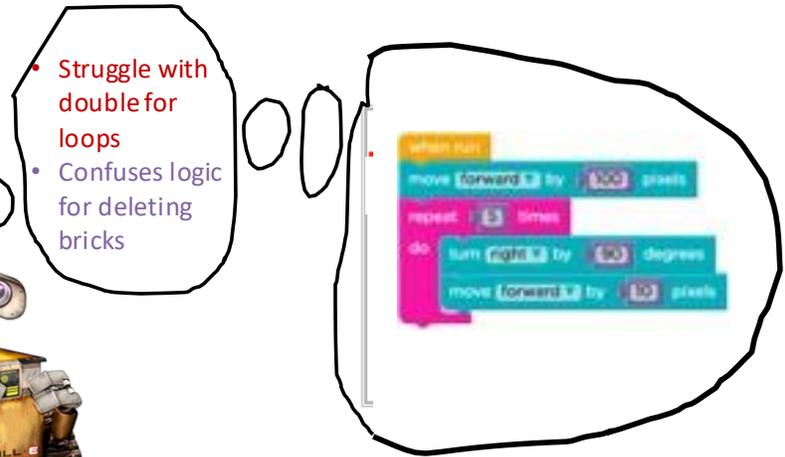


# Generative Understanding

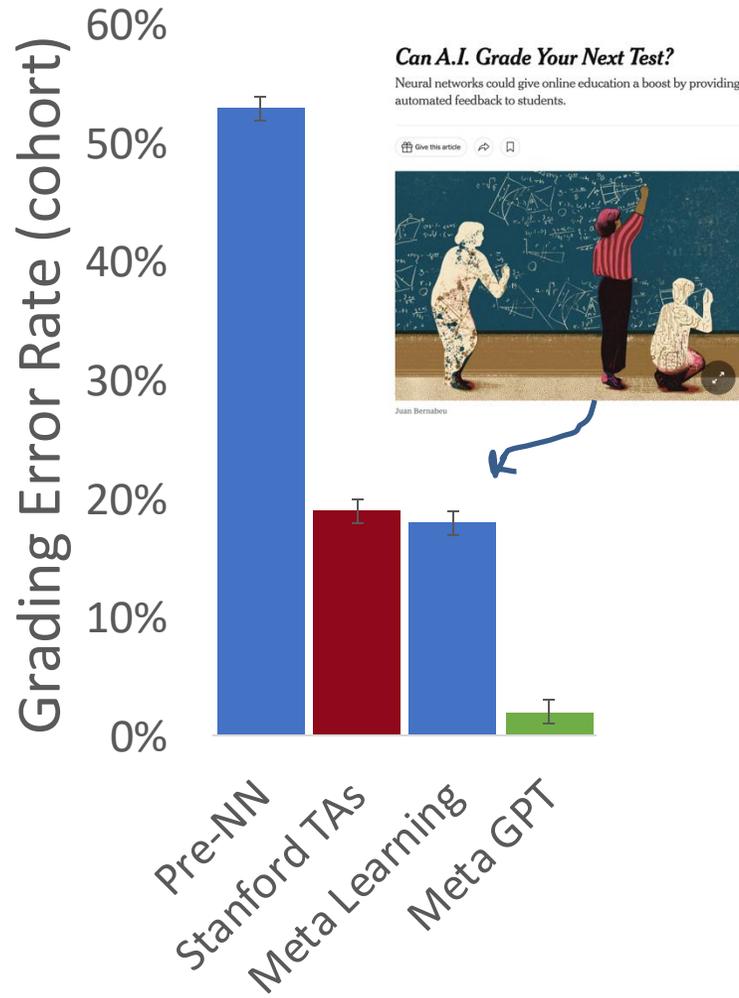
Label student code



*Outstanding Student  
paper award, AAAI 2019*



# Rubric Level Accuracy on Few-Shot Grading a Novel Question



Now lives in code.org's teaching assistant



# Stanford Code in Place:



**5000+** section leaders teach

**50,000+** students

**CS106A**

**As Community Service**

Featured in



**SCIENTIFIC  
AMERICAN**



# AI Realtime Feedback

**Style Feedback**

Once you solve the problem we will give you style feedback. Note that you can only request style feedback once every 10 minutes.

[Get Style Feedback](#)

**Style Best Practices**

- Choose meaningful names for functions and variables.
- Use constants for values that don't change.

```

1 import graphics
2 import time
3 import random
4 import math
5
6 canvas_width = 500
7 canvas_height = 600
8 paddle_y = canvas_height - 30
9 paddle_width = 80
10 paddle_height = 15
11 ball_radius = 10
12
13 brick_gap = 5
14 brick_width = (canvas_width - brick_gap * 9) / 10
15 brick_height = 10
16
17
18
19 def main():
20     canvas = graphics.create_canvas(canvas_width,
21     canvas_height)
22     ball = create_ball(canvas)
23     paddle = create_paddle(canvas)
24     create_bricks(canvas)
25     play_game(canvas, ball, paddle)
26
27 def play_game(canvas, ball, paddle):
28
29
30     n_bricks = 100
31     for i in range(3):
32         canvas.moveto(ball, canvas_width/2,
33         canvas_height/2)

```

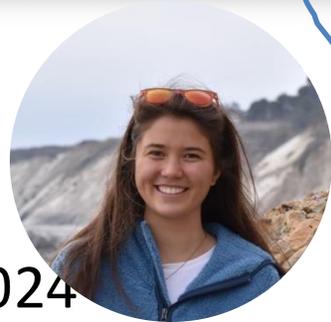
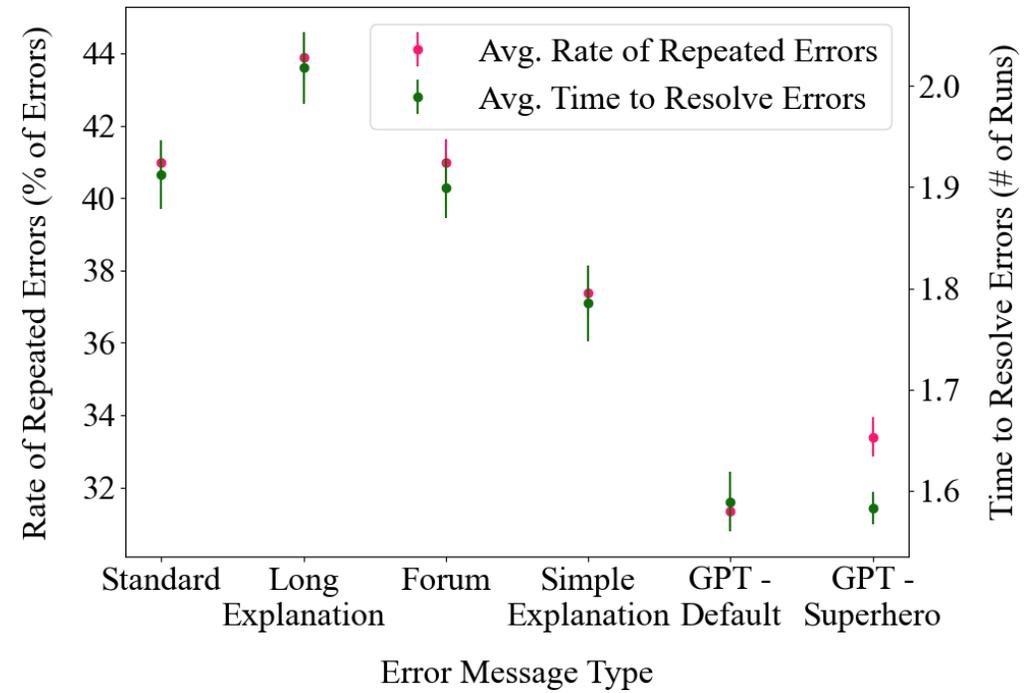
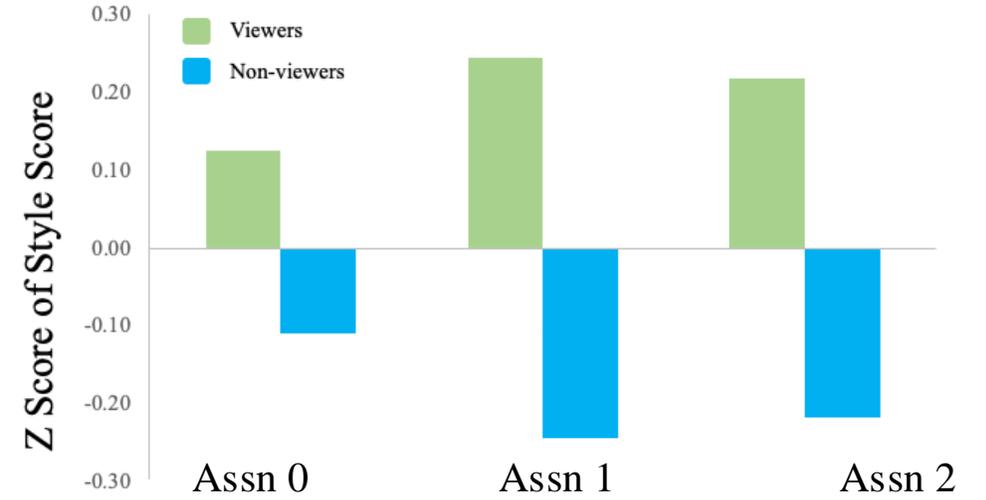
**Terminal** Error message type: Language Model

```

KeyboardInterrupt
% python main.py
Your code had an error so we are getting a message from a service called GPT. It might take a second, thank you for being patient!

(Line 19) SyntaxError: invalid syntax
This error occurs because there is a missing colon at the end of the main() function definition. The colon is necessary to indicate the start of the function body.

```



Both Accepted: SIGCSE 2024

# Grading Creative Projects



The screenshot shows a web browser window with the address bar displaying "DreamApp Grading Interface" and the URL "iris-ws-6:3000/grading\_display#". The page title is "demo\_student\_2" with a "Not Graded" status. The assignment is "Breakout".

Task	Score
Mouse movement	4/4
Brick drawing	5/5
Paddle drawing	4/4
Ball drawing	4/4
Constants	0/2
Wall bouncing	3/3
Paddle bouncing	2/2
<b>Paddle skewering</b>	<b>0/1</b>
Ball does not become skewered on paddle (1)	
New life	4/4
<b>Total Score</b>	<b>37/40</b>

On the right side of the interface, there is a video player showing a colorful grid of bricks (red, orange, yellow, green, blue) and a black ball. Below the video are buttons for "Video", "Code", and "Demo".

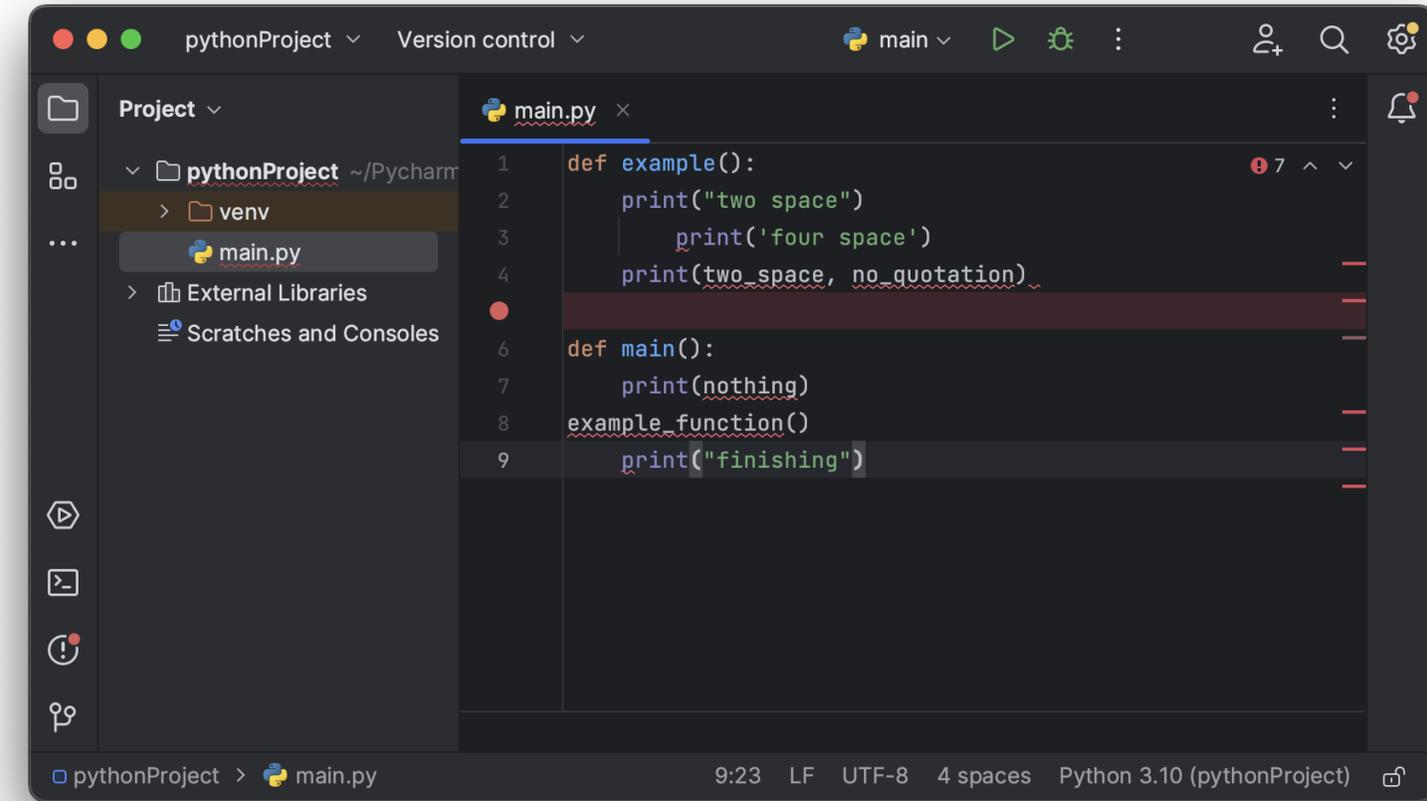
More than Education

```
def example():  
    → print("two space")  
    → → print('four space')  
    → print(two space, no quotation)
```

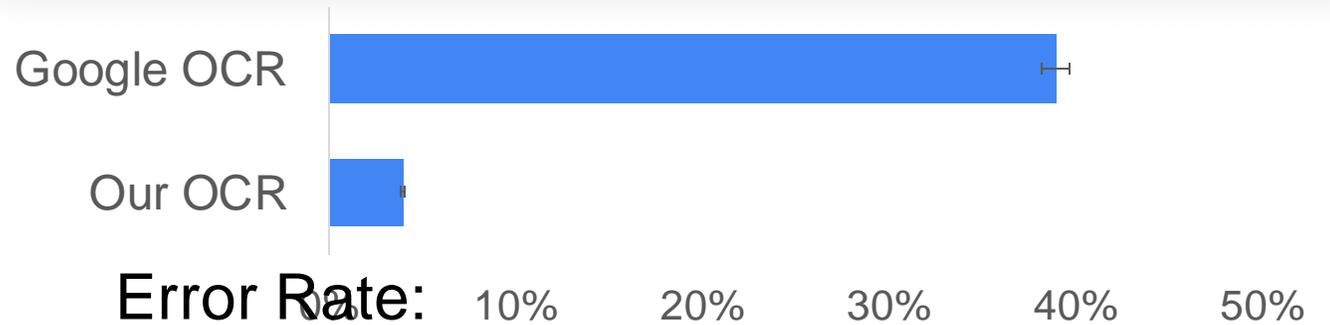
```
def main():  
    → print(rinting)
```

```
example_function()  
    → print("finishing")
```

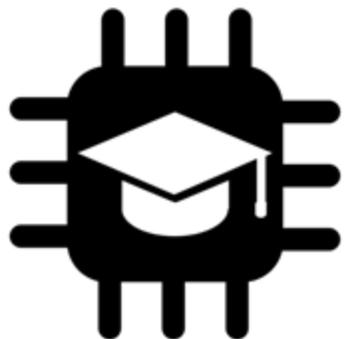
```
if __name__ == "__main__":  
    main()
```



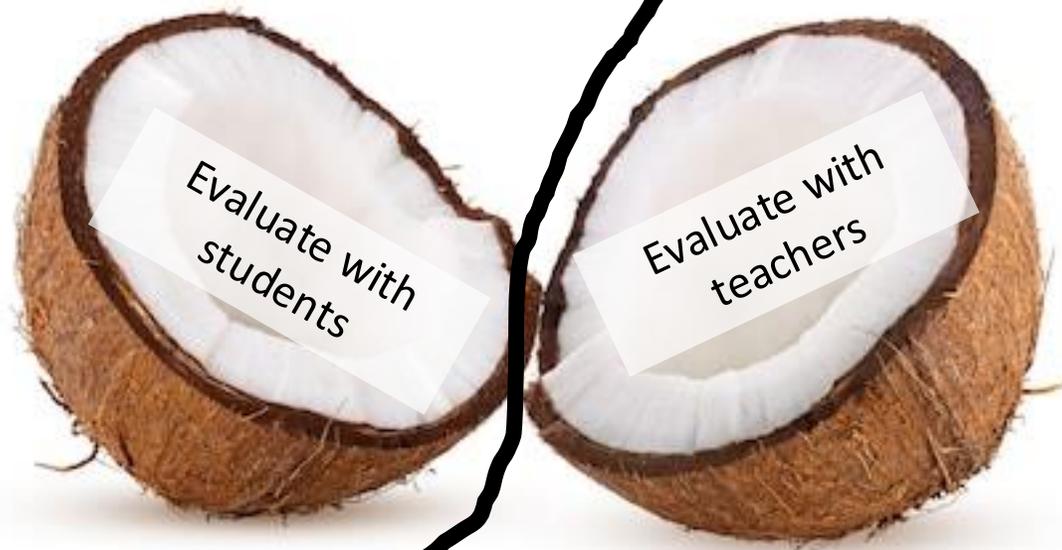
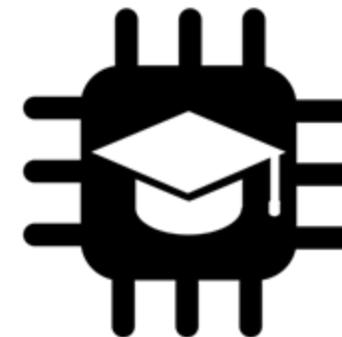
```
1 def example():  
2     print("two space")  
3     print('four space')  
4     print(two_space, no_quotation)  
6 def main():  
7     print(nothing)  
8     example_function()  
9     print("finishing")
```



**Auto Teacher:** I want to build AI to augment **student learning**



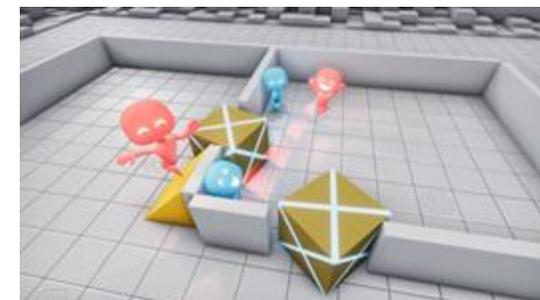
**Auto Student:** I want to build AI to augment **teacher training**



## The Education Dual

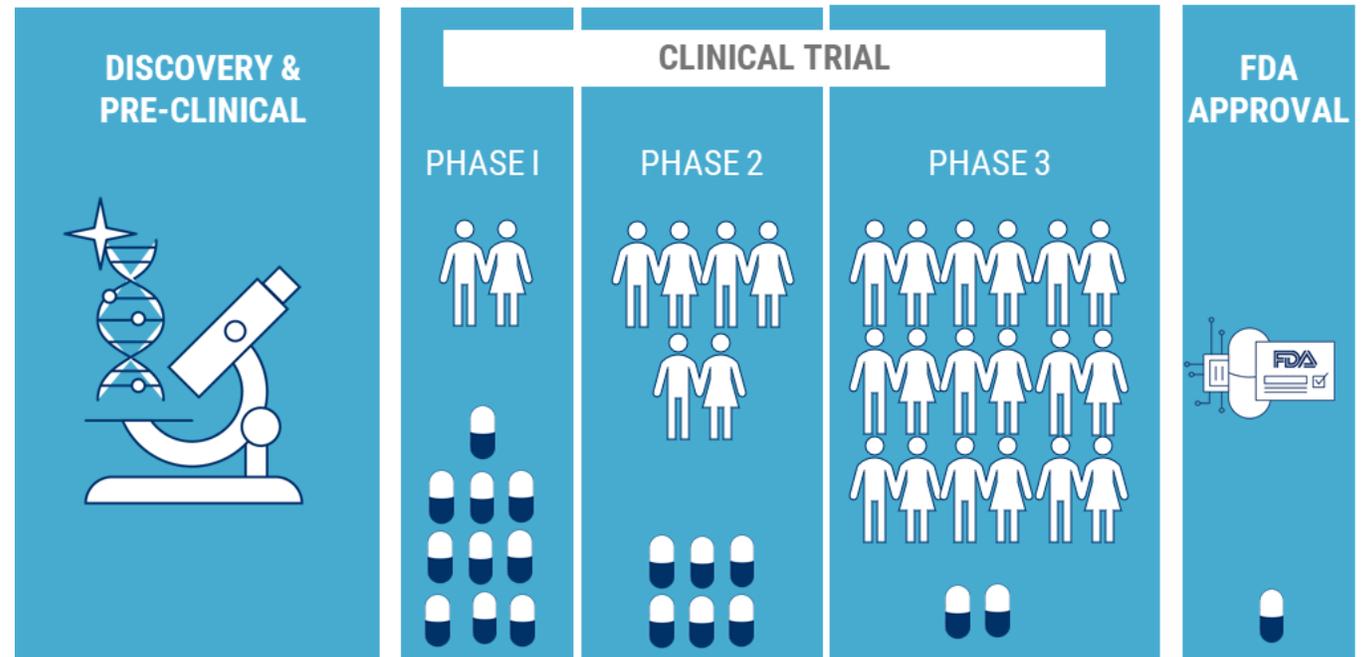
Best Paper Award, EDM 2022

**The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues**



# More than education

 Bringing a drug to market is a drawn-out process



Source: cbinsights.com

 CBINSIGHTS



# Chose k examples from a dataset of lots of student work

The screenshot shows a web browser window with the URL `cs109psets.netlify.app/fall22/pset1/randomchoice`. The page title is "PS1 numpy.random.choice". The main content area contains the following text:

How many unique sets could be returned by the following `main()` function:

```
import numpy as np

def main():
    options = [0,1,2,3,4,5,6,7,8,9]
    ordered = np.random.choice(options,
                               size = 4, replace=False)
    unordered = set(ordered)
    return unordered
```

Below the code, there is a link to the documentation for `numpy.random.choice` and a note: "Note that sets are considered unordered. If two sets have the same elements, they are not distinct."

On the right side of the interface, there is an "Answer Editor" section with a "Solution" tab. The "Numeric Answer" is 4. The "Explanation" is "asdfasdf!!!! Y = 5". Below the explanation, there is a code editor with the following code:

```
1 print('test')
```

Below the code editor, there is a "Run" button and a "Show" button. The output of the code is "abcdef" and "Y = 25".

At the bottom of the page, there are "Previous Question" and "Next Question" buttons.

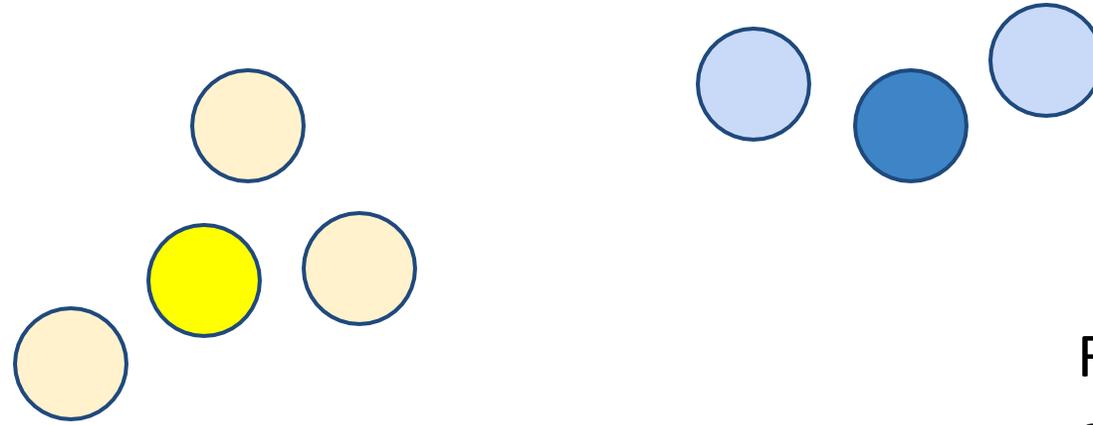
300 solutions

Find the 10 solutions which are most representative



# K Medoids: A Classic Algorithm

Choose the  $k$  nodes such that the sum of minimized distances is as small as possible



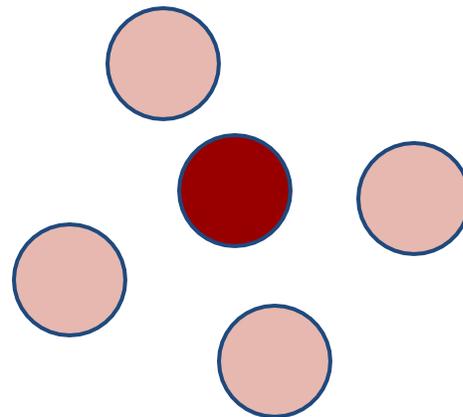
For any two points you can query the distance between them. May be non-Euclidean

Before

After

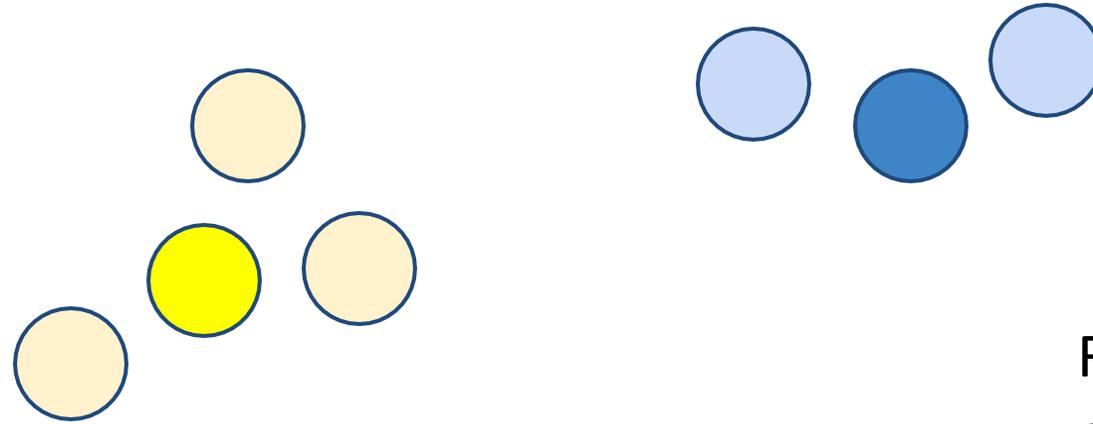
$$O(n^2)$$

$$O(n \log n)$$



# Step 1: Chose the node closest to the rest

Chose the k nodes such that the sum of minimized distances is as small as possible



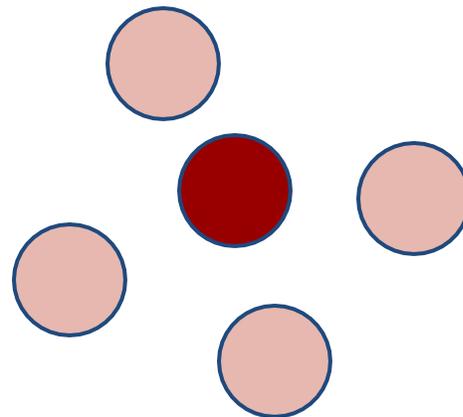
For any two points you can query the distance between them. May be non-Euclidean

Before

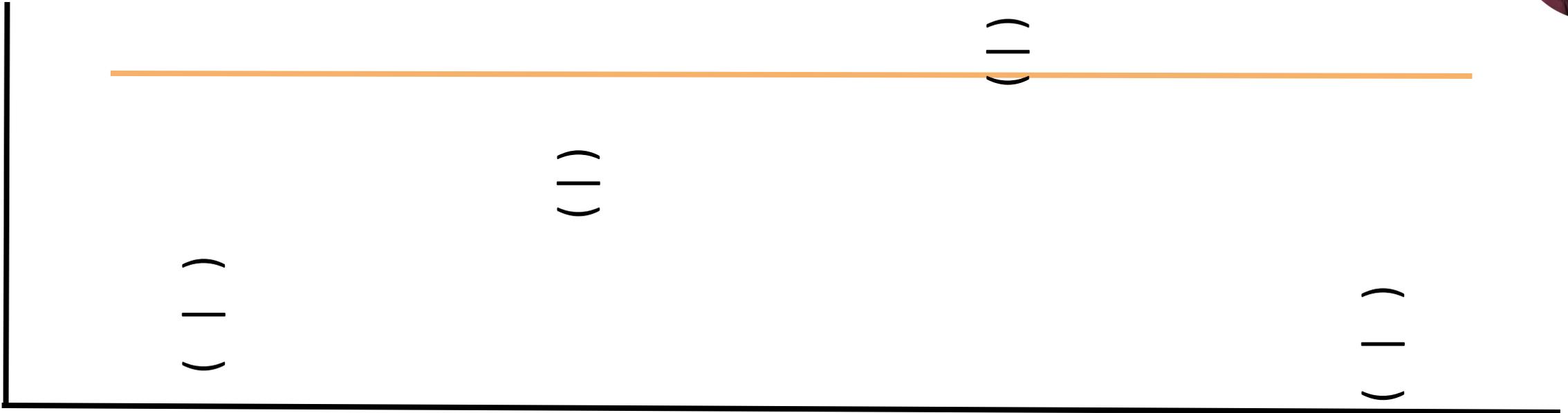
After

$$O(n^2)$$

$$O(n \log n)$$



# The inner loop can be thought of as Thompson Sampling



3, 4



10, 11, 9, 8



12, 11, 10, 14



2, 0



# Allowed us to revisit several core algorithms

- $k$ -Medoids
  - *BanditPAM: Almost Linear Time  $k$ -medoids Clustering via Multi-Armed Bandits*", NeurIPS 2020
- Random Forests
  - *"MABSplit: Faster Forest Training Using Multi- Armed Bandits"*, NeurIPS 2022
- Maximum Inner Product Search
  - *"Faster Inner Product Search in High Dimensions"*, Submitted to 2023

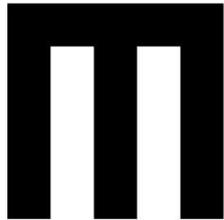


# More than education

Vision Test

myeyes.ai/measure

## Left Eye



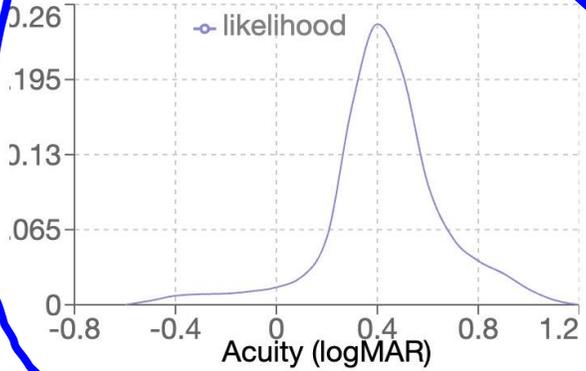
Featured in  
**THE LANCET**

Progress: 10%

## StAT Algorithm

N done: 2  
Curr size: 3.3 arcmin  
Curr size: 0.5 logMAR  
MAP acuity: 2.5 arcmin  
MAP acuity: 0.4 logMAR  
Interval: [1.0, 12.0] arcmins

Likelihood of Acuity Scores:



Acuity (logMAR)	Likelihood
-0.8	0.00
-0.4	0.00
0.0	0.01
0.4	0.26
0.8	0.05
1.2	0.00



What else should be a **random variable**?

Chris 2017: Ability to See??

What else should be a **random variable**?

Chris 2023: Grades??

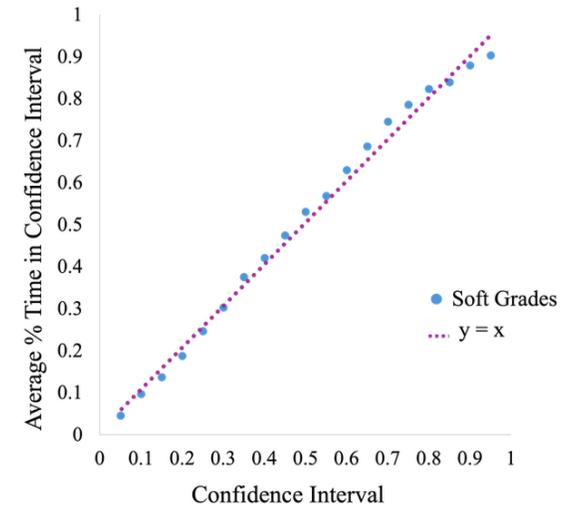
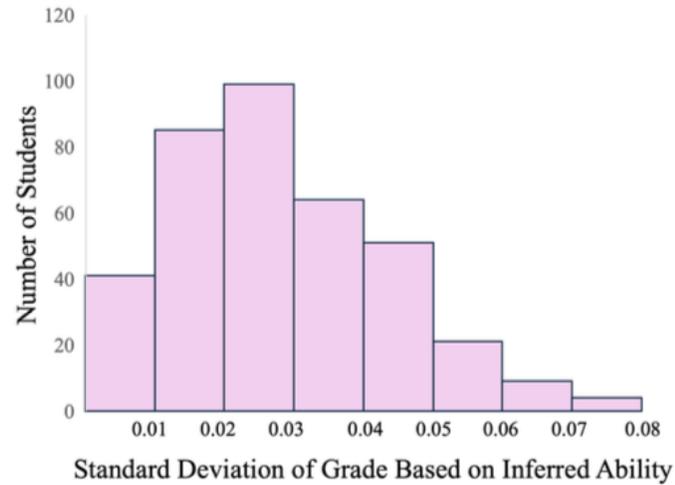
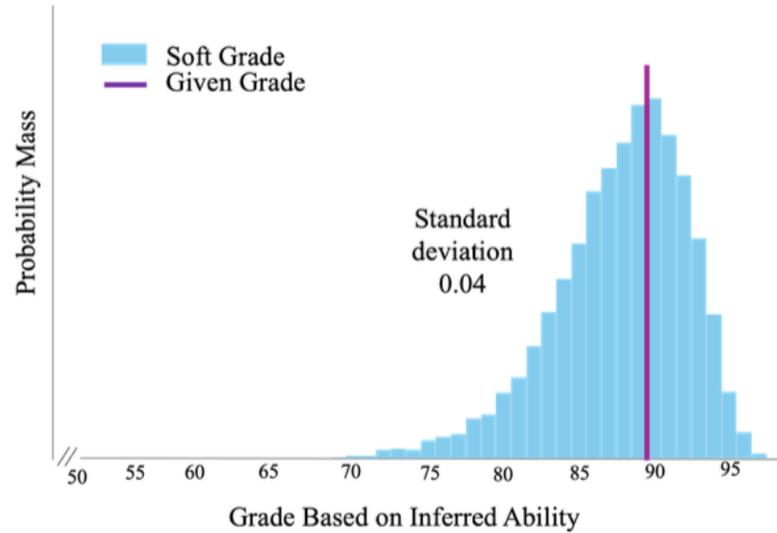
What else should be a **random variable**?

Juliette 2024: Grades!



TA for CS109, Fall 2023

# Soft Grades. Accepted for publication in 2025



## State of the Art imputation of grades, among other uses

	C1	C2	OULAD-1	OULAD-2	OULAD-3	OULAD-4	OULAD-5	OULAD-6	OULAD-7
<b>Soft Grades (RMSE)</b>	<b>0.042</b>	<b>0.058</b>	<b>0.069</b>	<b>0.067</b>	<b>0.062</b>	<b>0.089</b>	<b>0.071</b>	<b>0.046</b>	<b>0.076</b>
CRM Baseline (RMSE)	0.060	0.264	0.079	0.221	0.071	0.145	0.080	0.401	0.349

Table 3. Root Mean Square Error (RMSE) comparison between Soft Grades and CRM Baseline across all courses. Lower RMSE is better.

What else should be a **random variable**?

PSet Timing???

Travel Timing???

Stock Prices???

(You could do better!)

# Application -> Theory

Understand social science,  
especially with small data

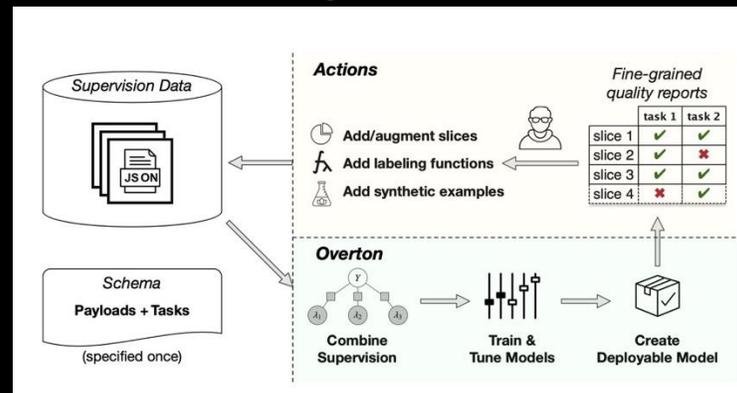
Explain why it made the  
choices it did

What are things that AI  
currently can't do?

Teach humans based on  
what it has learned

Prove it is correct /  
aligned with human  
values

Design itself



What should you do  
next?

Go solve amongst the abundance of important problems



Final Project | AA228/CS238

web.stanford.edu/clas...

Stanford University

# AA228/CS238

Decision Making under Uncertainty

MENU

## Final Project

The objective of the final project is to explore topics in decision making under uncertainty in greater depth than is permitted in class. The choice of topic is up to you, but it should be related to the general themes of the course. As part of the project you should:

- *describe* an approach (existing or newly developed),
- *apply* the approach to a problem of interest (which may or may not be related to aerospace), and
- *analyze* the performance of the approach according to a set of metrics.

CS221: Artificial Intelligence: P

stanford-cs221.github...



# CS221: Artificial Intelligence: Principles and Techniques

Stanford / Autumn 2022-2023

[\[Calendar\]](#) [\[Modules\]](#) [\[Coursework\]](#) [\[Schedule\]](#)

- Lectures: Mon/Wed 1:30-2:50pm in NVIDIA Auditorium.
- Problem sessions: Fridays 1:30-2:20pm in Huang 018.
- Office hours, homework parties: see the [Calendar](#).
- To contact the teaching staff, please use Ed; for more personal/sensitive matters, email [cs221-aut22-23-lead-staff@lists.stanford.edu](mailto:cs221-aut22-23-lead-staff@lists.stanford.edu).

## Teaching Staff



Percy Liang  
Instructor



Dorsa Sadigh  
Instructor

CS229: Machine Learning

cs229.stanford.edu

CS229

# CS229: Machine Learning

## Instructors



Andrew Ng



Moses Charikar



Carlos Guestrin

**Course Description** This course provides a broad introduction to machine learning and statistical pattern recognition. Topics include: supervised learning (generative/discriminative learning, parametric/non-parametric learning, neural networks, support vector machines); unsupervised learning (clustering, dimensionality reduction, kernel methods); learning theory

CS 228 - Probabilistic Graphical Models

ermongroup.github.io/...

# CS 228 - Probabilistic Graphical Models

Winter 2021-22

[Ed](#)
[Calendar](#)
[Course Notes](#)

[Logistics](#) | [Course Info](#) | [Syllabus](#) | [Other Resources](#)

## Logistics

- **Lectures:** Tue, Thu, 9:45am-11:15am, Nvidia Auditorium
- **Office Hours and Sections:** [Google Calendar](#)

Statistics 200: Introduction to

web.stanford.edu/clas...

# Statistics 200: Introduction to Statistical Inference

Zhou Fan, Stanford University, Autumn 2016

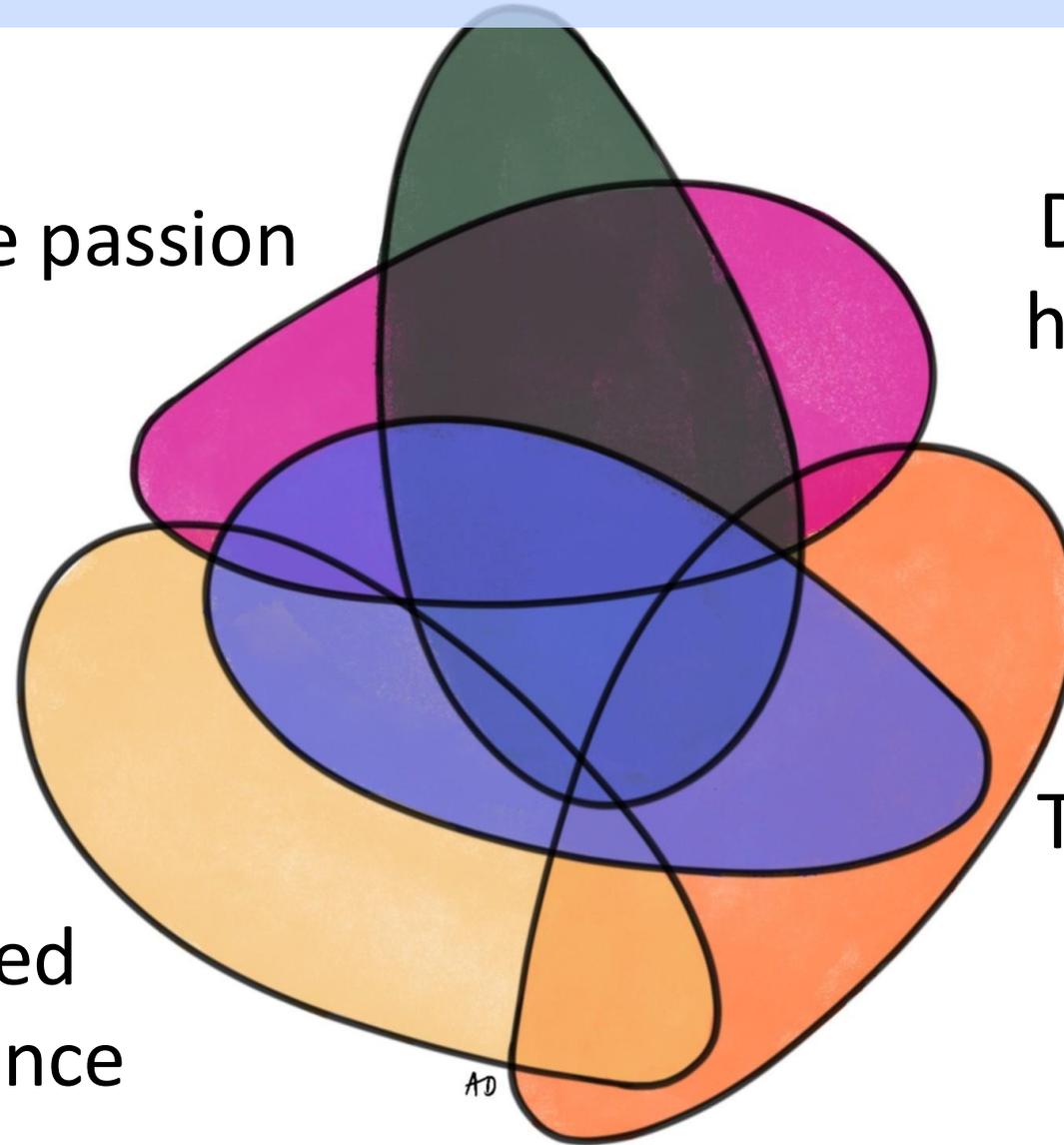
- Home
- Lectures
- Homework
- Grades
- Piazza



# Think about intersectionality

Your side passion

Data that you  
have access to

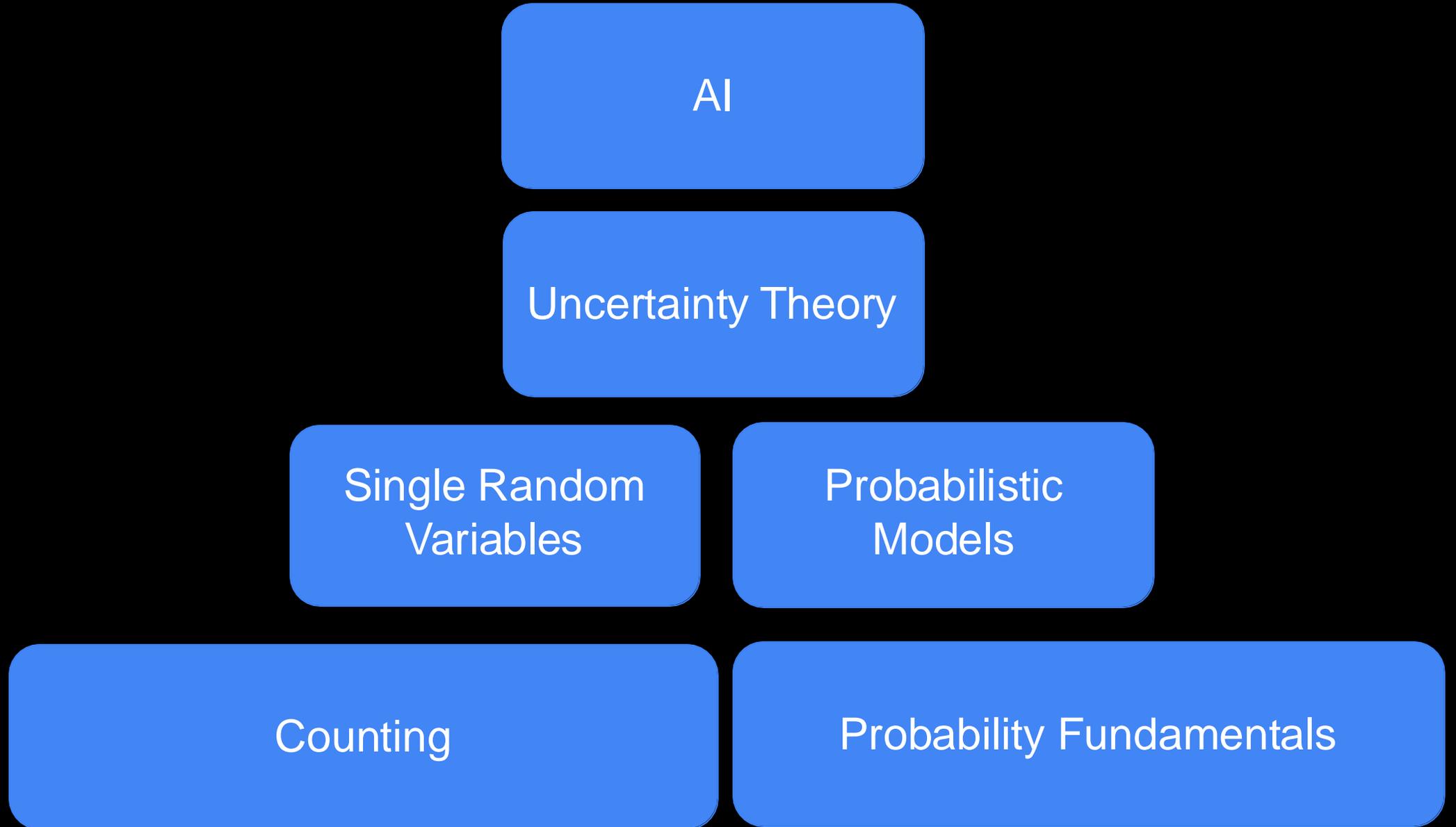


Thompson  
sampling

Your lived  
experience

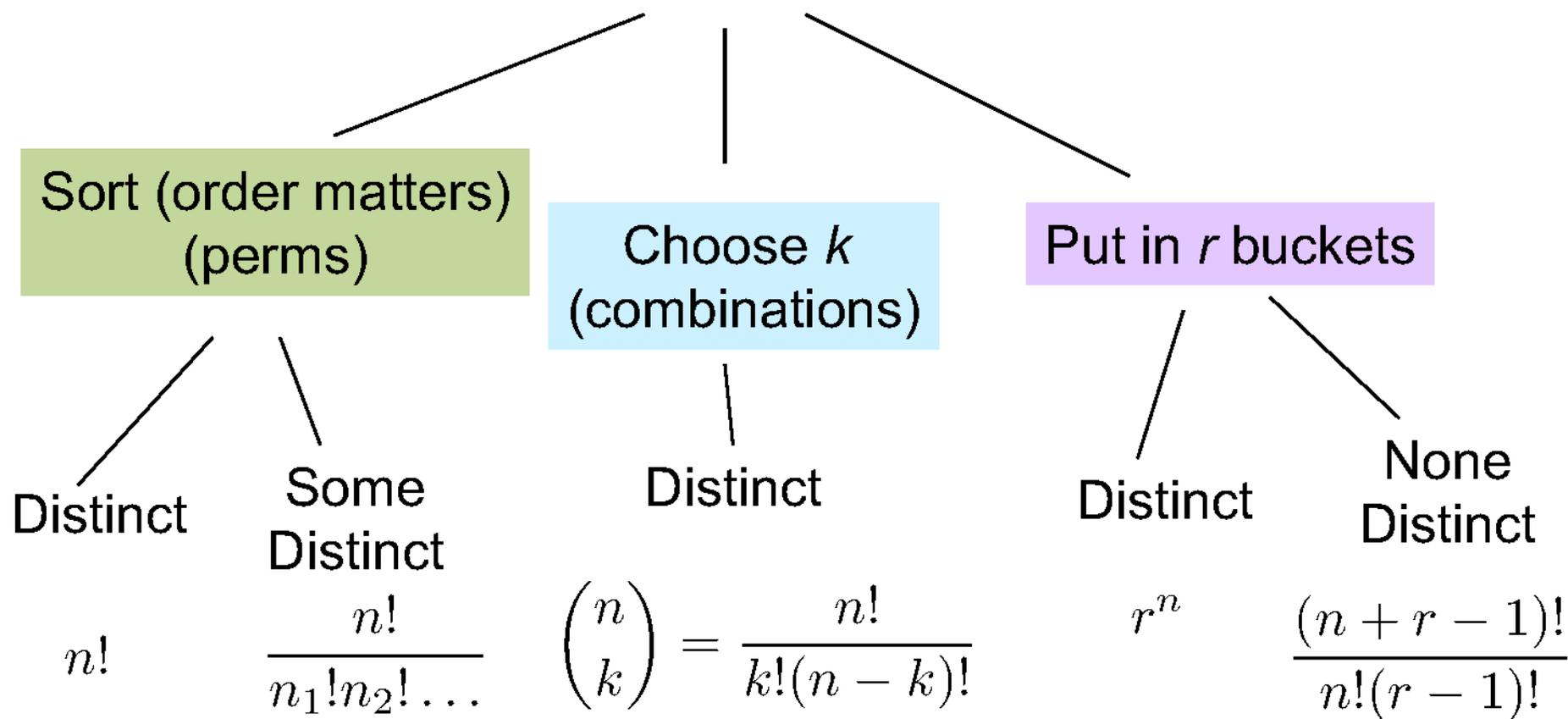


Last Class...



# Counting Rules

Counting operations on  $n$  objects



# Counting



Ayesha



Tim



Irina



Joey



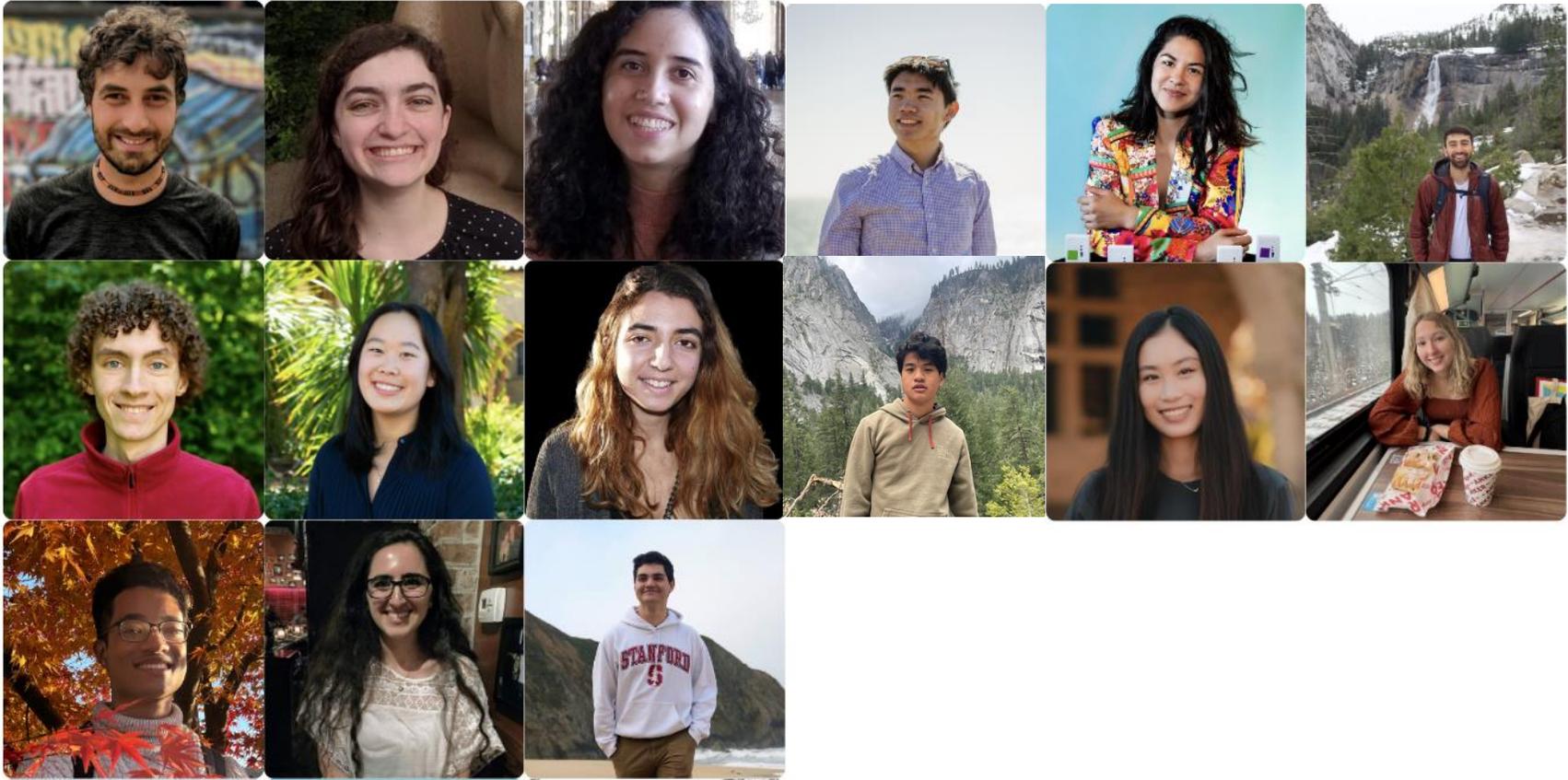
Waddie



## COURSE VALUES

Everyone is welcome.  
Intellectual joy. Be kind. Be humane. Social connection.  
Learn by doing. Thrill of building. Adapt to new contexts.

# Fantastic Teaching Team





**WHEN YOU MEET YOUR BEST FRIEND**

Somewhere you didn't expect to.



Trailing the dovetail shuffle to it's lair – Persi Diaconosis

# What is a Probability?

$$P(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n}$$



# What is a Probability

The screenshot shows a web browser window with the URL `cs109psets.netlify.app/fall24/pset1/sum100`. The page title is "Sum 20". On the left, a sidebar shows a list of questions, with question 13 selected. The main content area contains the following text:

Consider a game, which uses a random number generator that produces independent random integers between 1 and 5, inclusive. The game starts with a sum  $S = 0$ . The first player adds random numbers from the generator to  $S$  until  $S > 10$ , at which point they record their last random number  $X$ . The second player continues by adding random numbers from the generator to  $S$  until  $S > 20$ , at which point they record their last random number  $Y$ . The player with the highest number wins; e.g., if  $Y > X$ , the second player wins. Write a Python 3 program to simulate 100,000 games and output the estimated probability that the second player wins. Include your answer along with code used to compute it. Give your answer rounded to 3 places behind the decimal.

Here is an example run of the game. In this run player 1 has score 4 and player 2 has a score of 3 so player 1 wins:

```
Round 1
randint = 3, sum = 3
randint = 5, sum = 8
randint = 4, sum = 12
Round 1 over: Player 1 score is 4

Round 2
randint = 1, sum = 13
randint = 5, sum = 18
randint = 3, sum = 21
Round 2 over: Player 2 score is 3
```

You might find it helpful to use the python function `random.randint(min_value, max_value)` which returns a random integer in the range `min_value` to `max_value` inclusive. So for example this code will produce one of the integers `[1,2,3]`:

```
import random
my_num = random.randint(1,3)
print(my_num)
```

At the bottom of the page, there are buttons for "Previous Question" and "Next Question".

The right side of the browser shows an "Answer Editor" with a "Solution" tab selected. The "Answer editing" toggle is set to "Off". The code editor contains the following Python code:

```
Code:
1 import random
2
3 n_trials = 100000
4
5 def main():
6     n_wins = 0
7     print(f"Running {n_trials} trials")
8     for i in range(n_trials):
9         player_2_wins = run_trial()
10        if player_2_wins:
11            n_wins += 1
12        print(n_wins / n_trials)
13
14 def run_trial():
15     a = 0
16     b = 0
17     s = 0
18     while s <= 10:
19         a = random.randint(1, 5)
20         s += a
21     while s <= 20:
22         b = random.randint(1, 5)
```

Below the code editor is a "Run" button. The output of the code is displayed in a box below the button:

```
Running 100000 trials
0.38563
```

# Netflix and Learn

$$P(E|F) = \frac{P(EF)}{P(F)}$$

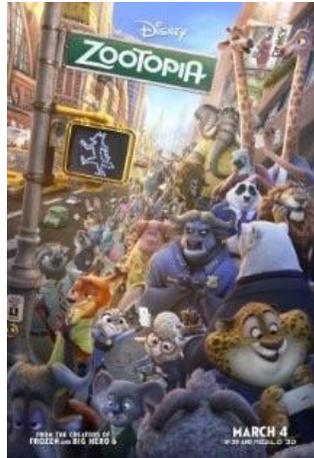
Definition of  
Cond. Probability

- Let  $E$  be the event that a user watches the given movie.
- Let  $F$  be the event that the same user watches CODA (2021).



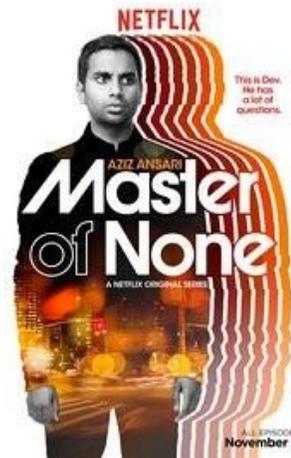
$$P(E) = 0.19$$

$$P(E|F) = 0.14$$



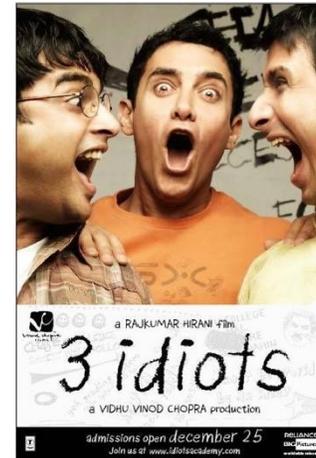
$$P(E) = 0.32$$

$$P(E|F) = 0.35$$



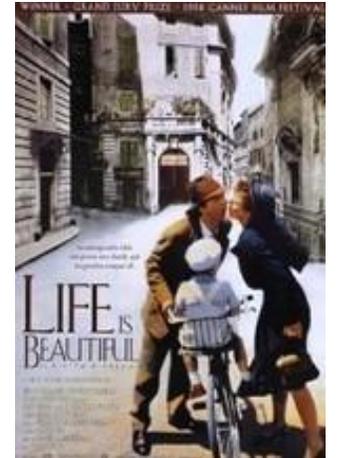
$$P(E) = 0.20$$

$$P(E|F) = 0.20$$



$$P(E) = 0.09$$

$$P(E|F) = 0.72$$



$$P(E) = 0.20$$

$$P(E|F) = 0.42$$

# Montey Hall Problem



Marilyn discovers the  
Probability Bug

# Zika Test



Positive Zika.

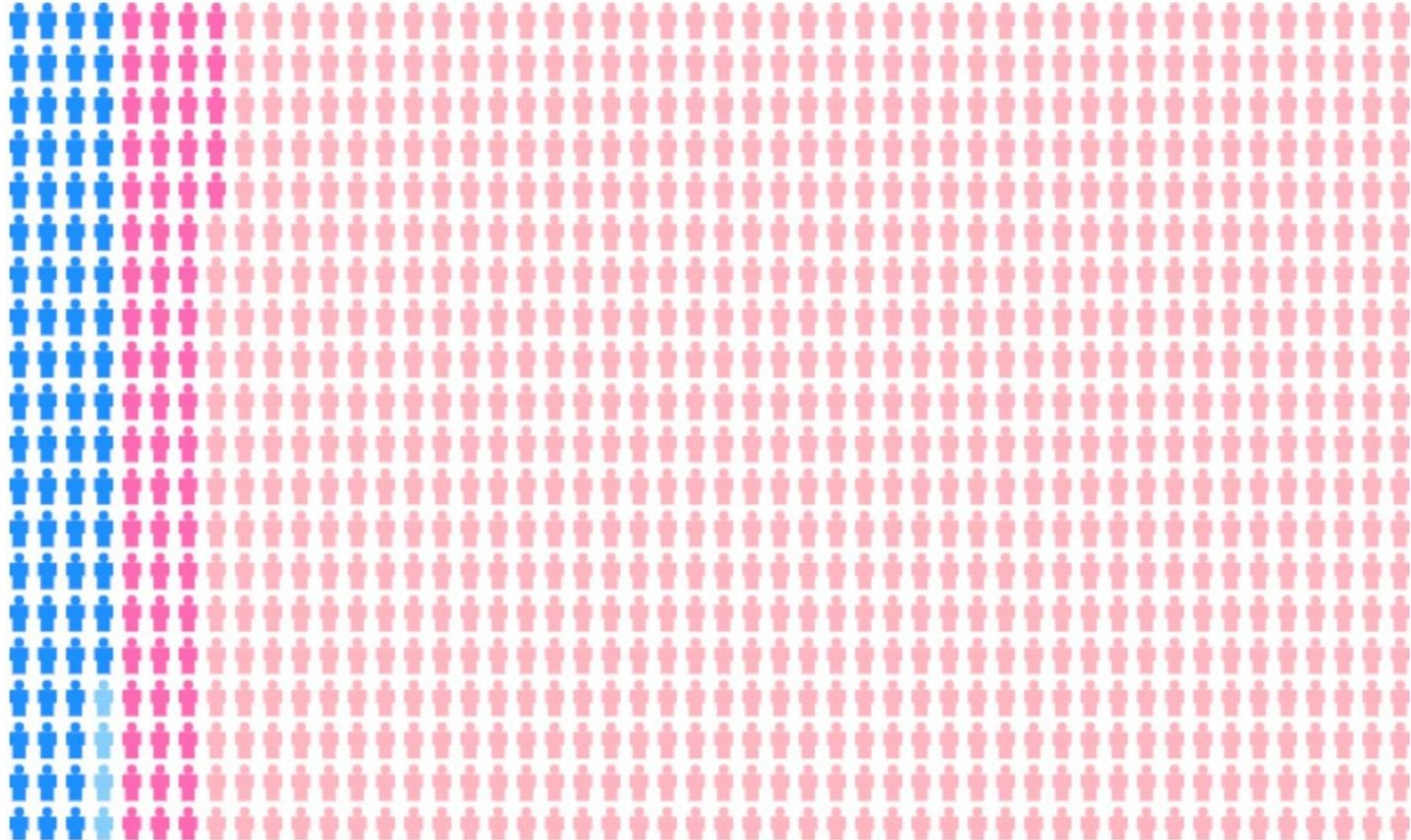
*What is the probability of zika?*

- 
- *0.1% of people have zika*
  - *90% positive rate for people with zika*
  - *7% positive rate for people without zika*

The right answer is 1%



# Bayes Theorem Intuition



# Program the General Version

cs109psets.netlify.app/fall24/pset2/medical\_diagnosis

## Medical Test

Write a function:

```
def predict_positive_given_test_result(
    prior_disease,
    p_true_given_disease,
    p_true_given_no_disease,
    test_result):
```

That can be used for any noisy (binary) medical test, such as a Covid-19 test, or an Ebola test. Your function takes in a prior belief that a patient has a disease, statistics on a noisy test, and the test result from the noisy test. Based off this information, you should compute the probability that the patient is "positive" for the disease (in other words, they have the disease). Your return value must be a number between 0 and 1, not a boolean prediction. This problem requires you to code up a general implementation of Bayes' Theorem for a binary prediction!

Hint: you might find it helpful to read the medical example from the [Bayes Theorem](#) chapter.



Noisy Test: [Previous Question](#) [Next Question](#)

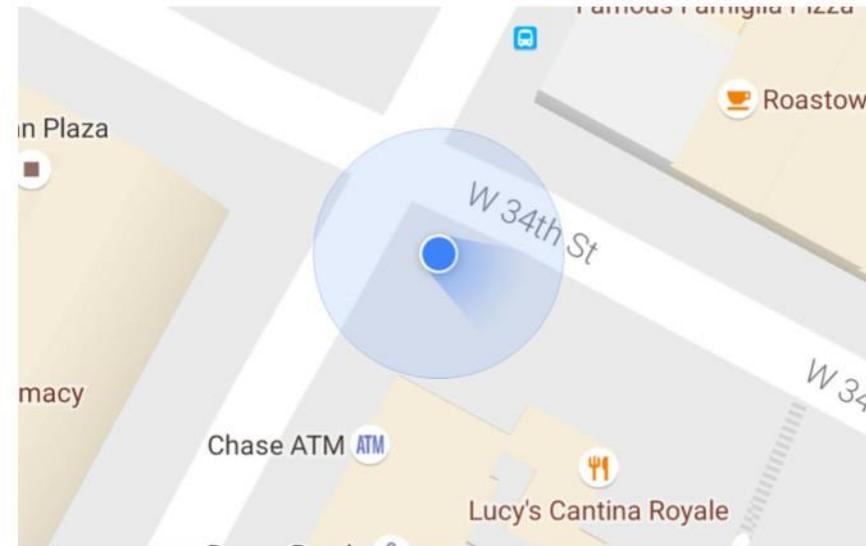
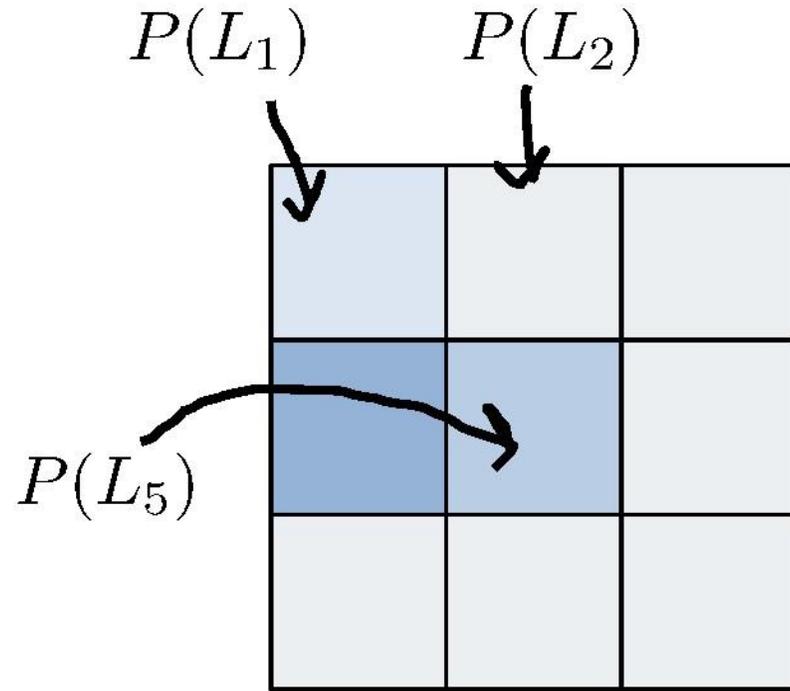
Answer Editor Solution

Agent:

```
1 def predict_positive_given_test_result(
2     prior_disease,          # prior prob that the patient has the disease
3     p_true_given_disease,  # the "true positive" probability
4     p_true_given_no_disease, # the "false positive" probability
5     test_result):         # True/False test result
6     # TODO: your code here
7     return 0.5
```

Run One Game Test Agent

# Update Belief



Before Observation



# Recall our Ebola Bats



# Fourth Year of Sections



I'm not a robot

  
reCAPTCHA  
[Privacy - Terms](#)

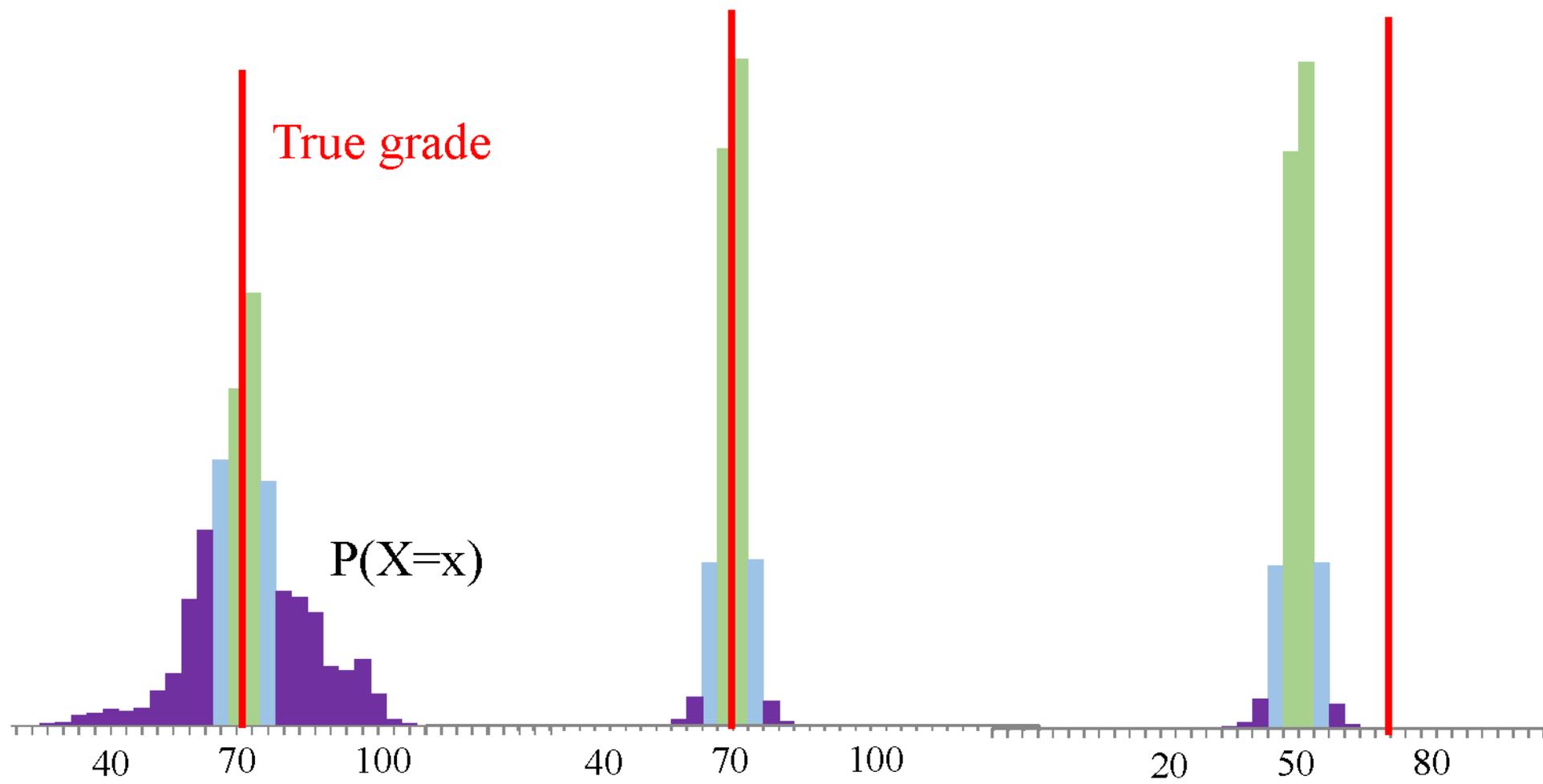


X		O
O	X	
		X



# Random Variables

X is the score a peer grader gives to an assignment submission

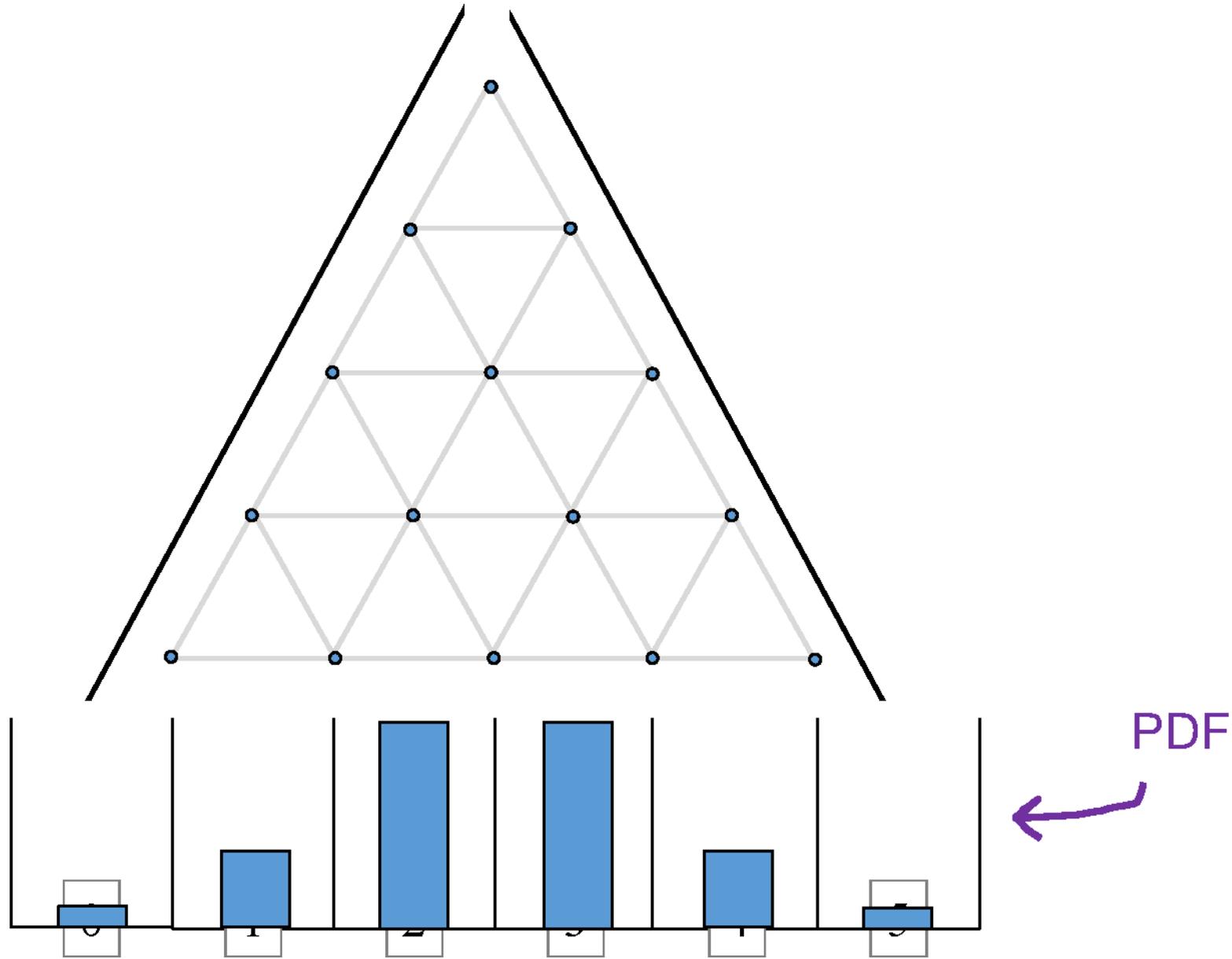


A

B

C

# Binomial

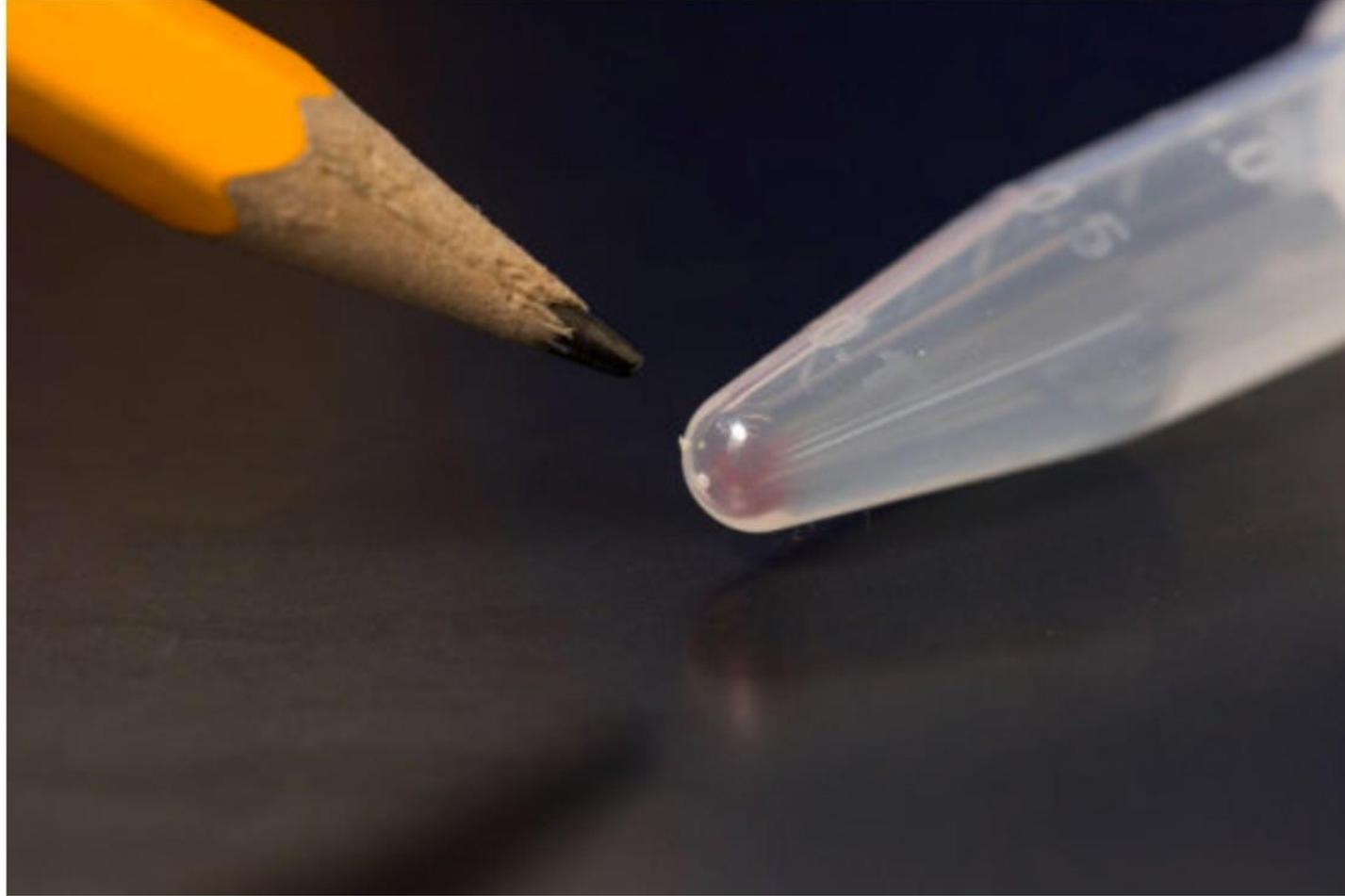


# Geometric

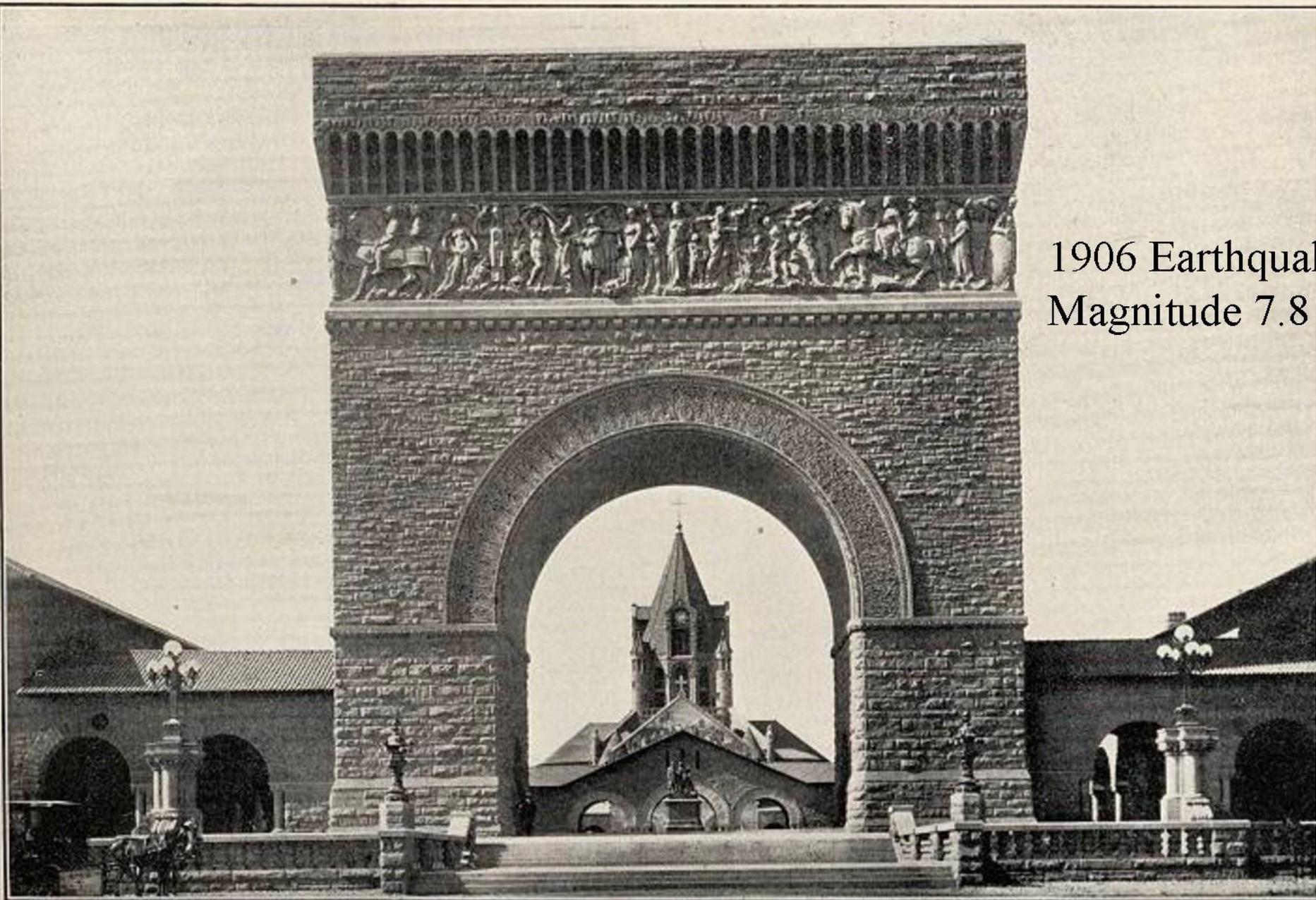
Sequence 1:

TTHTHTTHTTTHTTTHTTTHTTHTHTHT  
HTTHTTTHHTHTHTTHTTHTTHTTHTT  
HTHTHTHTHTTHTTHTTHTHTHTTHTTHT  
TTHTHTTHTHTHTHTHTHTHTHTHTHTHT  
TTHTHTHTHTHTHTTHTTHTTHTTHTHTHT

# Storing Data on DNA



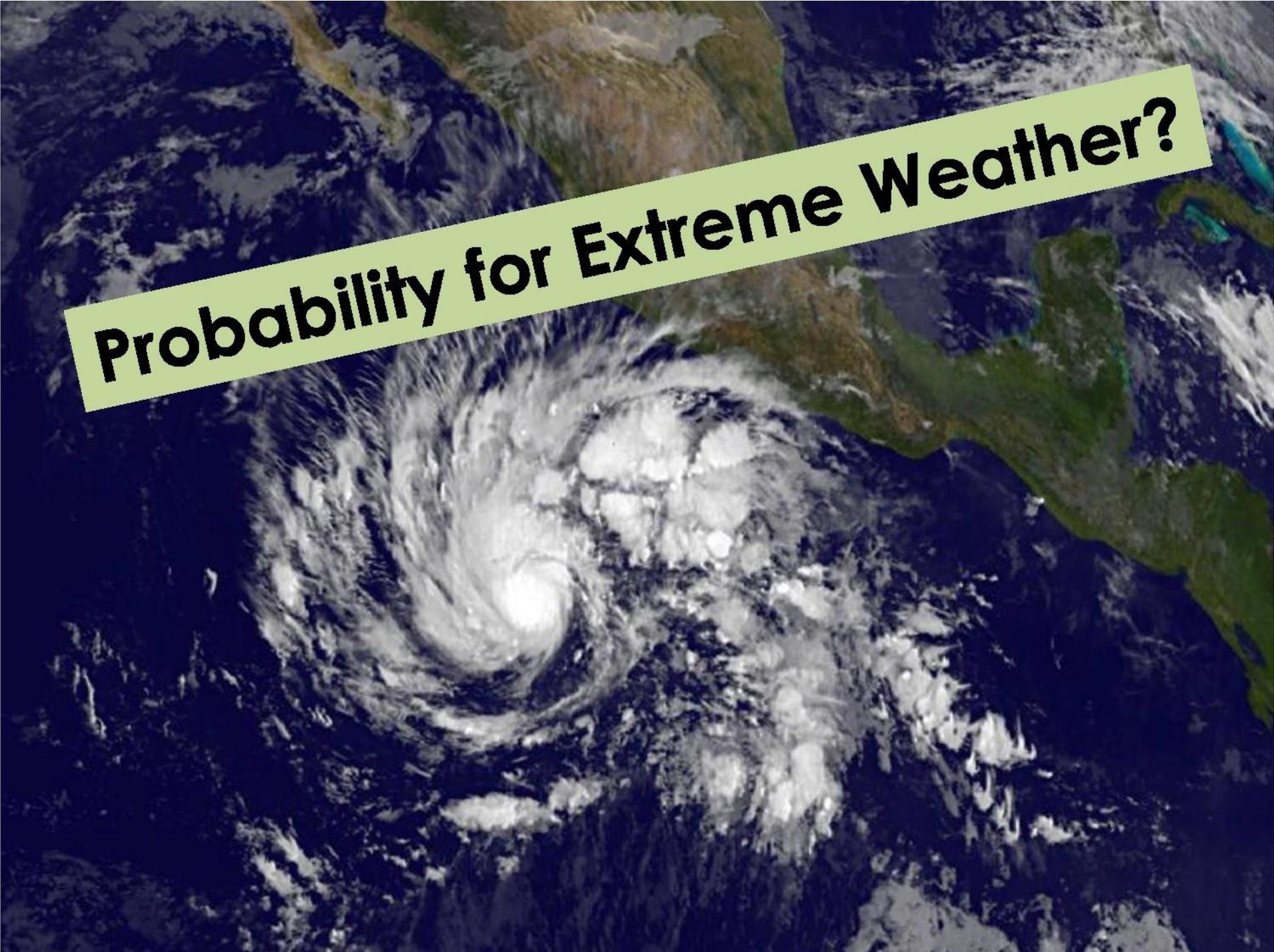
All the movies, images, emails and other digital data from more than 600 smartphones (10,000 gigabytes) can be stored in the faint pink smear of DNA at the end of this test tube.



1906 Earthquake  
Magnitude 7.8

ILL. No. 65. MEMORIAL ARCH, WITH CHURCH IN BACKGROUND, STANFORD UNIVERSITY, SHOWING TYPES OF CARVED WORK WITH THE SANDSTONE.

**Probability for Extreme Weather?**

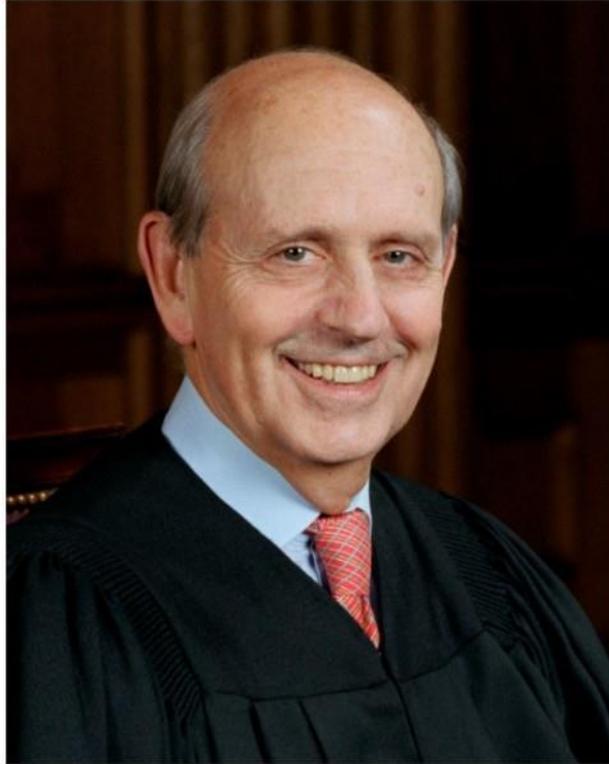


# Bit Coin Mining

You “mine a bitcoin” if, for given data  $D$ , you find a number  $N$  such that  $\text{Hash}(D, N)$  produces a string that starts with  $g$  zeroes.

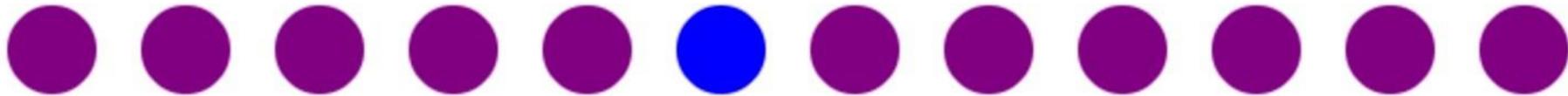


# Representative Juries



Simulate

Simulation:

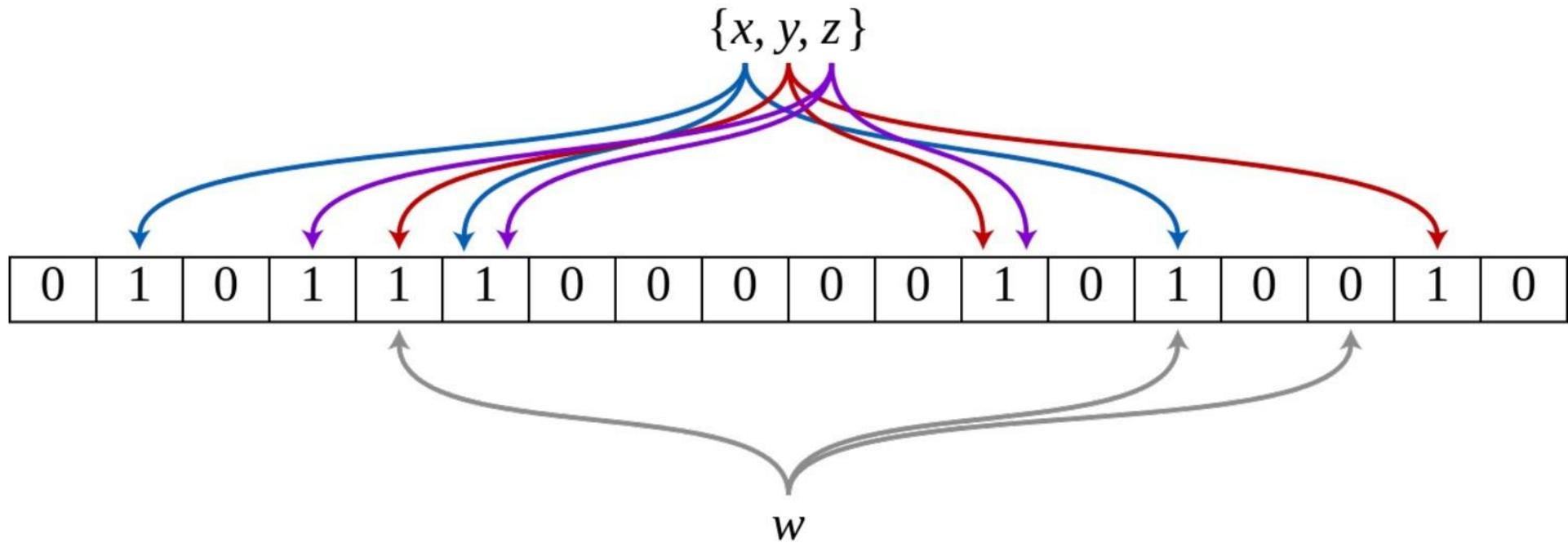


# Dating at Stanford

Each person you date has a 0.2 probability of being someone you spend your life with. What is the average number of people one will date? What is the standard deviation?



# Bloom Filter



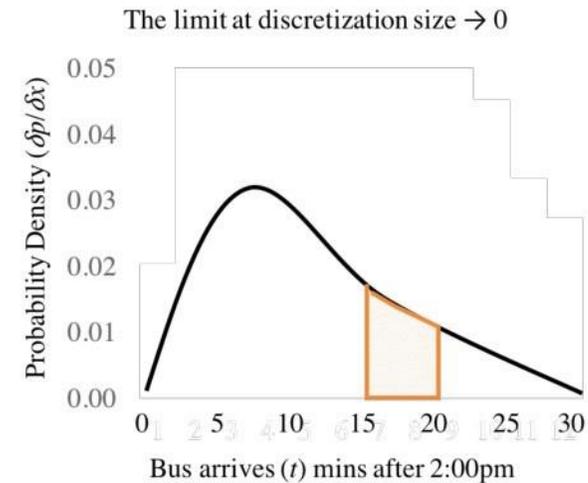
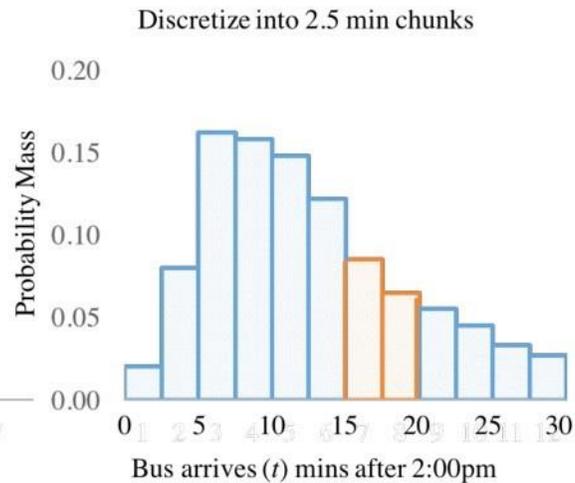
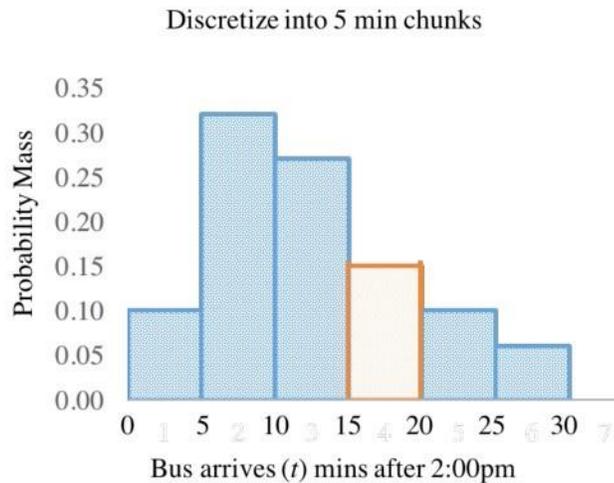
random( ) ?

# Riding the Marguerite



You are running to the bus stop.  
You don't know exactly when  
the bus arrives. You arrive at  
2:20pm.

What is  $P(\text{wait} < 5 \text{ min})$ ?



# Integrals

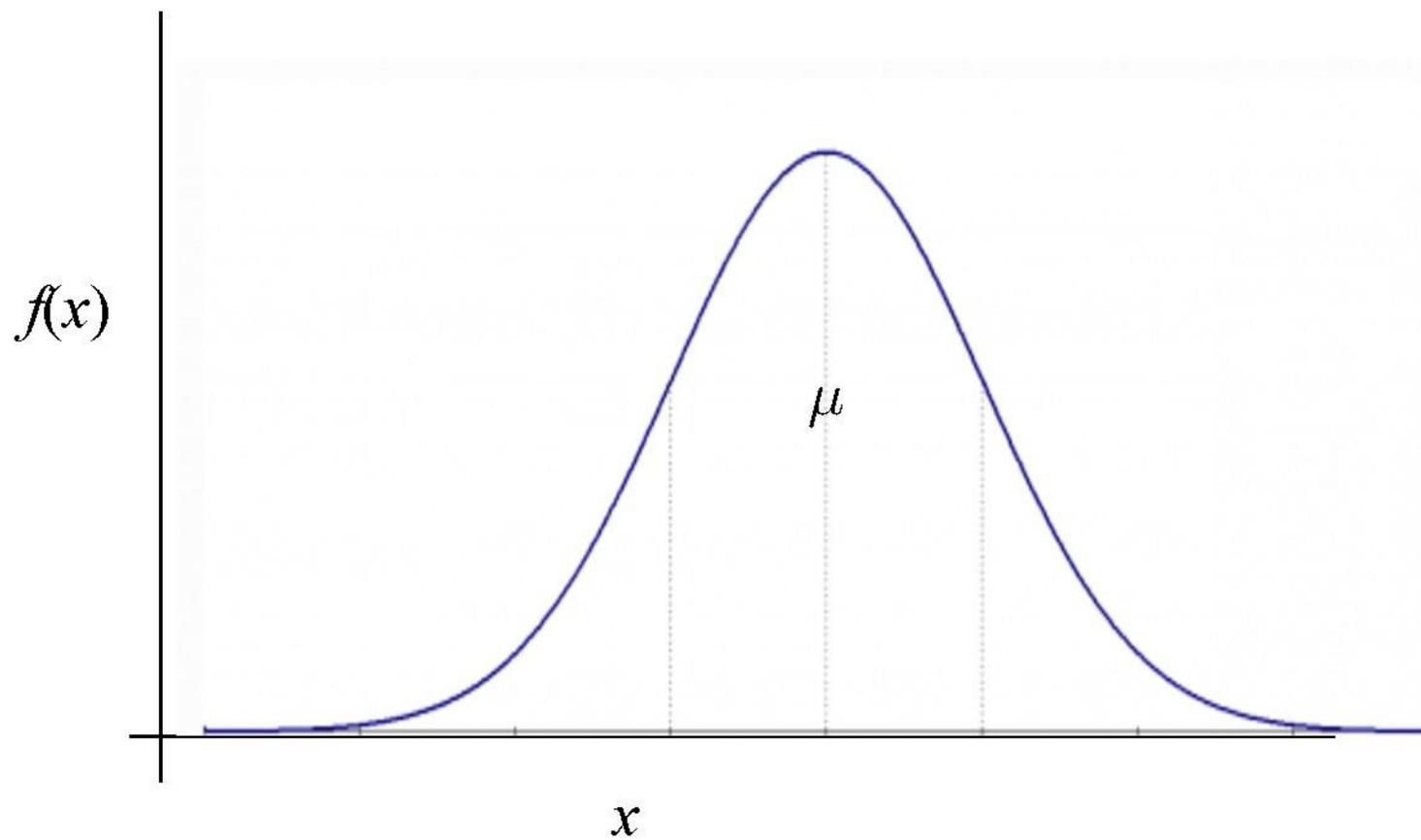


\*loving, not scary

# Probability Density Function

$$\mathcal{N}(\mu, \sigma^2)$$

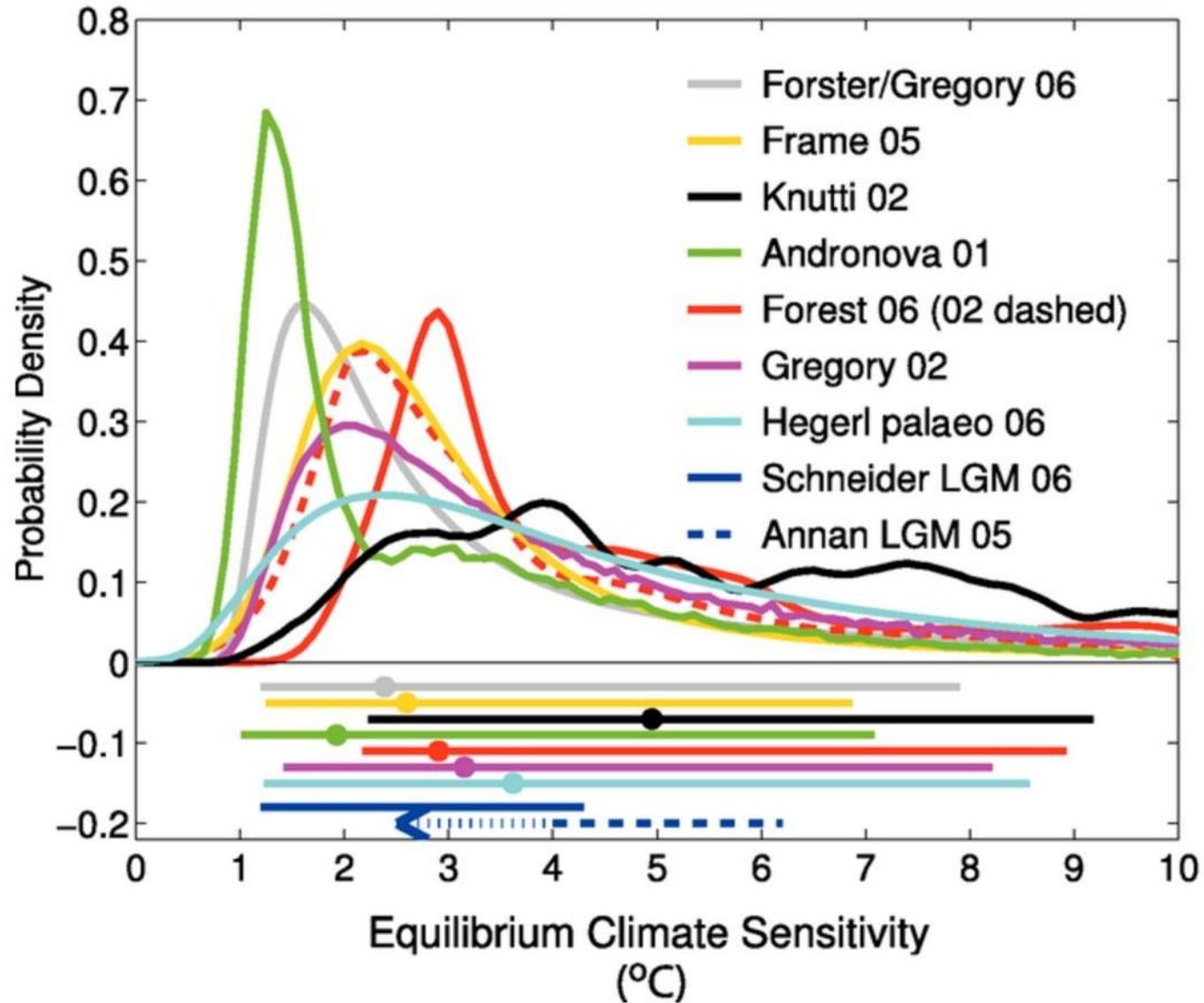
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



What do you get if you  
integrate over a  
*probability density function*?

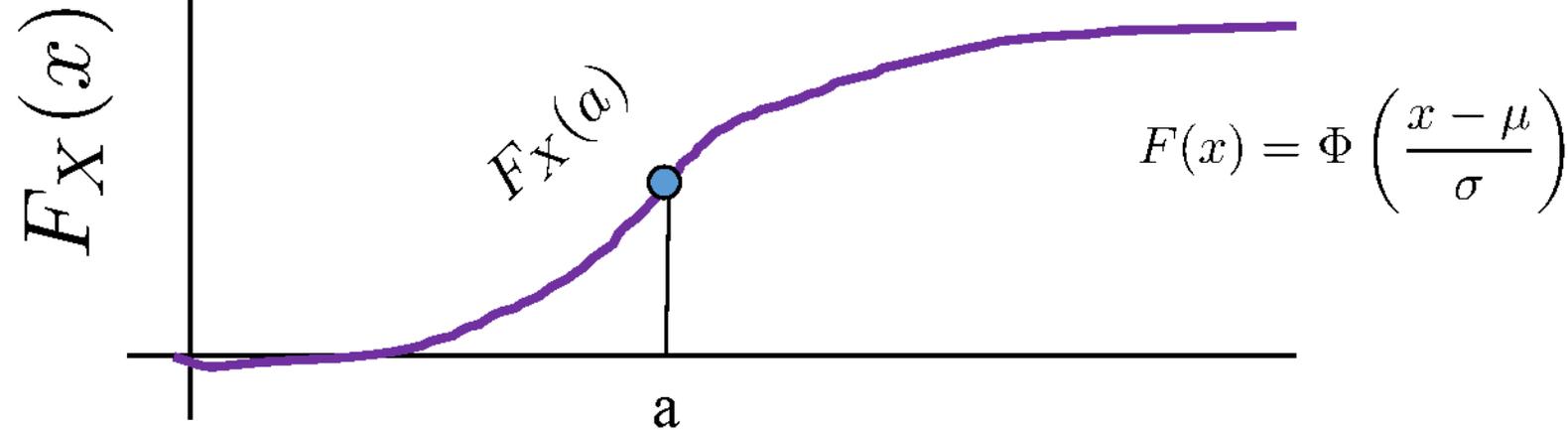
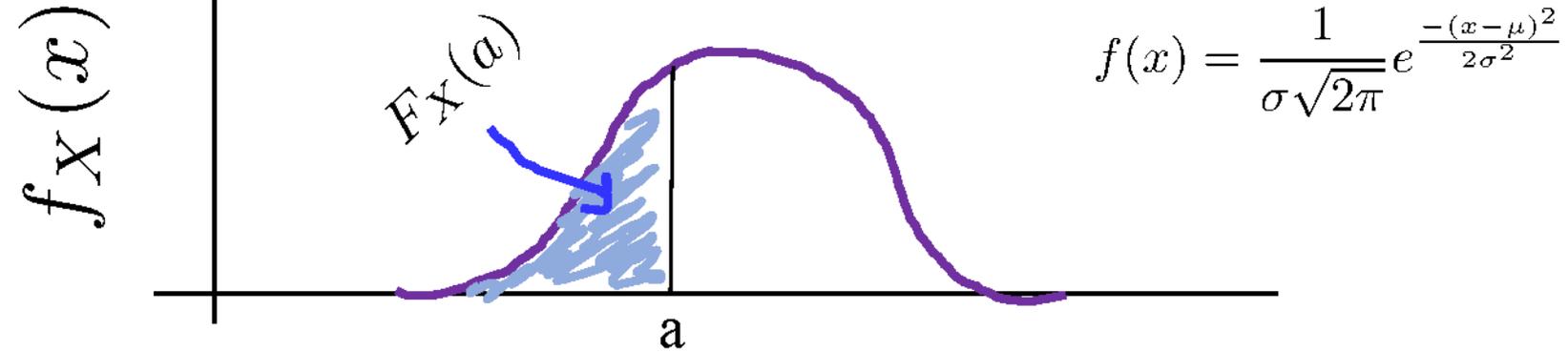
**A probability!**

# Climate Sensitivity



# PDF and CDF of a Normal

$$X \sim N(\mu, \sigma^2)$$



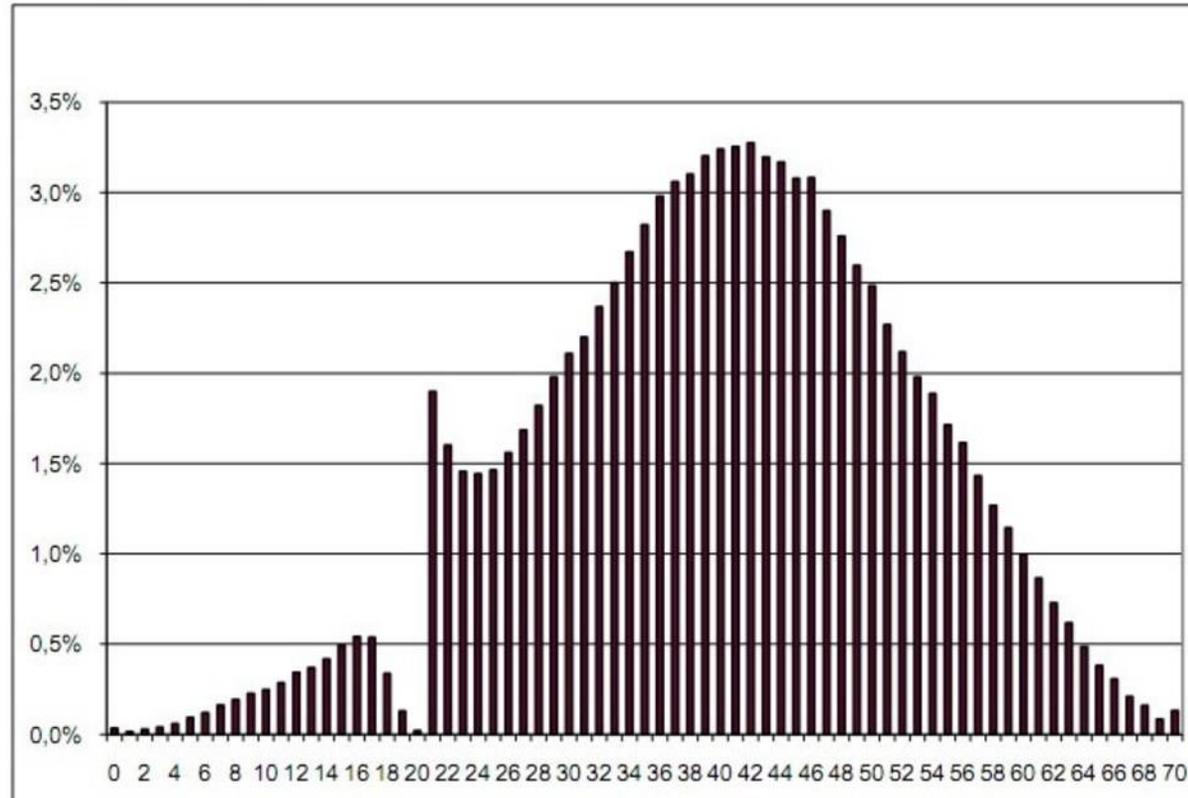
A CDF is the integral from  $-\infty$  to  $x$  of the PDF

# Altruism?

Scores for a standardized test that students in Poland are required to pass before moving on in school

See if you can guess the minimum score to pass the test.

## 2.1. Poziom podstawowy



Wykres 1. Rozkład wyników na poziomie podstawowym

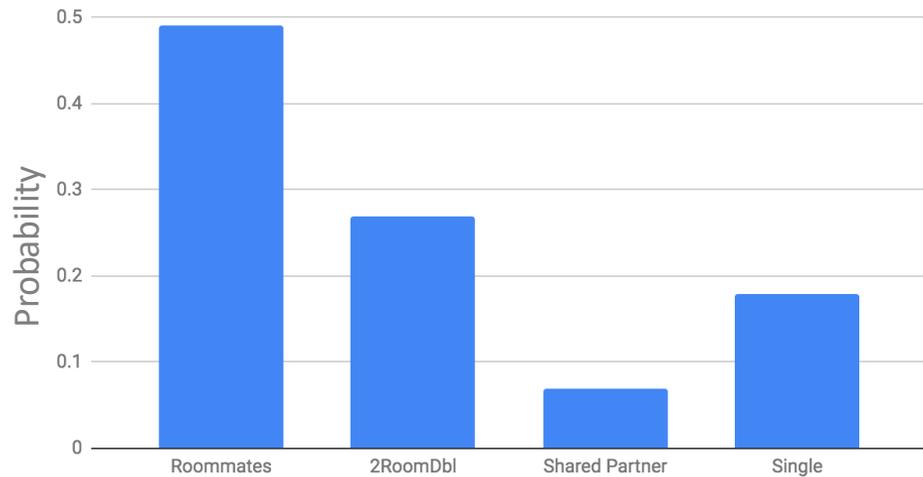
# Probabilistic Models



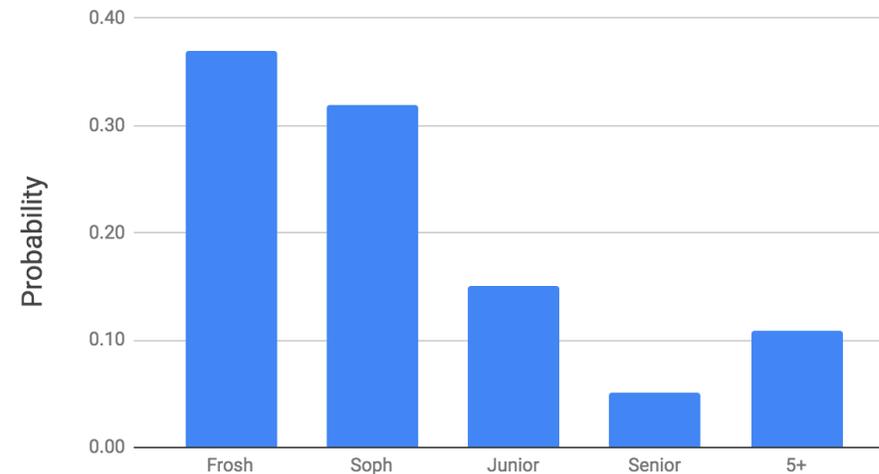
# Joint Probability Table

	Roommates	2RoomDbI	Shared Partner	Single	
Frosh	0.30	0.07	0.00	0.00	0.37
Soph	0.12	0.18	0.00	0.03	0.32
Junior	0.04	0.01	0.00	0.10	0.15
Senior	0.01	0.02	0.02	0.01	0.05
5+	0.02	0.00	0.05	0.04	0.11
	0.49	0.27	0.07	0.18	1.00

Marginal Room type



Marginal Year



# Multinomial

Example document:

“Pay for Viagra with a credit-card. Viagra is great.  
So are credit-cards. Risk free Viagra. Click for free.”

$n = 18$

$$P \left( \begin{array}{l} \text{Viagra} = 2 \\ \text{Free} = 2 \\ \text{Risk} = 1 \\ \text{Credit-card: } 2 \\ \dots \\ \text{For} = 2 \end{array} \middle| \text{spam} \right) = \frac{n!}{2!2! \dots 2!} p_{\text{viagra}}^2 p_{\text{free}}^2 \dots p_{\text{for}}^2$$

It's a Multinomial!

Probability of seeing  
this document | spam

The probability of a word in  
spam email being viagra



# General “Inference”



# General "Inference"

WebMD Symptom Checker BETA

INFO SYMPTOMS QUESTIONS CONDITIONS DETAILS TREATMENT

### Add more symptoms

or Choose common symptoms

bloating cough diarrhea dizziness fatigue

fever headache muscle cramp nausea

throat irritation

AGE 30 GENDER Male

**MY SYMPTOMS**

cough × throat irritation ×

sneezing ×

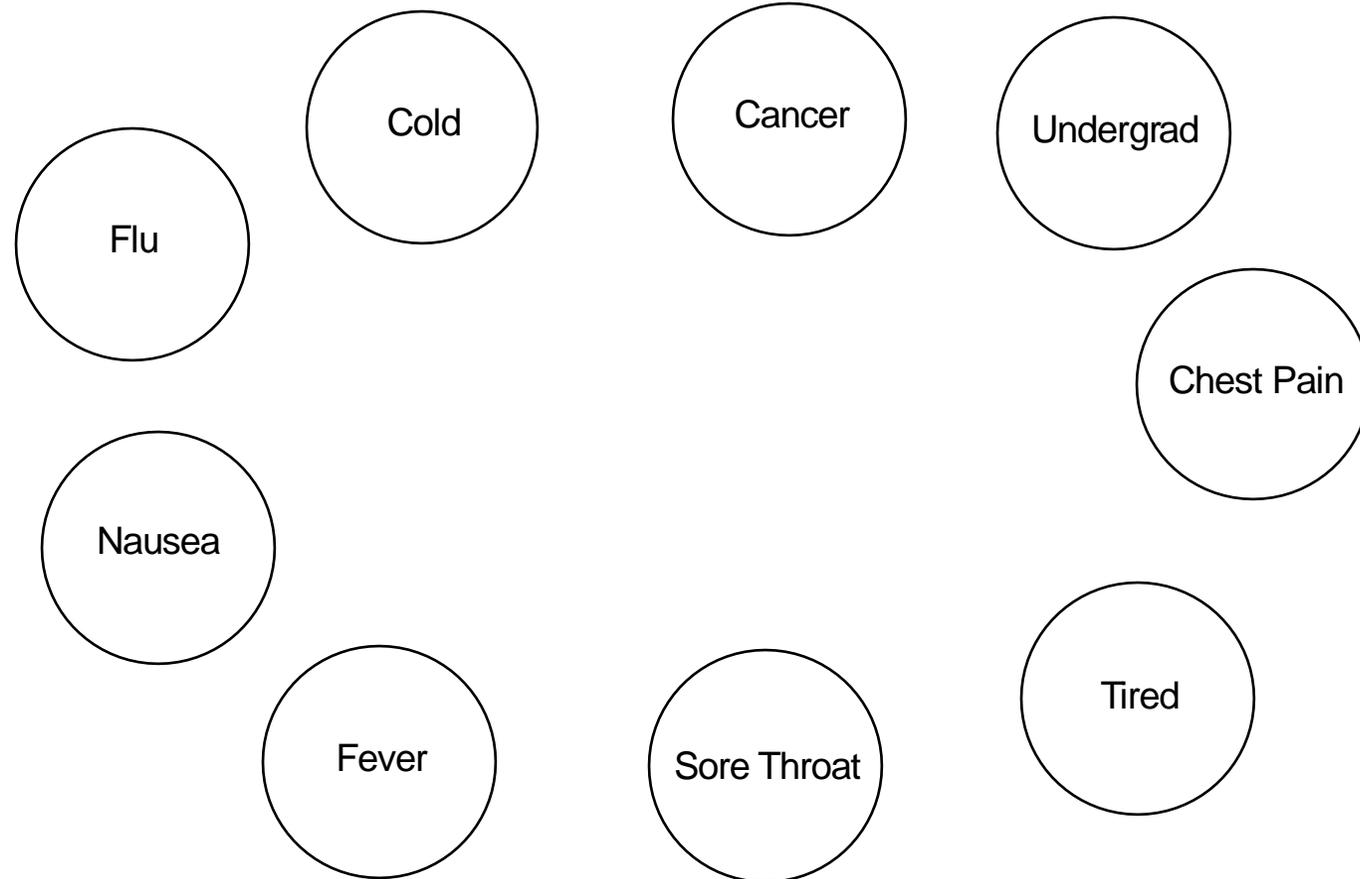
Results Strength: **MODERATE**



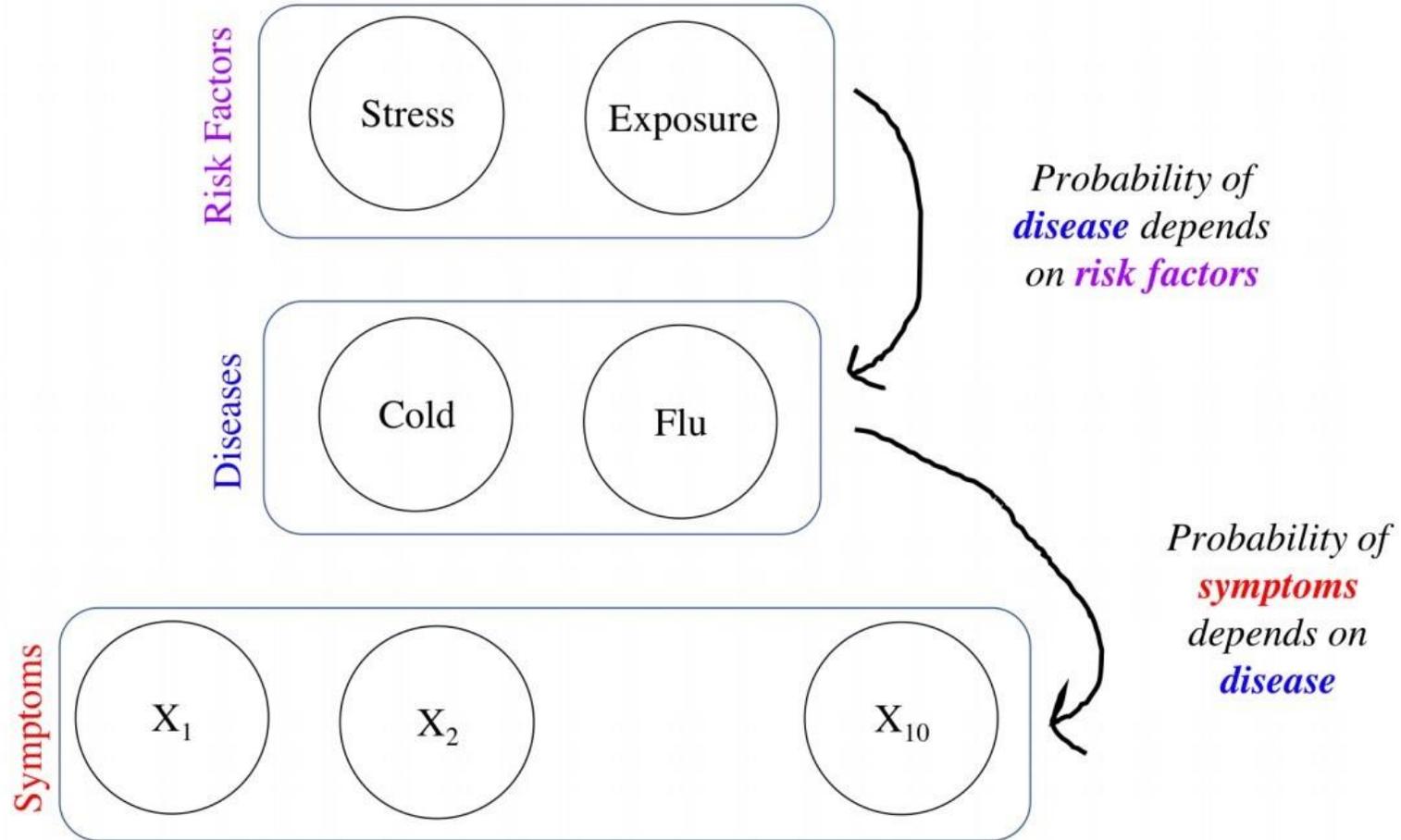
< Previous

Info  
Continue >

# Lots of Random Vars?



# Huge Joint Models

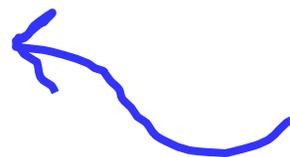


# Alg #1: Rejection Sampling

```
3 N_SAMPLES = 100000
4
5 # Program: Joint Sa
6 # -----
7 # we can answer any
8 # with multivariate
9 # where conditioned
10 def main():
11     obs = getObserv
12     print 'Observat
13
14     samples = sampl
15     prob = probFluG
16     print 'Pr(Flu)
```

```
[0, 0, 0, 0]
[0, 1, 0, 1]
[1, 0, 1, 0]
[1, 1, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 1, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[1, 1, 1, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[1, 1, 1, 1]
[0, 1, 0, 0]
```

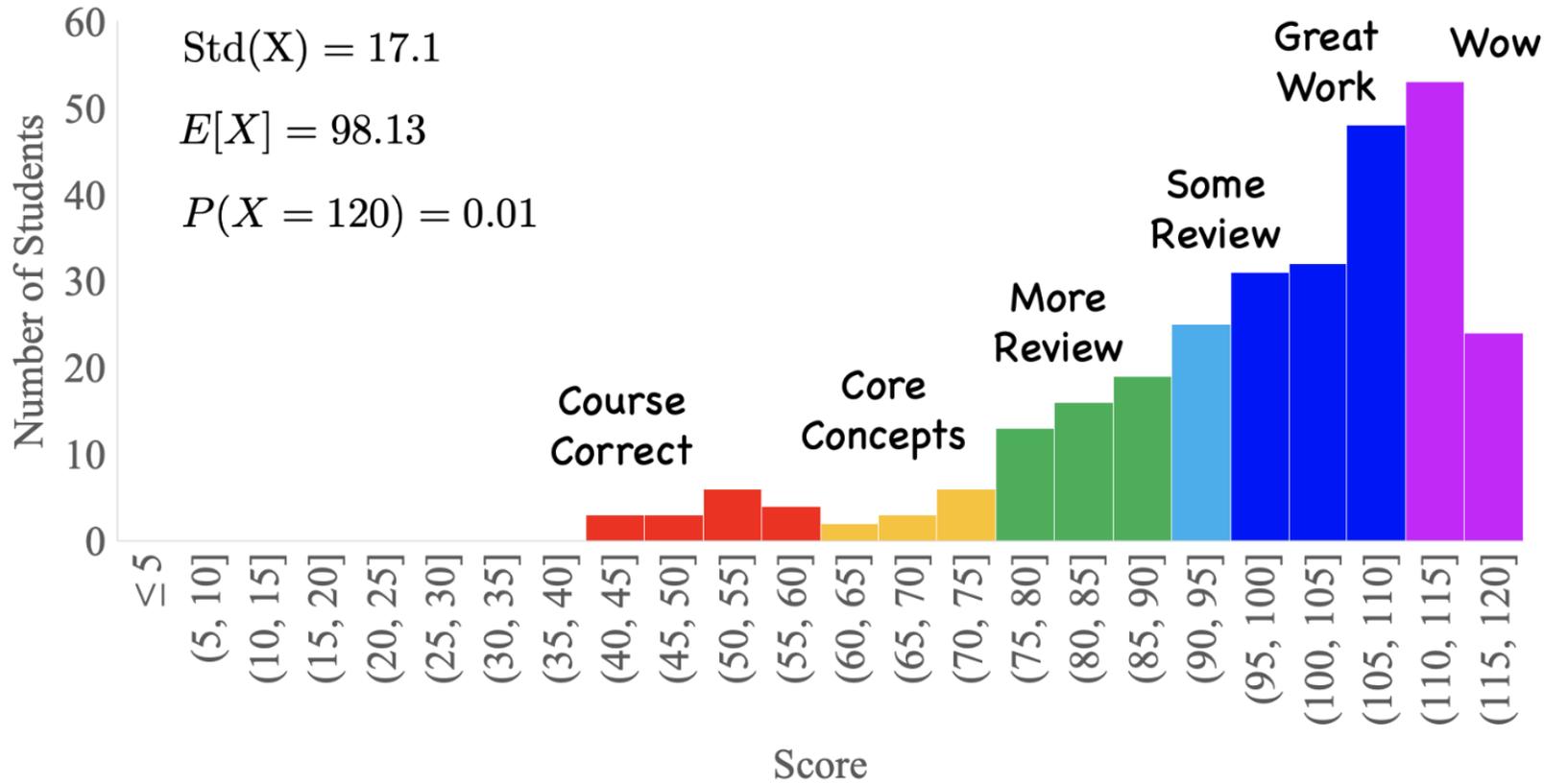
Each one of these is  
one posterior sample:



[Flu, Ugrad, Fever, Tired]

```
Observation = [None, None, None, 1]
Pr(Flu | Obs) = 0.140635888502
>
```

# Hard Midterm, Great Job



# PEP

New this year

## Personalized Exam Prep Signup: Final

PEP is back for the final. The only difference is that Final PEP is 15 mins long.

### What is PEP?

This quarter we are trying something **new in CS109!** In the past we have talked to many students *after* the final to get feedback on how they studied and what they found challenging. We often have some good insights for students, but it can feel like those insights are a few weeks too late. This quarter we are trying to get you those insights **before** the final so you can master the material more effectively. We call these 1:1s **Personalized Exam Prep**.

You meet in-person with a TA a week before the final for **15 mins**. You don't need to prepare or bring anything. The TA gets to know you and, after the session, sends you home with a draft of a study plan. Participating will get you an automatic 4 points on the final. If you can't participate, that is fine, your final will be graded as usual. You likely will not get your section TA, but it is possible (what is the probability???)

[View My Personalized Guide](#)

## Reserve a Time

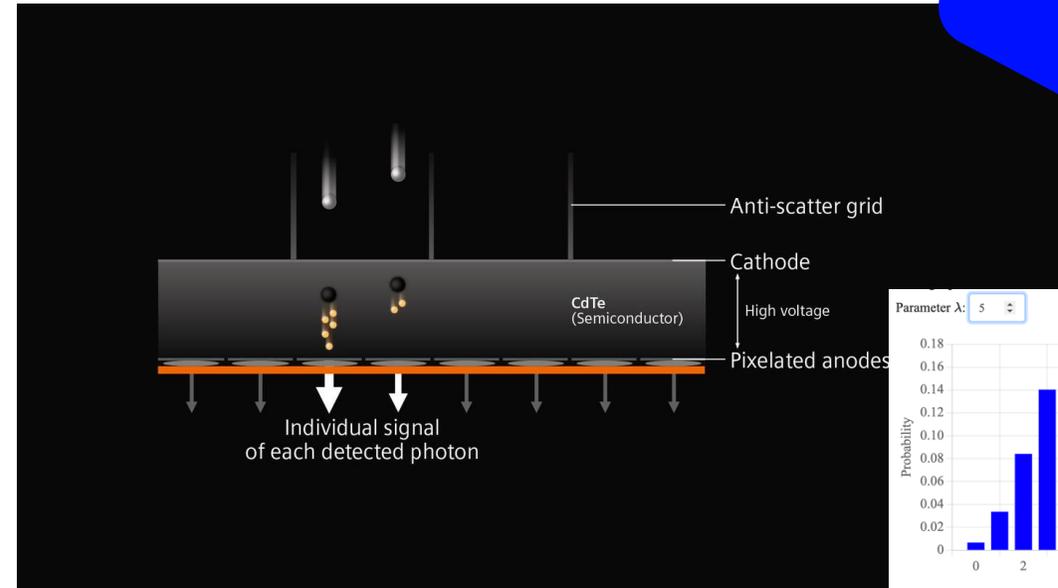
Tuesday, Dec 3

Select a time slot

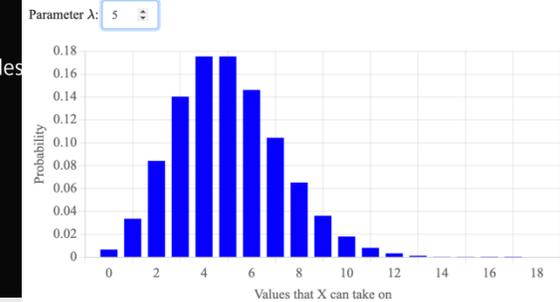
Wednesday, Dec 4

✓ Select a time slot

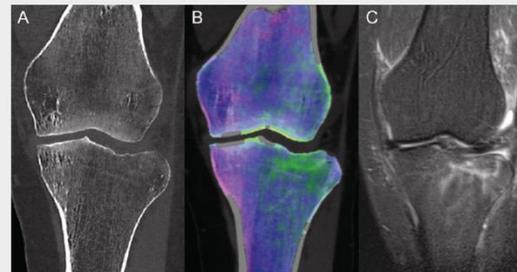
- Wednesday, Dec 4 9:15 AM
- Wednesday, Dec 4 9:30 AM
- Wednesday, Dec 4 12:00 PM
- Wednesday, Dec 4 12:15 PM



Poisson



## Bone marrow edema at photon-counting CT



A 56-year-old female patient with knee injury. (A) Diagnostic CT image acquired with photon-counting CT (PCCT) shows a nondisplaced fracture of the lateral tibial plateau. The associated bone marrow edema (BME) is shown on (B) the BME map reconstructed from the unenhanced PCCT image and (C) the fat-suppressed T2-weighted MRI scan.



"A 56-year-old female patient presented with severe pain after a ground-level fall on the left knee. Antero-posterior and lateral view radiographs of the knee demonstrated no evident abnormalities."



"Photon-counting CT (PCCT) demonstrated a nondisplaced fracture of the lateral tibial plateau. Furthermore, the bone marrow edema (BME) map reconstructed from the unenhanced PCCT image showed associated BME (Figure, B). This finding correlated well with BME seen on the fat-suppressed T2-weighted MRI scan."

Inference

Information Theory

288 Midterm, 290 Final 120 Hours of Extra TA time

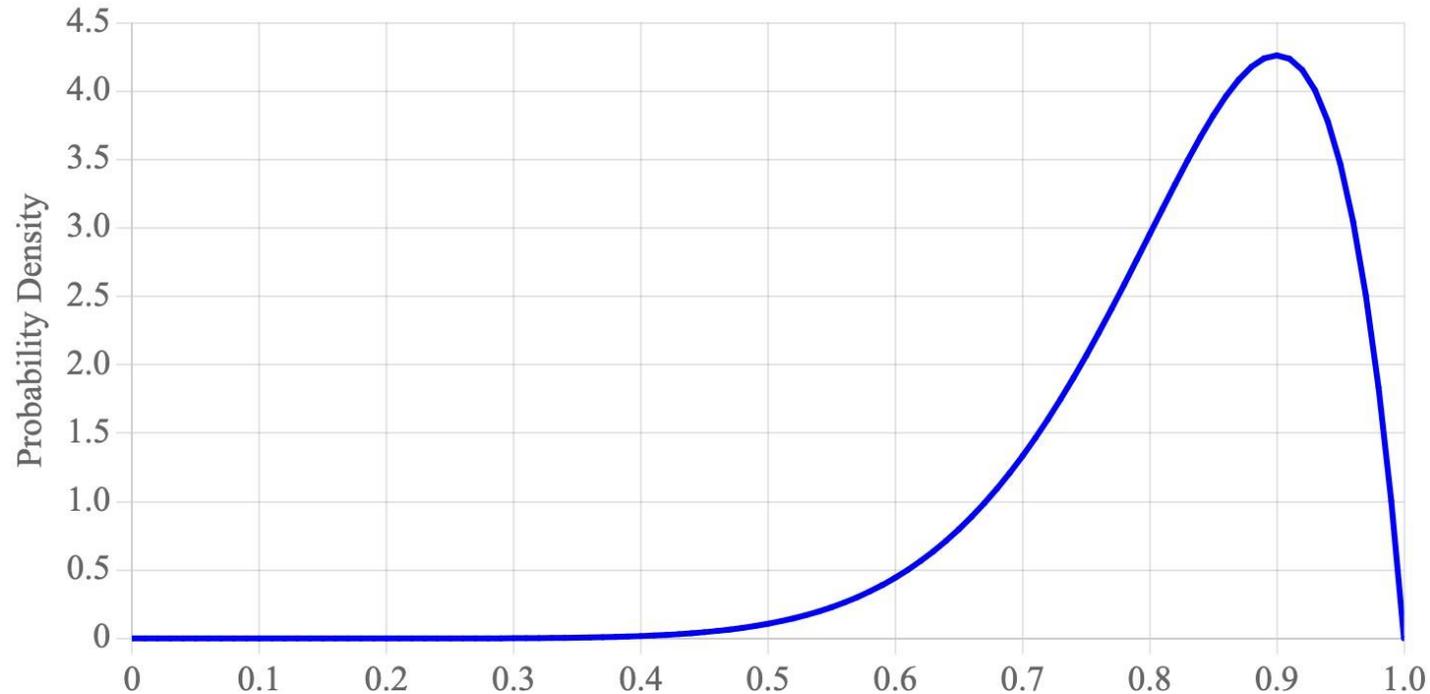
Learning Goal: Be fluent in the language of probability

# Uncertainty Theory



$$\begin{aligned}
& f(X = x|H = 9, T = 1) \\
&= \frac{P(H = 9, T = 1|X = x) \cdot f(X = x)}{P(H = 9, T = 1)} && \text{Bayes Theorem} \\
&= \frac{\binom{10}{9} x^9 (1-x)^1 \cdot f(X = x)}{P(H = 9, T = 1)} && \text{Binomial PMF} \\
&= \frac{\binom{10}{9} x^9 (1-x)^1 \cdot 1}{P(H = 9, T = 1)} && \text{Uniform PDF} \\
&= \frac{\binom{10}{9}}{P(H = 9, T = 1)} x^9 (1-x)^1 && \text{Constants to front} \\
&= K \cdot x^9 (1-x)^1 && \text{Rename constant}
\end{aligned}$$

Lets take a look at that function. For now we can let  $K = \frac{1}{110}$ . Regardless of  $K$  we will get the same shape, just scaled:



# Lets Play!

Drug A

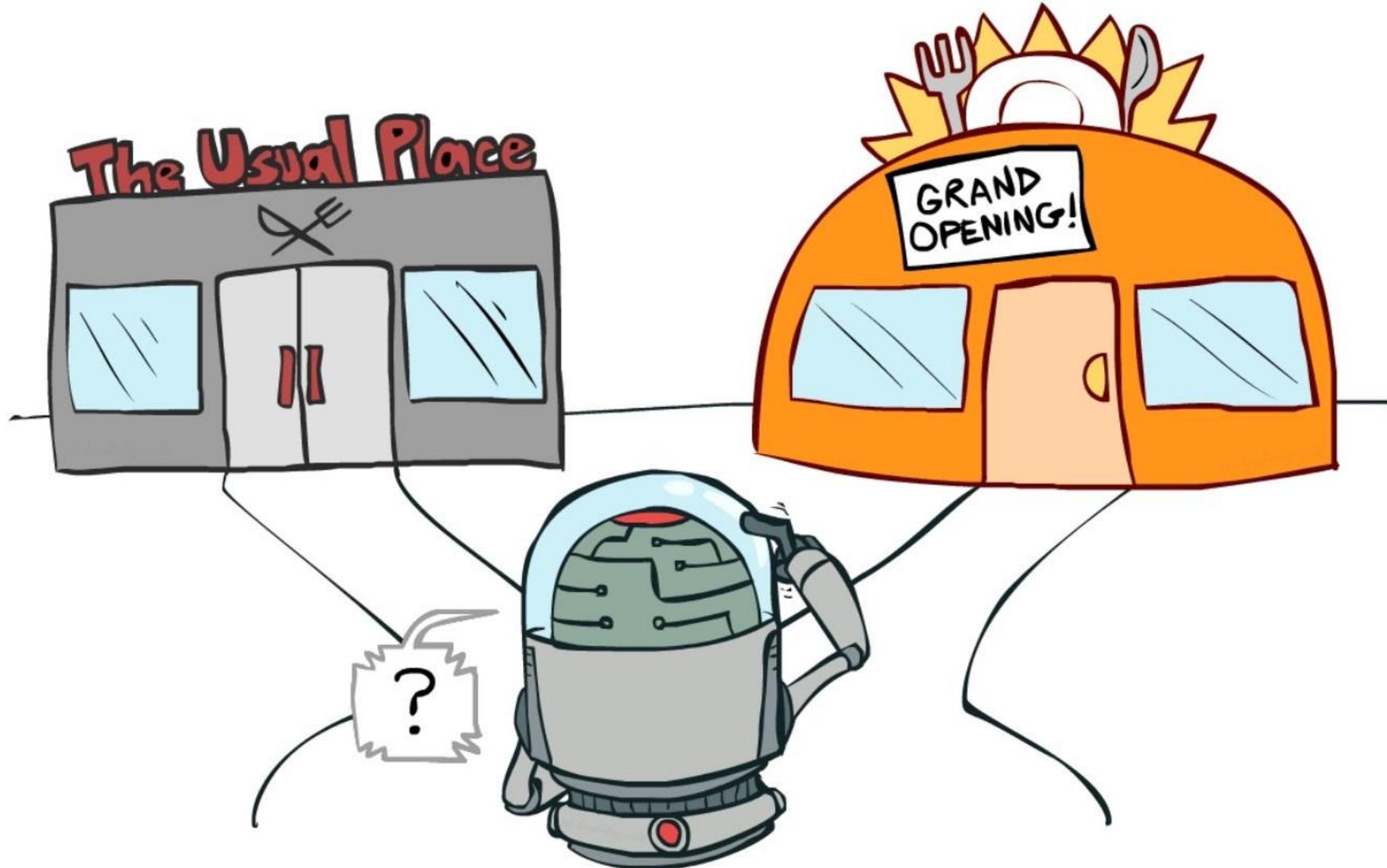


Drug B

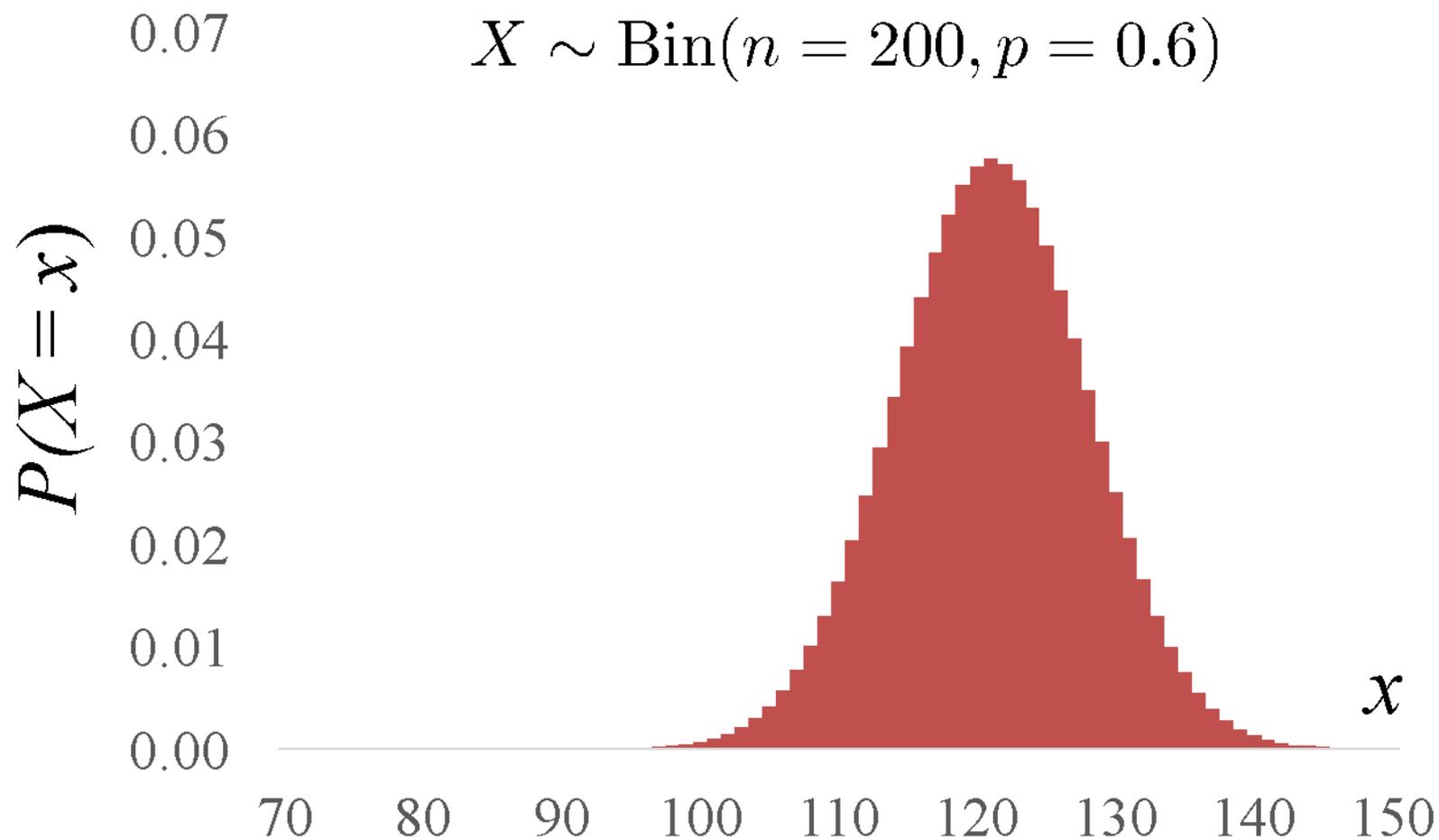


Which one do you give to a patient?

# Thompson Sampling

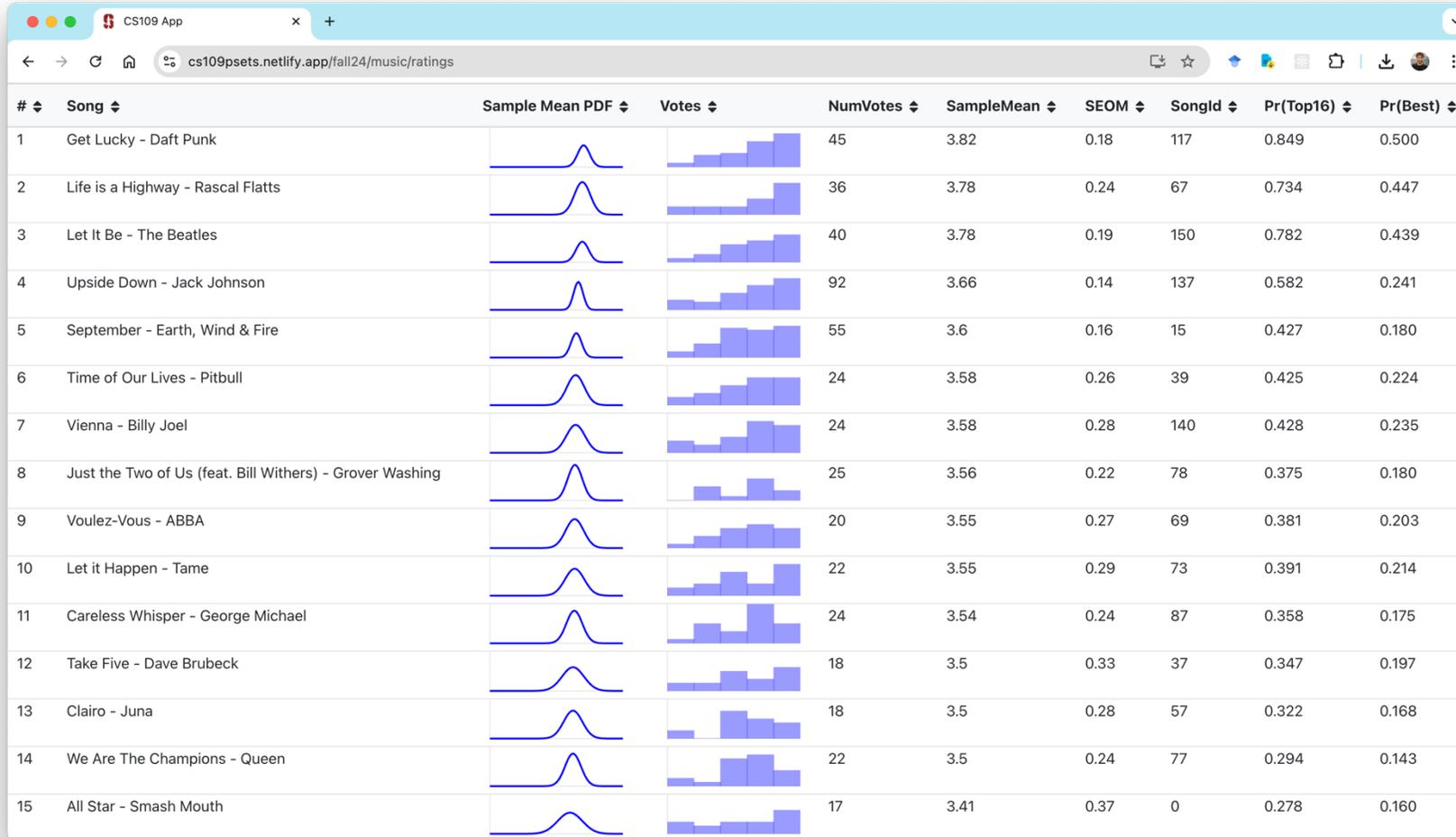


# C.L.T. Explains This



# Class Theme Song

New this year



The screenshot shows a web browser window with the URL `cs109psets.netlify.app/fall24/music/ratings`. The page displays a table of 15 songs with various metrics. Each row includes a sample mean PDF (a blue curve) and a histogram (a blue bar chart) representing the distribution of votes for that song.

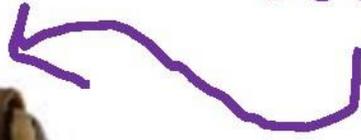
#	Song	Sample Mean PDF	Votes	NumVotes	SampleMean	SEOM	SongId	Pr(Top16)	Pr(Best)
1	Get Lucky - Daft Punk			45	3.82	0.18	117	0.849	0.500
2	Life is a Highway - Rascal Flatts			36	3.78	0.24	67	0.734	0.447
3	Let It Be - The Beatles			40	3.78	0.19	150	0.782	0.439
4	Upside Down - Jack Johnson			92	3.66	0.14	137	0.582	0.241
5	September - Earth, Wind & Fire			55	3.6	0.16	15	0.427	0.180
6	Time of Our Lives - Pitbull			24	3.58	0.26	39	0.425	0.224
7	Vienna - Billy Joel			24	3.58	0.28	140	0.428	0.235
8	Just the Two of Us (feat. Bill Withers) - Grover Washing			25	3.56	0.22	78	0.375	0.180
9	Voulez-Vous - ABBA			20	3.55	0.27	69	0.381	0.203
10	Let it Happen - Tame			22	3.55	0.29	73	0.391	0.214
11	Careless Whisper - George Michael			24	3.54	0.24	87	0.358	0.175
12	Take Five - Dave Brubeck			18	3.5	0.33	37	0.347	0.197
13	Clairo - Juna			18	3.5	0.28	57	0.322	0.168
14	We Are The Champions - Queen			22	3.5	0.24	77	0.294	0.143
15	All Star - Smash Mouth			17	3.41	0.37	0	0.278	0.160

C.L.T. tells us how to think about the average rating after few votes.

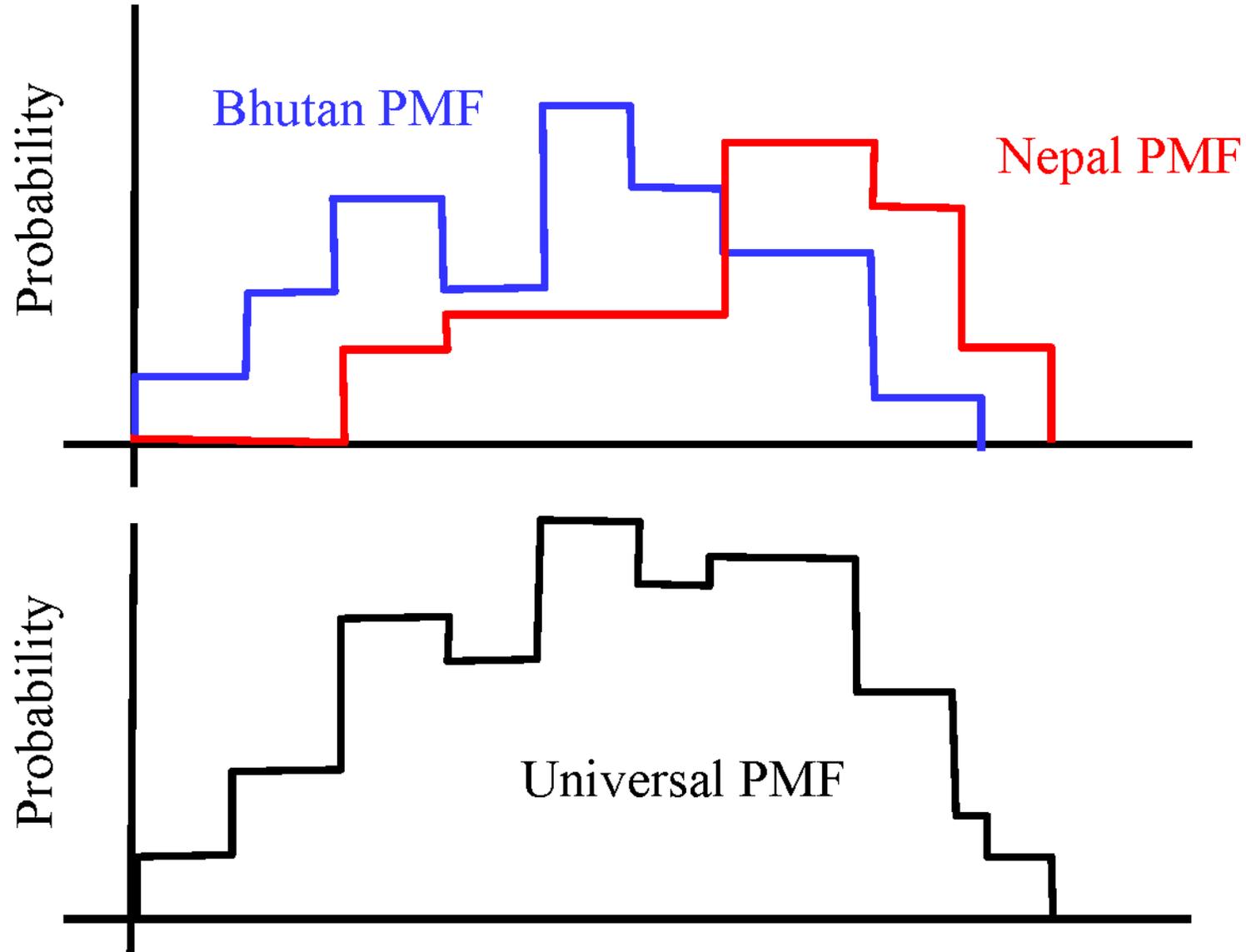
Used to adaptively decide which song to vote on next

Reflection: I love music. Should we focus more on love of songs instead of average?

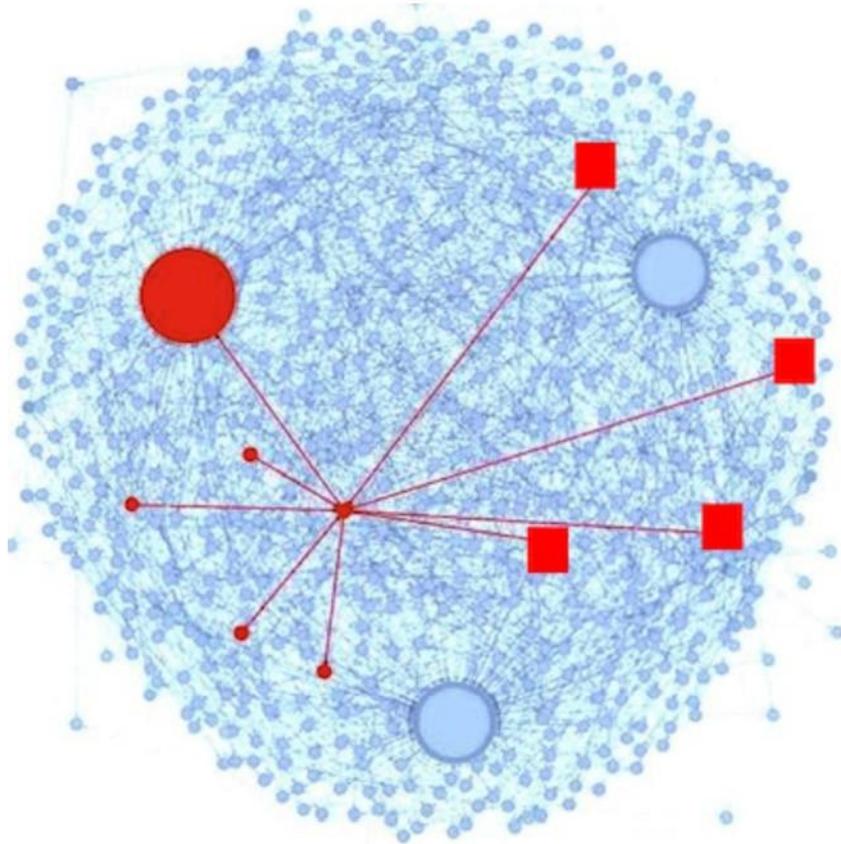
Bootstrap



# Universal Sample



# Peer Grading

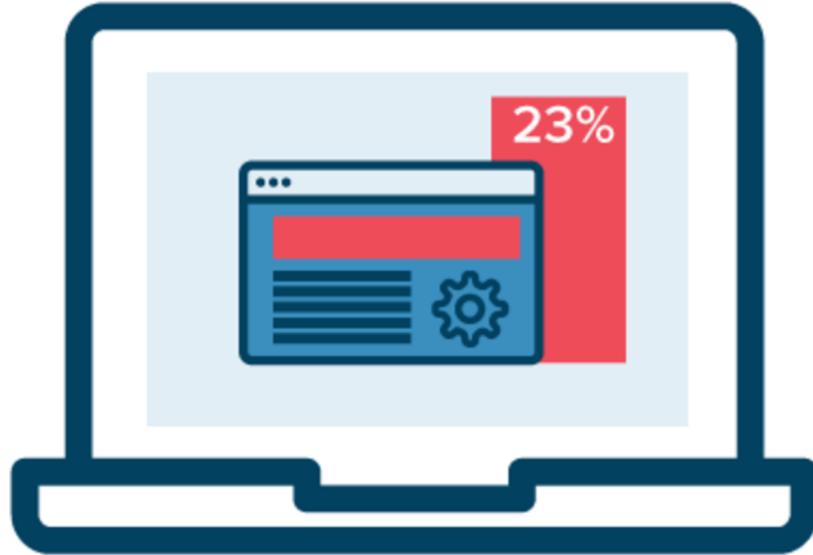


Peer Grading on Coursera  
HCI.

31,067 peer grades for  
3,607 students.

# A/B Testing

## A



CONTROL

## B



VARIATION



# Information Theory

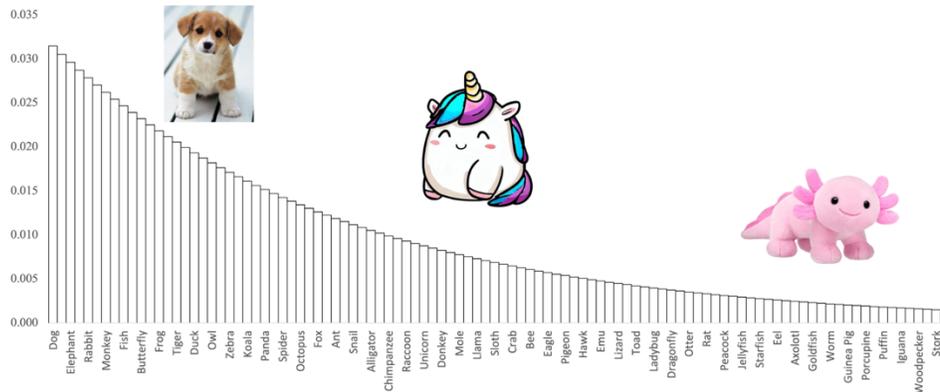


## Information Theory

Information theory is an incredibly powerful perspective which plays a central role in a ton of algorithms, including Decision Trees, the WordleBot, Adaptive Tests, Optimal Poker Play and even ompression of data (like Huffman Encoding or even Jpeg files)! The goal of this chapter is to balance showing off the awesome power of Information Theory while also keeping things as straight forward as possible. To that end, a great place to start is thinking about how you could write a bot that can play the question answering game of, "Think of an Animal".

### Think of an Animal!

The game of "Think of an Animal" goes like this: The human is going to be thinking of an animal. We assume that the distribution of how often they chose an animal is known (based off how popular the animal is to four year olds):



The task of your algorithm is to select which question to ask next. Assume you are given a bank of yes or no questions which include classics like:

- Is it a pet?
- Does it live in the water?
- Are you thinking of a dog?

The screenshot shows a browser window with a Wordle game. The game board is partially filled with colored squares: grey (unknown), green (correct letter in correct position), and yellow (correct letter in wrong position). A cartoon robot is positioned in the center of the board. The text above the board asks for the next guess based on the remaining words: bring, girls, storm, tears, rates, grind, agirt. Below the board, a Python code editor shows a probability mass function (pmf) for the remaining words. The code defines a dictionary for word probabilities and a function for generating feedback based on a guess.

```
word_pmf = {"bring": 0.365, "girls": 0.296, "storm": 0.135, "tears": 0.074, "rates": 0.061, "grind": 0.068, "agirt": 0.001}
```

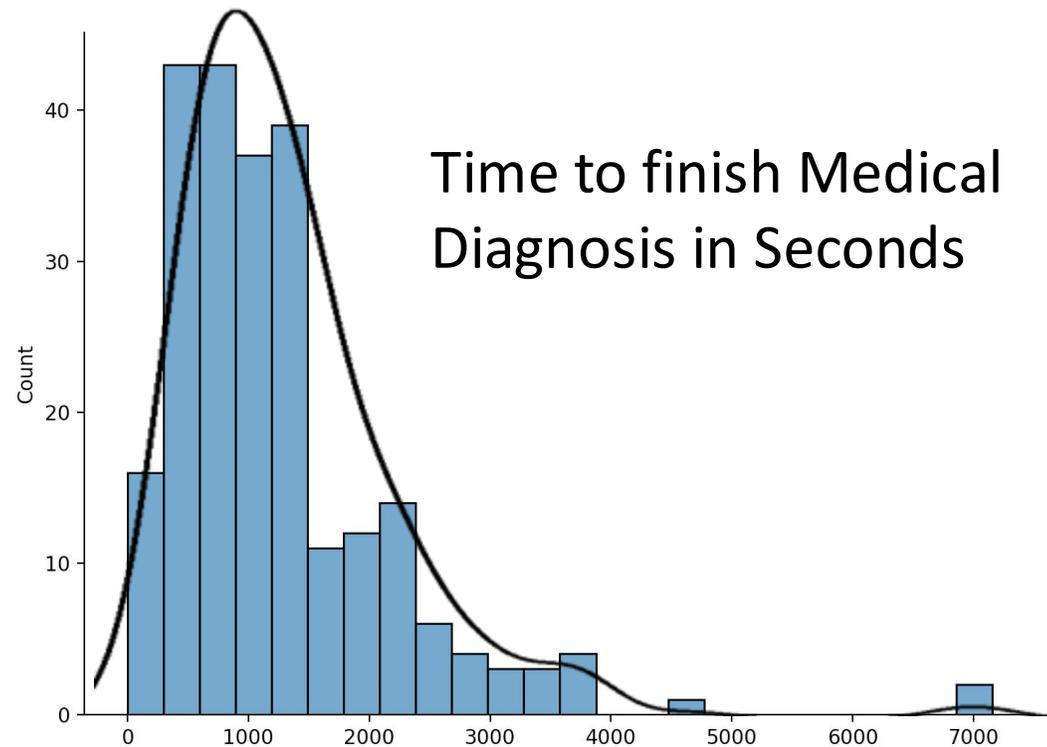
```
def main():  
    word_pmf = {  
        "bring": 0.365,  
        "girls": 0.296,  
        "storm": 0.135,  
        "tears": 0.074,  
        "rates": 0.061,  
        "grind": 0.068,  
        "agirt": 0.001  
    }  
  
    # after you guess "girls" you could get any of  
    # the following feedback. Each feedback  
    # string maps to a list of all possible words  
    # consistent with that feedback.  
    feedback_map = {  
        "YYY--": ["bring", "agirt"],  
        "GGGGG": ["girls"],  
        "Y...": ["storm"]  
    }
```

# Machine Learning



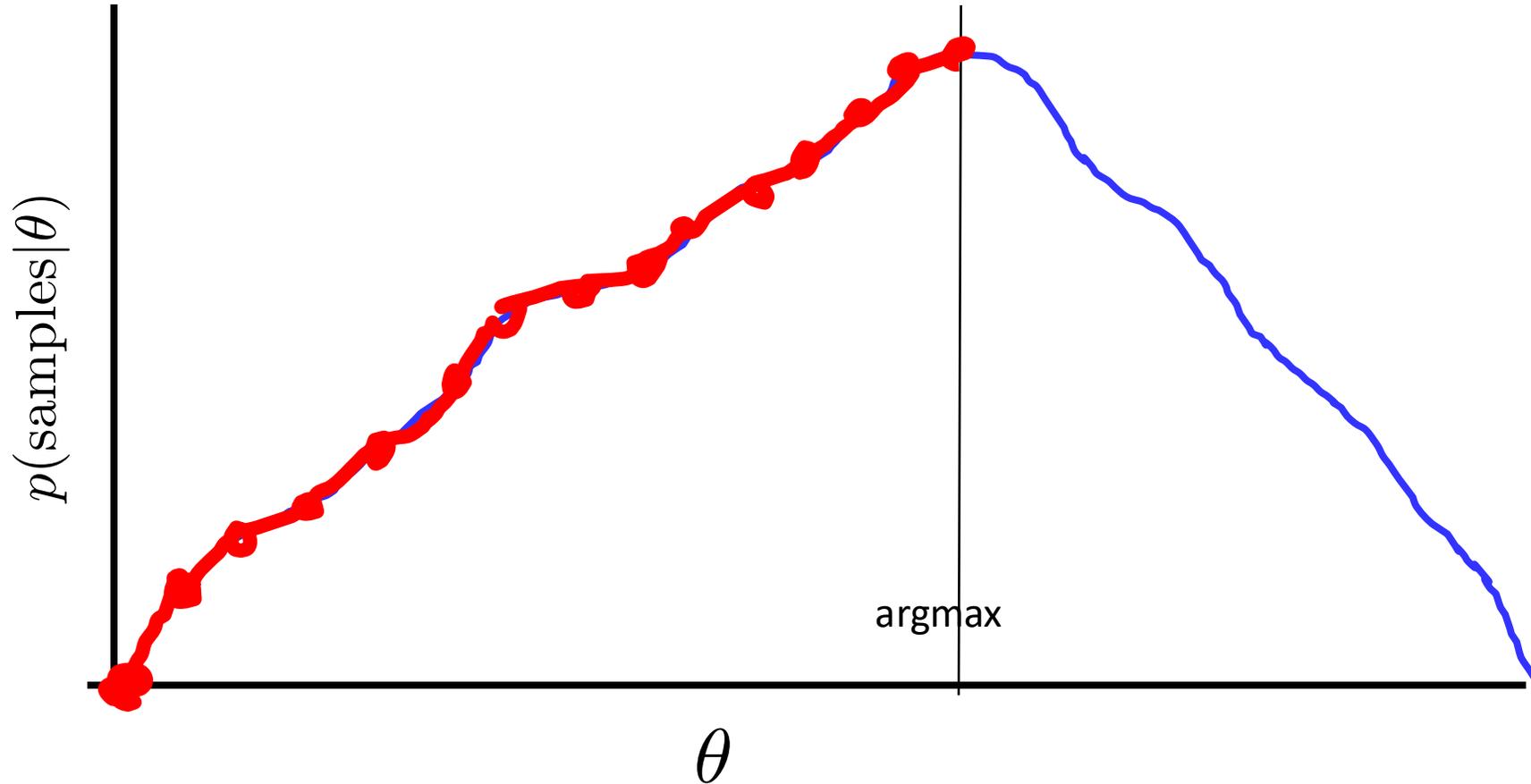
# MLE for Choosing Params

```
[3.002, 0.983, 2.186, 1.624, 3.997, 1.777,  
2.809, 0.42, 0.515, 1.582, 0.948, 0.458, 1.  
066, 0.8, 2.398, 0.794, 2.561, 2.61, 0.  
595, 3.897, 1.852, 1.182, 3.043, 0.905, 1.  
45, 0.405, 0.445, 2.103, 1.425, 3.12, 0.  
973, 1.056, 3.715, 2.952, 1.817, 2.686, 4.  
173, 0.358, 2.185, 2.581, 7.134, 0.206, 2.  
049, 0.896, 2.095, 4.39, 2.199, 3.434, 5.  
696, 0.819, 0.416, 1.571, 1.337, 2.79, 2.  
701, 3.061, 4.677, 0.671, 1.594, 3.586, 2.  
708, 1.417, 1.799, 1.137, 1.771, 2.12, 0.  
93, 6.835, 3.213, 2.541, 2.505, 1.257, 1.  
99, 1.5, 0.014, 3.856, 0.979, 2.413, 2.  
596, 1.653, 0.881, 4.457, 0.717, 3.305, 2.  
456, 3.462, 1.737, 0.968, 0.528, 0.18, 1.  
626, 2.224, 1.466, 1.6, 1.572, 0.12, 2.86,  
1.062, 2.139, 1.217]
```



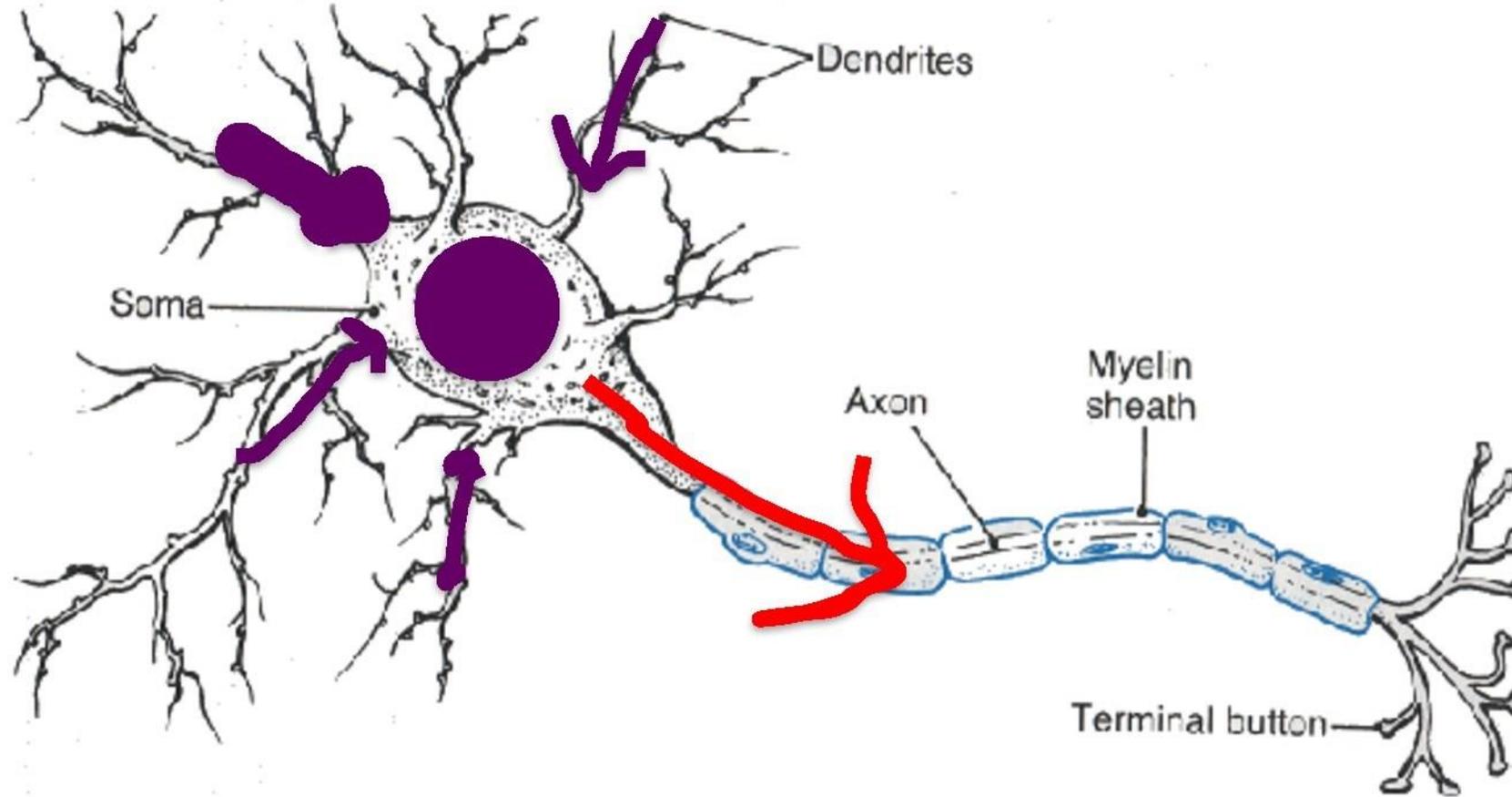
$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}$$

# Gradient Ascent



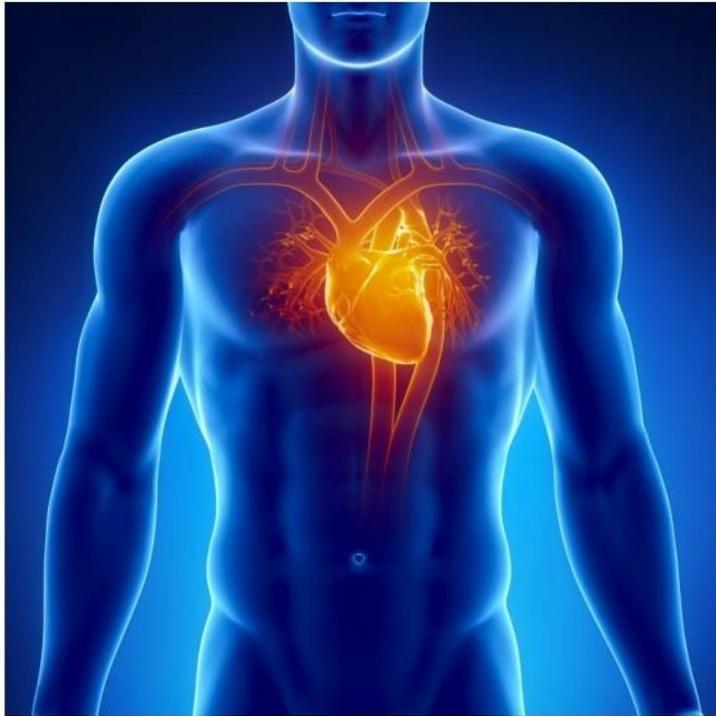
Walk uphill and you will find a local maxima  
(if your step size is small enough)

# Logistic Regression



# Machine Learning

Heart



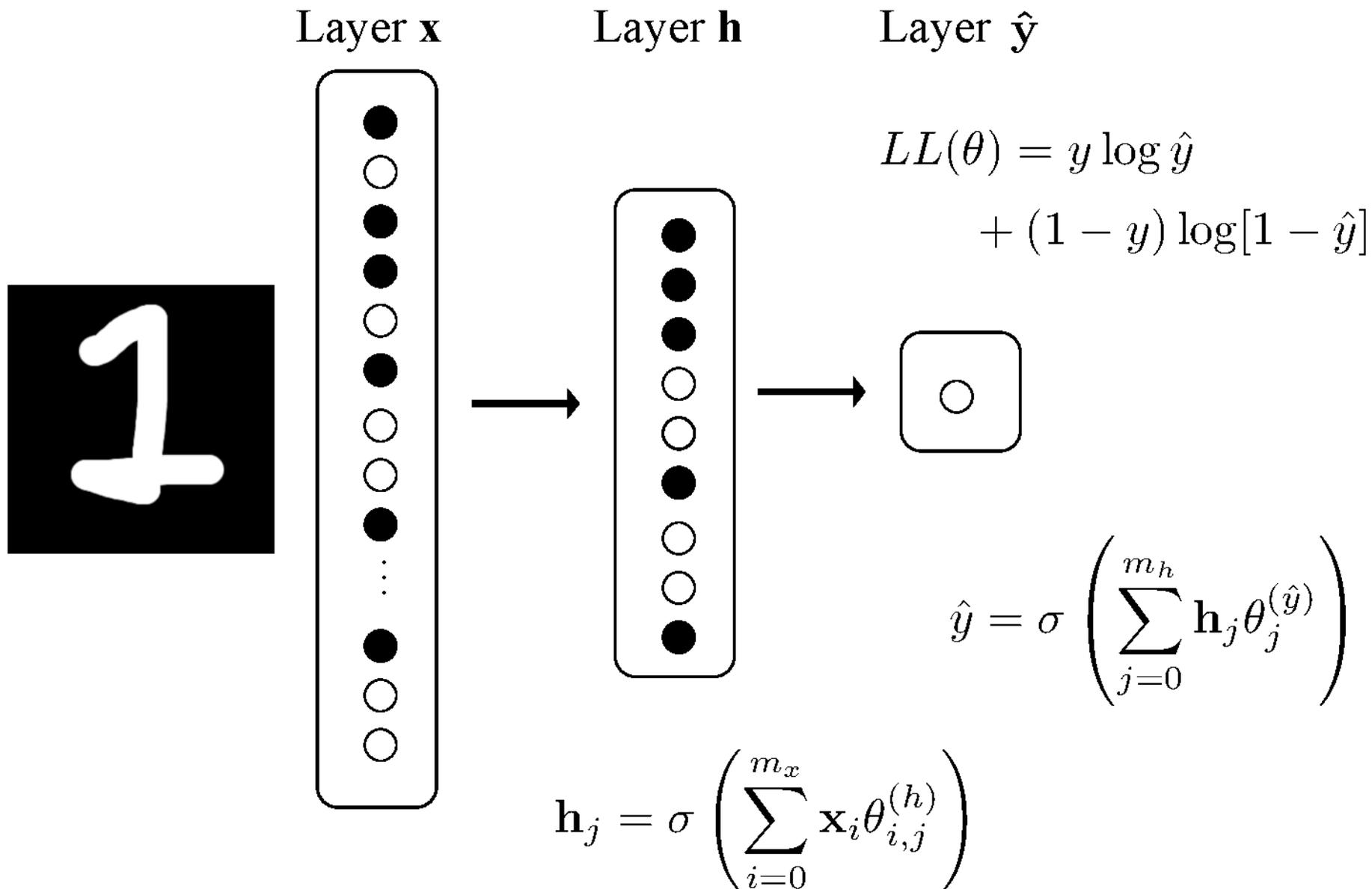
Ancestry



Netflix

**NETFLIX**

# Deep Learning



# Calibration

New!

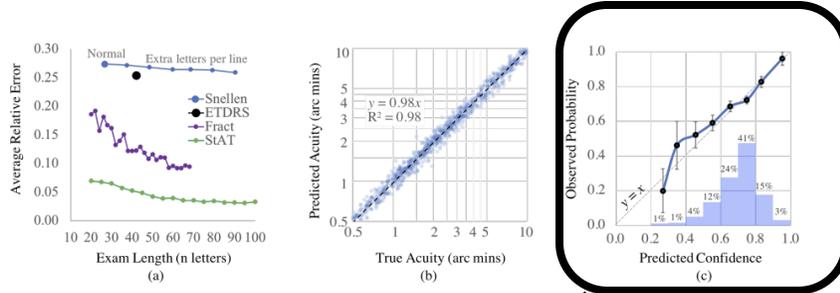


Figure 4: (a) The tradeoff between length of exam and error for the different algorithms. (b) Visualization of the predictions made by StACT. (c) Calibration test: StACT confidences correspond to how often it is correct.

## 4.2 Baseline Acuity Tests

We use the following baselines and prior algorithms to compare against the StACT algorithm.

**Const Policy.** This policy always predicts the most common visual acuity in our data i.e. the mode of the visual acuity prior. This serves as a true null model because it doesn't take patient responses into account at all.

**Snellen and ETDRS.** The Revised 2000 Series ETDRS charts and the Traditional Snellen Eye Chart were programmed so that we could simulate their response to different virtual patients. Both exams continue until the user incorrectly answers questions for more than half of the letters on a line. ETDRS has a function for predicted acuity score that takes into account both the last line passed, and how many letters were read on the last line not-passed. Both charts use 19 unique optotypes.

**FrACT.** We use an implementation of the FrACT algorithm (Bach and others 1996), with the help of code graciously shared by the original author. We also included the ability to learn the "s" parameter as suggested by the 2006 paper (Bach 2006), and verified that it improved performance.

## 5 Results and Evaluation

The results of the experiments can be seen in Table 1.

**Accuracy and error.** As can be seen from Table 1, the StACT test has substantially less error than all the other baselines. After 20 optotype queries, our algorithm is capable of predicting acuity with an average relative error of 0.069. This prediction is a 74% reduction in error from our implementation of the ubiquitous Snellen test (average error = 0.276), as well as a 67% reduction in error from the FrACT test (average error = 0.212). One possible reason for the improvement over FrACT is that the simulations used in our evaluations are based off the Floored-Exponential model that StACT uses. However, even when we evaluate StACT on simulations drawn from the FrACT logistic assumption we still achieve a 41% reduction. The improved accuracy of the StACT algorithm suggests our Bayesian approach

	$\mu$ Acuity Error	$\mu$ Test length
Const	0.536	0
Snellen <sup>†</sup>	0.264	27
ETDRS <sup>†</sup>	0.254	42
FrACT	0.212	20
StACT	<b>0.069</b>	20
StACT-noSlip	0.150	20
StACT-greedyMAP	0.132	20
StACT-logistic	0.125	20
StACT-noPrior	0.090	20
StACT-goodPrior	<b>0.047</b>	20
StACT-star	<b>0.038</b>	63

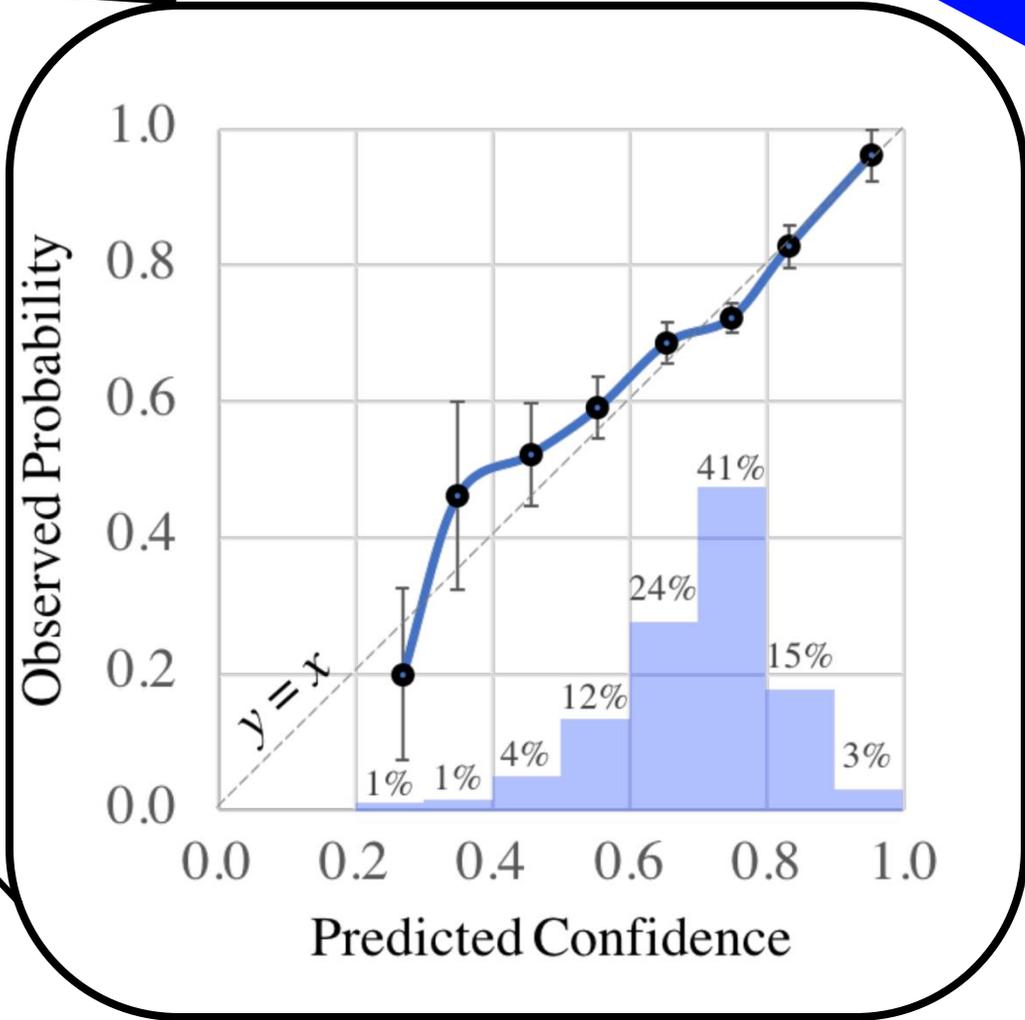
Table 1: Average relative error for each algorithm. Except for Snellen each test was allowed 20 letters. Results are average relative error after 1000 tests. <sup>†</sup> Snellen and ETDRS used 19 unique optotypes.

to measuring acuity is a fruitful proposal both because of our introduction of the floored exponential as well as our Thompson-sampling inspired algorithm to chose a next letter size.

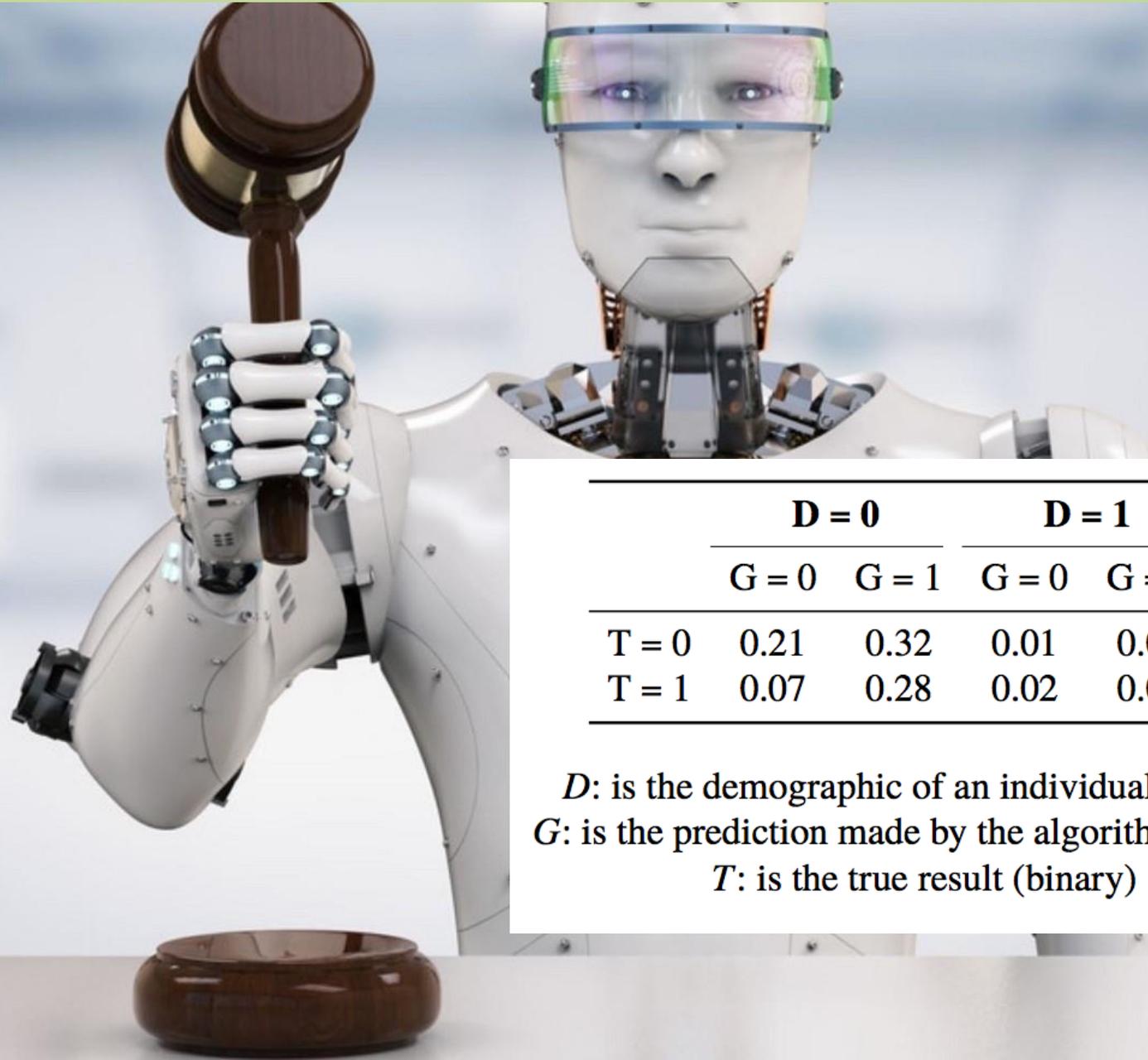
Figure 4 (b) visualizes what StACT's small relative error means in terms of predictions. Each point in the plot is a single patient. The x-axis is the true acuity of the patient and the y-axis is the predicted accuracy. We can qualitatively observe that the predictions are often accurate, there are no truly erroneous predictions, and that the exam is similarly accurate for patients of all visual acuities.

Moreover, as seen in Figure 4 (a), StACT's significant improvement in error rate holds even when the length of the exam is increased. It is also evident that increasing exam length reduces our error rate: if we increase the exam length to 200 letters, the average error of StACT falls to 0.020. While this is highly accurate, its far too long an exam, even for patients who need to know their acuity to high precision.

**StACT Star Exam.** Our primary experiments had a fixed



# Algorithmic Fairness



	<b>D = 0</b>		<b>D = 1</b>	
	<b>G = 0</b>	<b>G = 1</b>	<b>G = 0</b>	<b>G = 1</b>
<b>T = 0</b>	0.21	0.32	0.01	0.01
<b>T = 1</b>	0.07	0.28	0.02	0.08

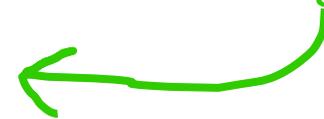
*D*: is the demographic of an individual (binary)  
*G*: is the prediction made by the algorithm (binary)  
*T*: is the true result (binary)

# Other Tasks, Other Models

New!

Model	Train Accuracy	Test Accuracy
Baseline	0.6138	0.6300
Logistic Regression	0.7300	0.7200
Naive Bayes	0.7275	0.7200
Decision Tree	0.7975	0.6150
Random Forest	0.7950	0.7100
<b>Gradient Boosting</b>	0.7738	<b>0.7250</b>
AdaBoost	0.7588	0.7100

The Kaggle Champion



By the numbers

# ~50 Major Keys



By the Central Limit Theorem, the mean of IID variables are distributed normally. As  $n \rightarrow \infty$

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

~600 Fruit



# 0 Fruit Related Injuries



# 1500 EdStem Answers

The screenshot shows a web browser window displaying an EdStem discussion page. The browser tabs include 'Probability for Computer Science' and '(2) CS 109 - Ed Discussion'. The address bar shows 'edstem.org/us/courses/67646/discussion/5828924'. The page header is 'ed CS 109 - Ed Discussion'. On the left, there is a sidebar with navigation options: 'New Thread', 'Chat', 'COURSES' (with 'CS 109' selected), 'Drafts', 'Scheduled', and 'CATEGORIES' (General, Lectures, Sections, Problem Sets, Exams, Fun). The main content area is titled 'Bernoulli MLE #745' and shows a post by 'E' (Anonymous) from 'Lectures' 17 hours ago. The post text reads: 'Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$   
▪  $X_i \sim \text{Ber}(p)$   
▪ Probability mass function,  $f(X_i = x_i | P = p)$ '. Below the text are two graphs: 'PMF of Bernoulli' (a bar chart with bars at 0 and 1) and 'PMF of Bernoulli (p = 0.2)' (a line graph showing a decreasing curve). The formulas  $f(x_i | p) = p^{x_i} (1-p)^{1-x_i}$  and  $f(x_i | p = 0.2) = 0.2^{x_i} (1-0.2)^{1-x_i}$  are shown. The post has 68 views and 1 answer. The answer is by 'Anna Mattinger TA' 4 hours ago, starting with 'Hi!' and 'If I'm following, your concern is whether you'd know the PMF of a Bernoulli offhand, or whether you'd be able to derive some hairier PMF/PDF on the fly?'. The answer is marked as correct with a green checkmark. The bottom of the page shows '198 others online'.

# 101 Course Reader Chapters

New Reader

Probability for Computer Science

probabilityforcs.firebaseio.com/book/name2age

- Continuous Distributions
  - Uniform Distribution
  - Exponential Distribution
  - Normal Distribution
  - Binomial Approximation
  - Stories
    - Many Binomial Problems
    - Winning Series
    - Approximate Counting
    - Jury Selection
    - Grading Eye Inflammation
    - Grades are Not Normal
    - Curse of Dimensionality
    - Algorithmic Art
    - Game of Ur
- Part 4: Probabilistic Models
  - Joint Probability
  - Marginalization
  - Multinomial
  - Continuous Joint
  - Inference
  - Bayesian Networks
  - Independence in Variables
  - Correlation
  - General Inference
  - Name to Age
  - Stories
    - Fairness in Artificial Intelligence
    - Federalist Paper Authorship
    - Probability of Baby Delivery
    - Bayesian Carbon Dating
    - Pyramid Hidden Chambers
    - Digital Vision Test
    - Bridge Distribution

## Name to Age

Because of shifting patterns in name popularity, a person's name is a hint as to their age. The United States publishes a data which contains counts of how many US residents were born with a given name in a given year, based off Social Security applications. We can use inference to compute the reverse probability distribution: an updated belief in a person's age, given their name. As a reminder, if I know the year someone was born, I can calculate their age within one year.

Query Name: Katherine ✓

Age	Probability
11	0.006
16	0.007
21	0.010
26	0.013
31	0.015
36	0.019
41	0.017
46	0.013
51	0.008
56	0.009
61	0.008
66	0.010
71	0.012
76	0.010
81	0.006
86	0.004
91	0.004
96	0.005
101	0.007
106	0.008
111	0.006

Records with name: 589753

This demo is based on real data from US Social Security applications between 1914 and 2014. Thank you to <https://www.kaggle.com/kaggle/us-baby-names> for compiling the data.

# 122 Personal Challenges



# 1 New Counter



Incredible time. Incredible  
school at which to study  
probability

thank you

A row of ten light-colored wooden blocks, each with a single lowercase letter in black font, spelling out the words 'thank you'. The blocks are arranged on a wooden surface. The background is a soft-focus bokeh of warm, golden-yellow lights, creating a warm and appreciative atmosphere.