



# Beyond CS109

Chris Piech  
CS109, Stanford University

# Class Theme 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	---	---	---	---	0.241
5	September - Earth, Wind & Fire			55	---	---	---	---	0.180
6	Time of Our Lives - Pitbull			24	---	---	---	---	0.224
7	Vienna - Billy Joel			24	---	---	---	---	0.235
8	Just the Two of Us (feat. Bill Withers) - Grover Washing			25	---	---	---	---	0.180
9	Voulez-Vous - ABBA			20	---	---	---	---	0.203
10	Let it Happen - Tame			22	---	---	---	---	0.214
11	Careless Whisper - George Michael			24	---	---	---	---	0.175
12	Take Five - Dave Brubeck			18	---	---	---	---	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.

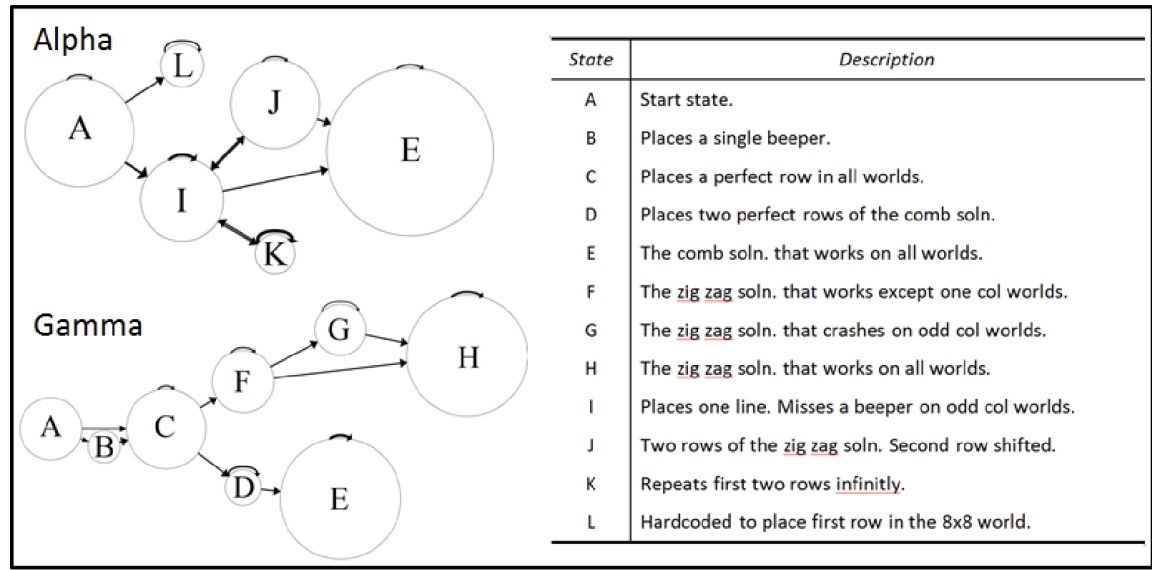
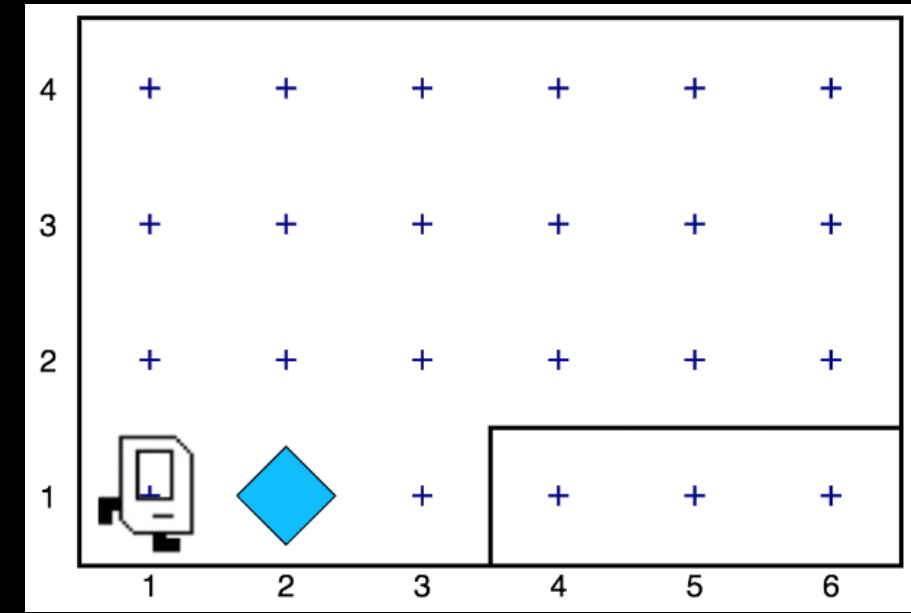
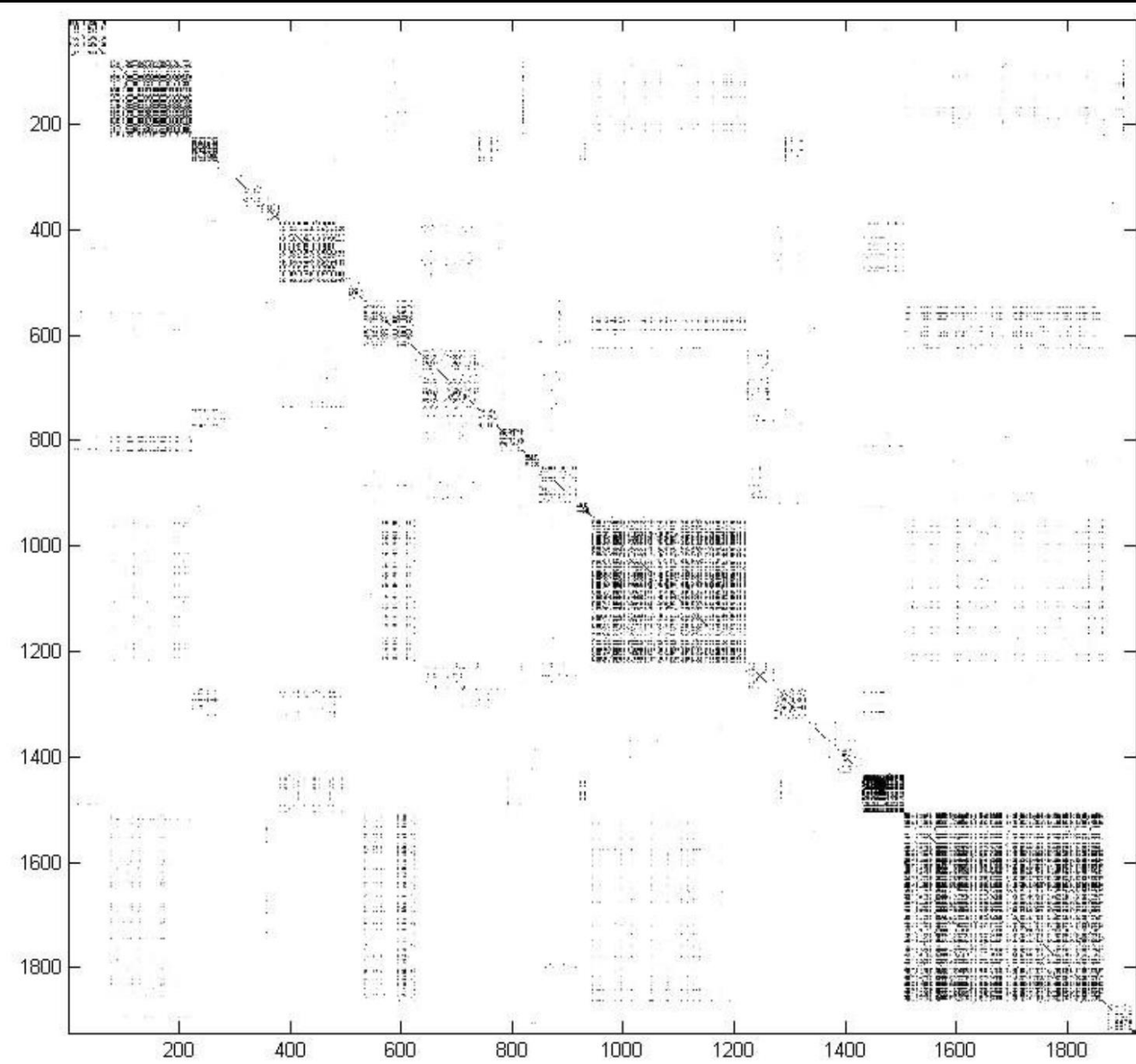
Algo for top 16?

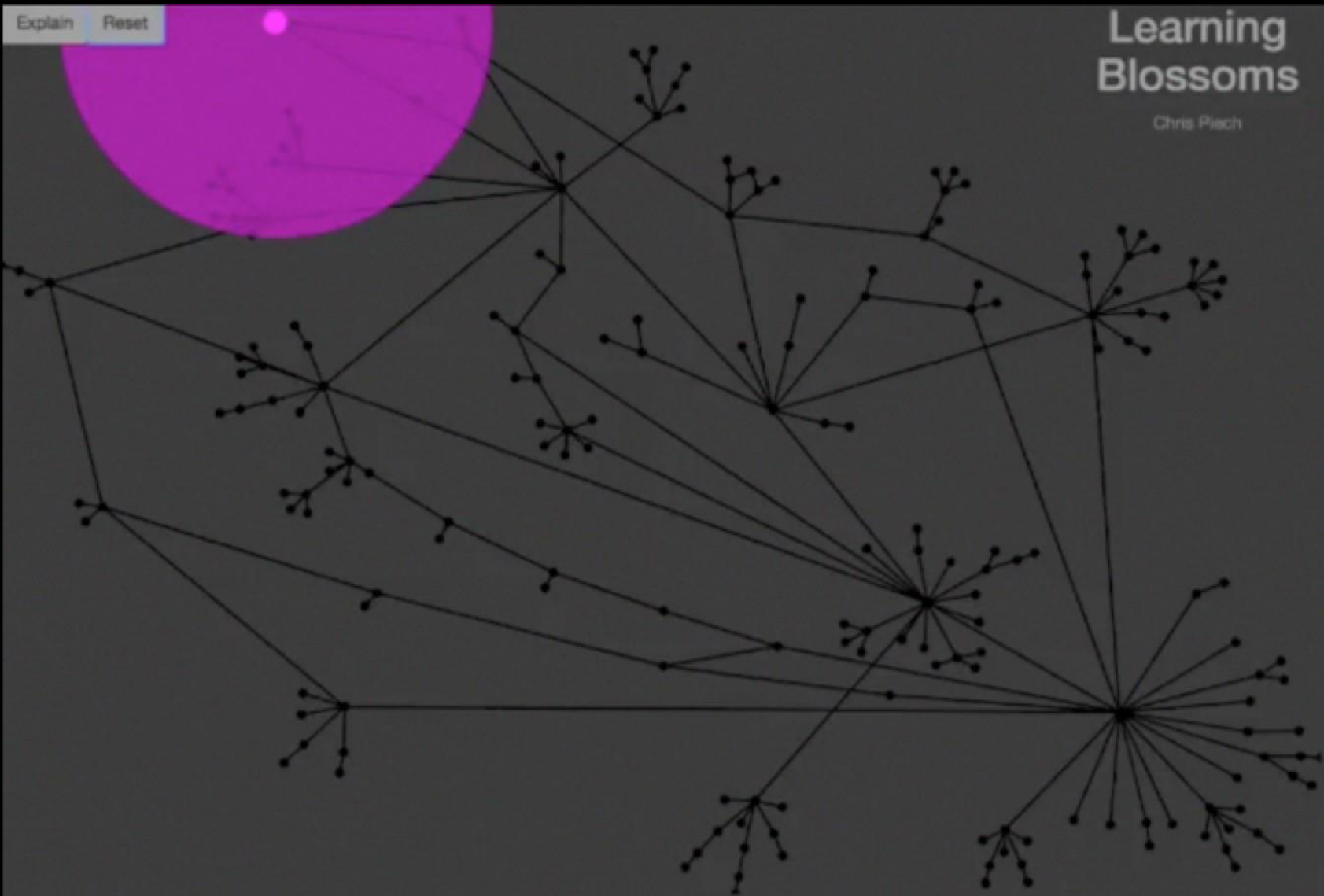
This quarter

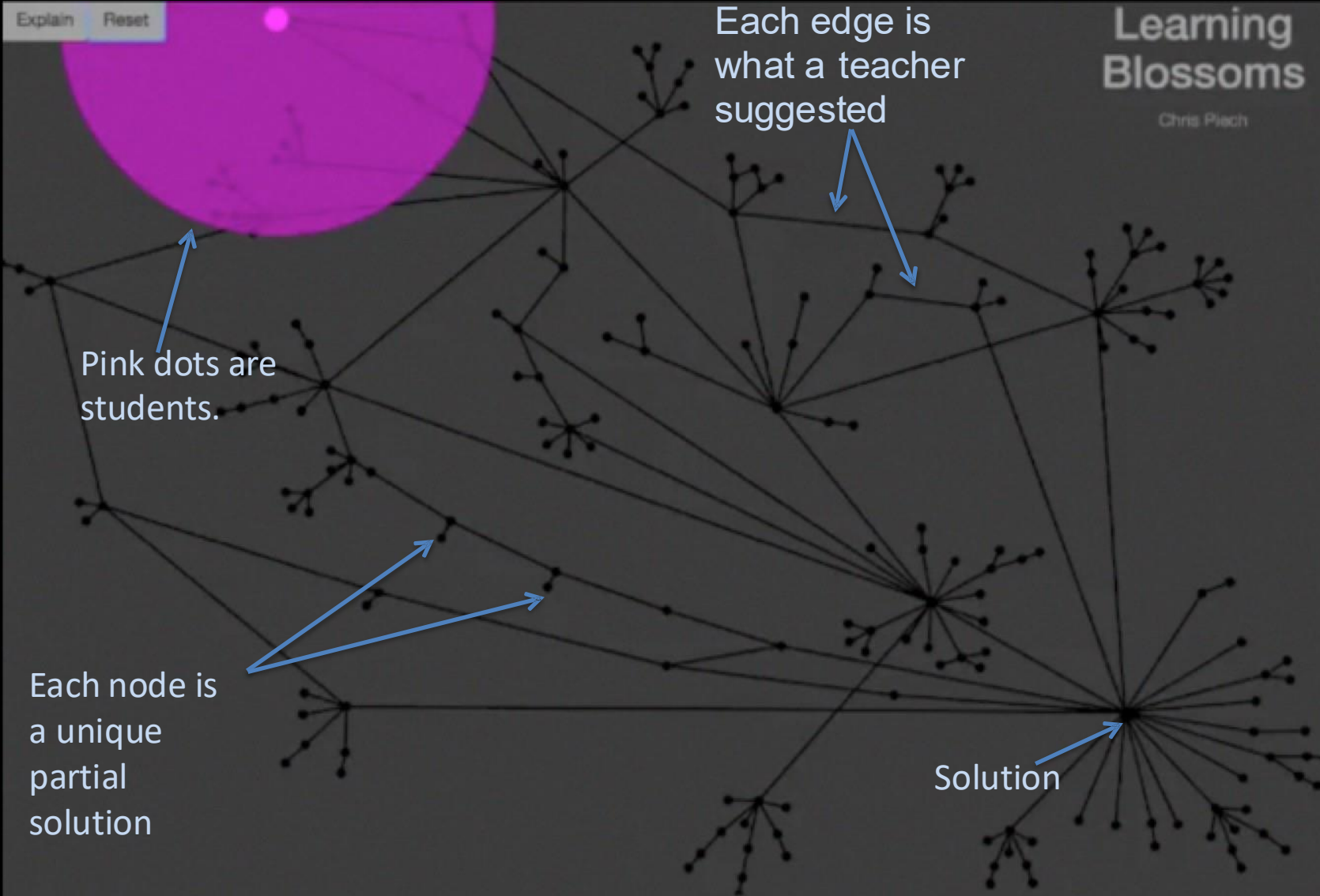
1	Sereia - Mundo Azul	
2	Smart - Le Sserafim	
3	Viva La Vida - Coldplay	
4	Take Me Home, Country Roads - John Denver	
5	Real Love Baby - Father John Misty	
6	Sports car - Tate Mcree	
7	Something Just Like This - The Chainsmokers	
8	My Shot - Lin-Manuel Miranda	
9	Young Folks - Peter Bjorn and John	
10	Don't Let Me Down - The Beatles	
11	Feel Again - With Heartbeats - OneRepublic :D	
12	Parce que je t'aime - Elodie Frege	
13	Golden - KPOP Demon Hunters	
14	Oghniat Al Wadaa - Fairuz	
15	Tokyo Daylight - Lyn	
16	Tampa - Filho do Zua	

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

Probability gives you a new  
lens on the world



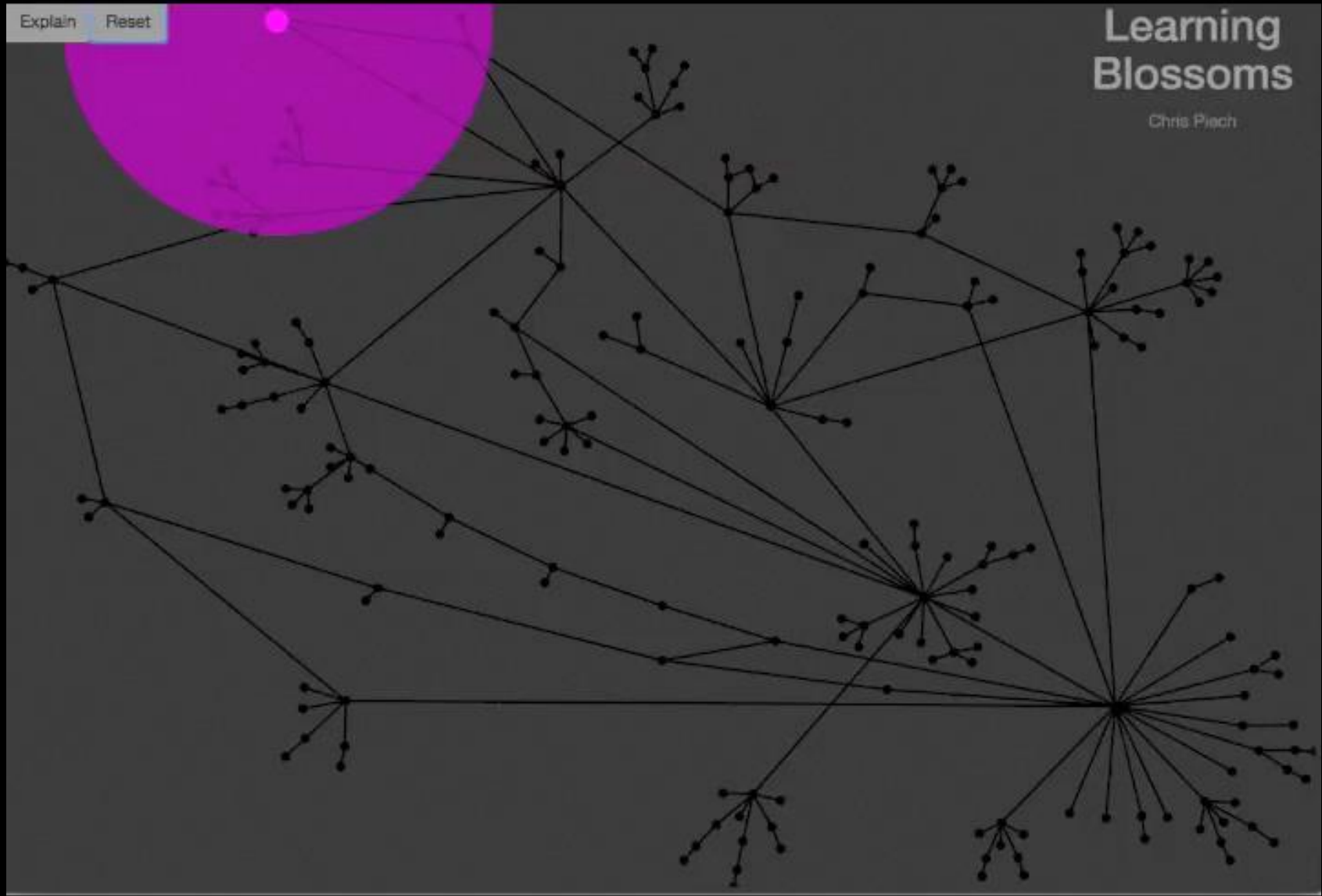




Explain Reset

# 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 ↻ ▼

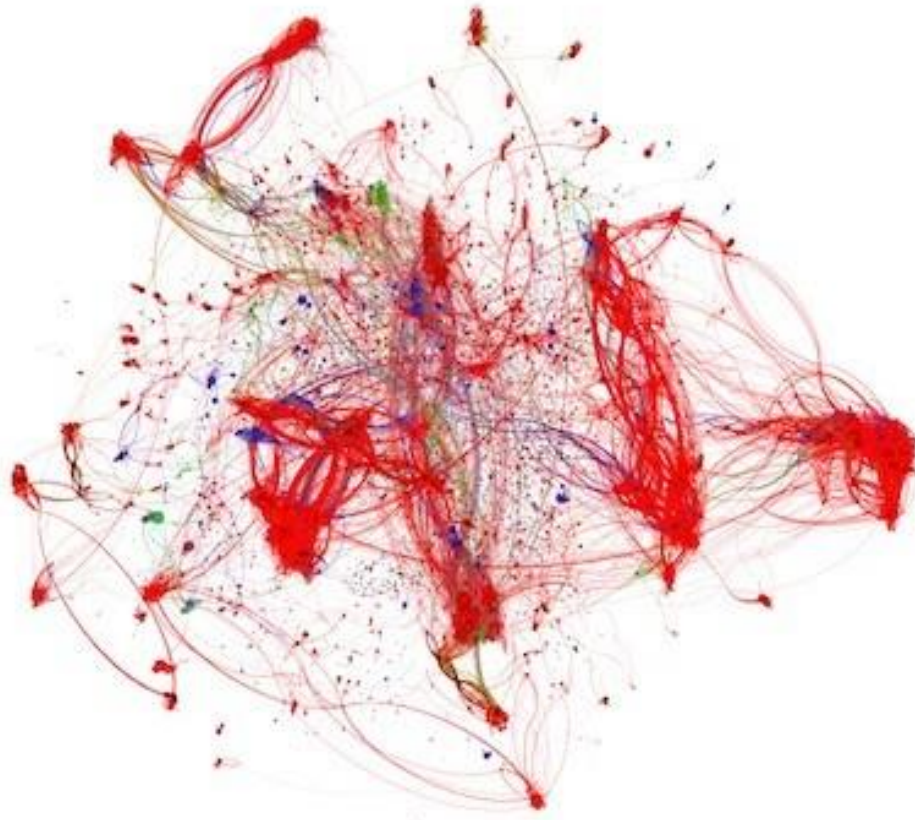
move forward

12%



# Hard Problem!

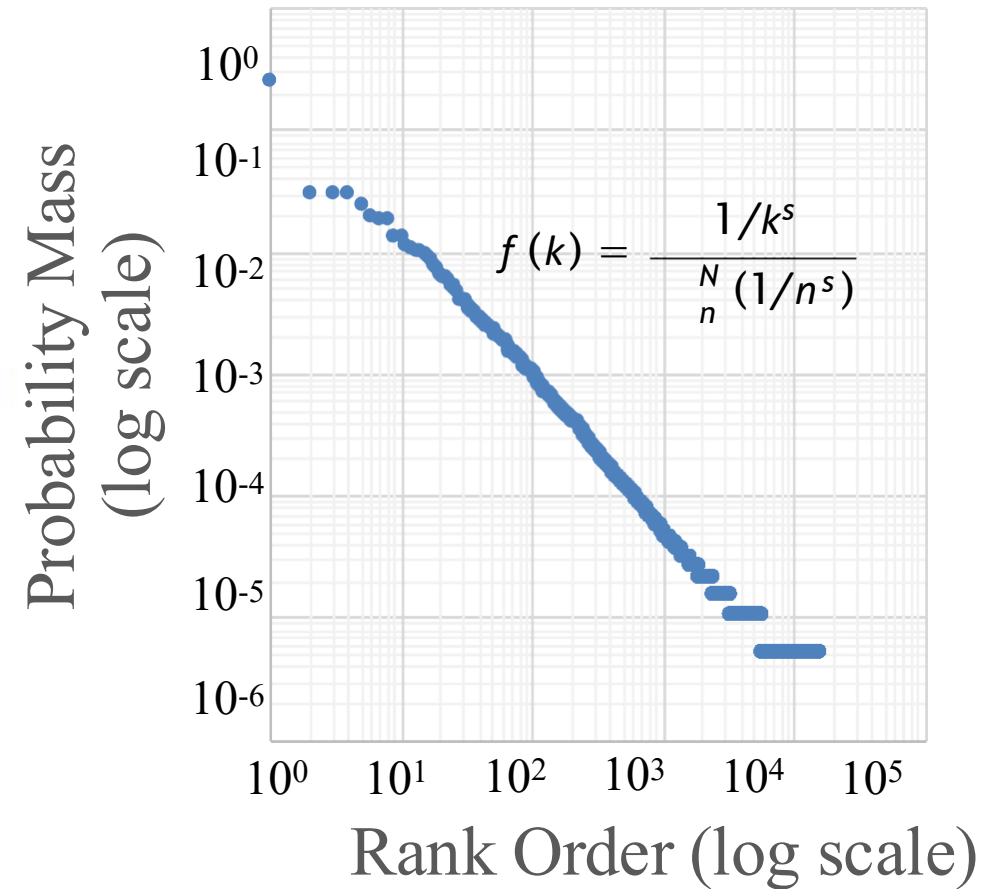
Brute force solution?



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.

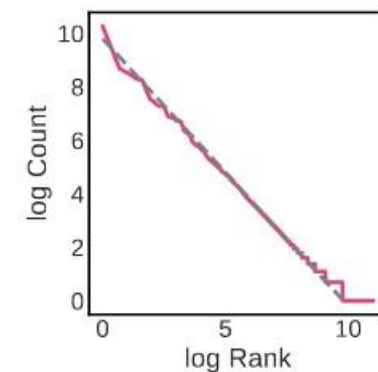
```
public void run() {
    for (int i=START; i>0; i--){
        println(i);
    }
    println("Liftoff");
}
```

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

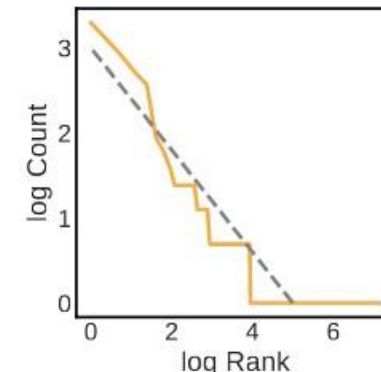
PyramidSnapshot

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

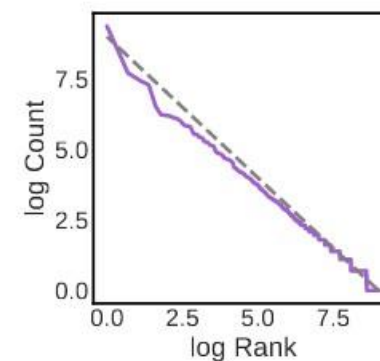
(b) Code.org P8



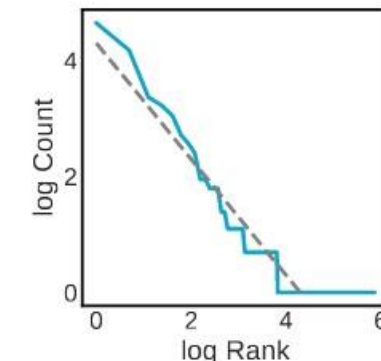
(c) CS1: Liftoff



(d) Pyramid



(e) Powergrading



# Original deep learning for education paper

KHAN  
Student



1

10

Exercise index

Exercise Type:

Answer:



Solving for x-intercept



Correct



Solving for y-intercept

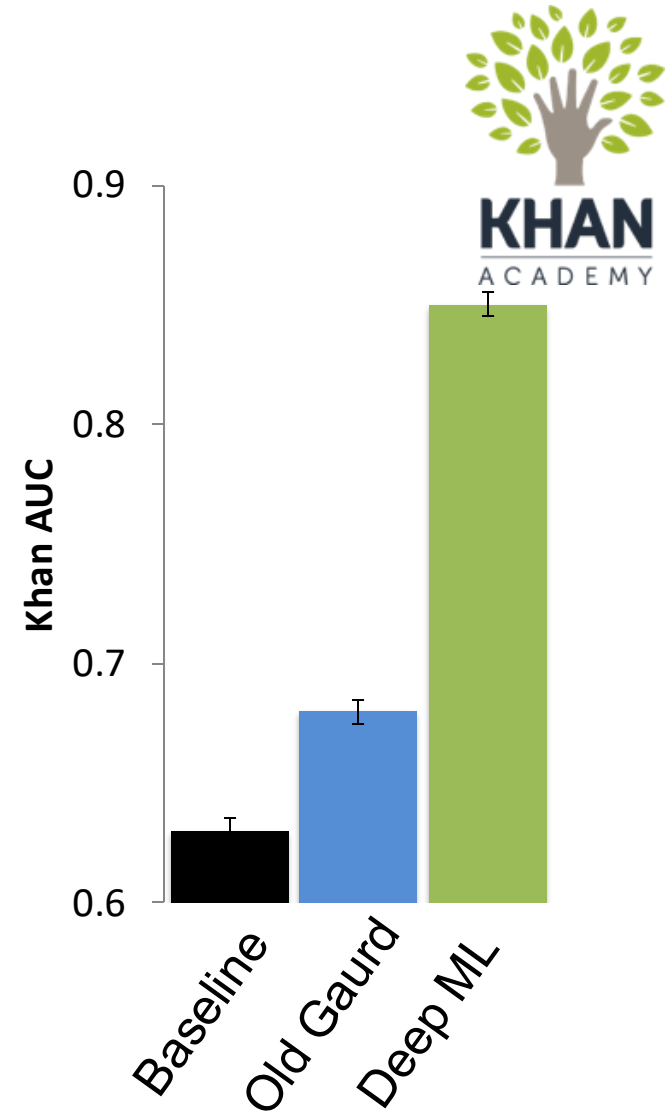


Incorrect

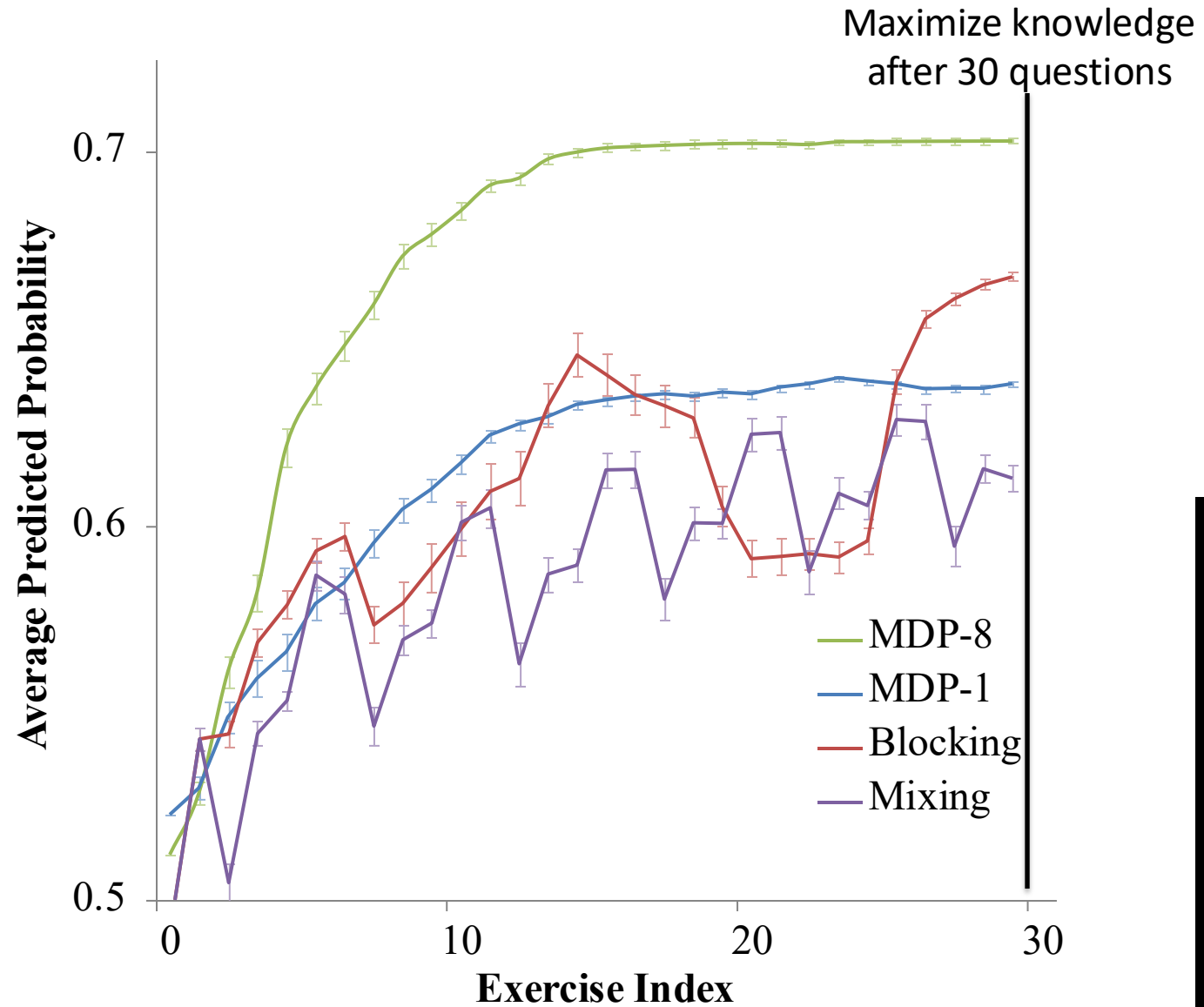


Square roots

Powers DuoLingo Bird Brain (as of 2023)



# Optimal Teaching

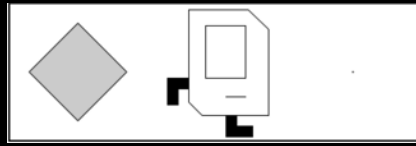


Used in DuoLingo  
Bird Brain (last  
checked in 2023)

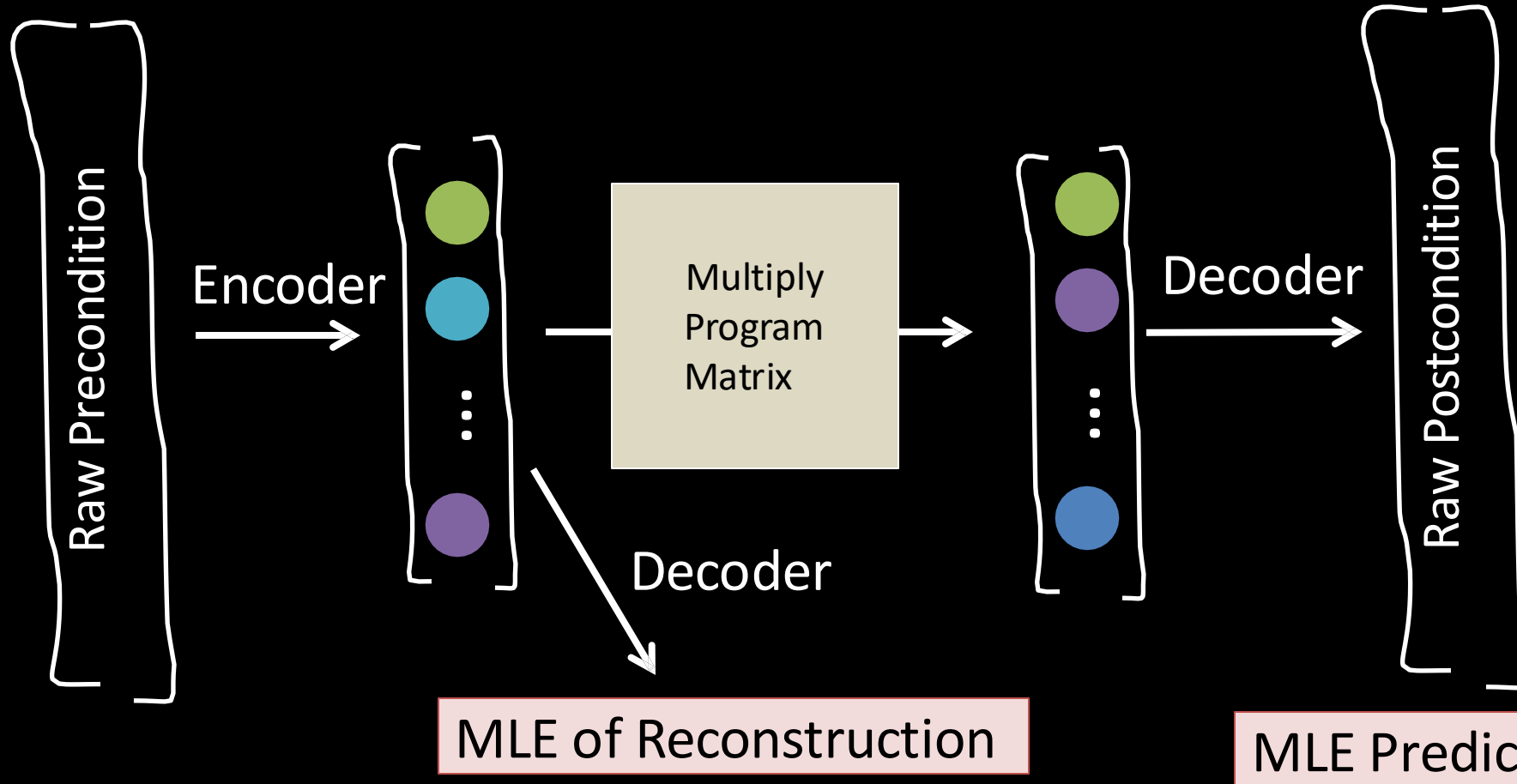
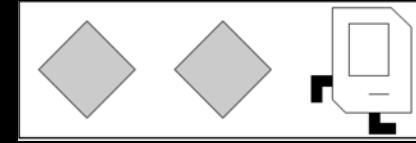


Back to coding!

# Neural Network for Programs



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

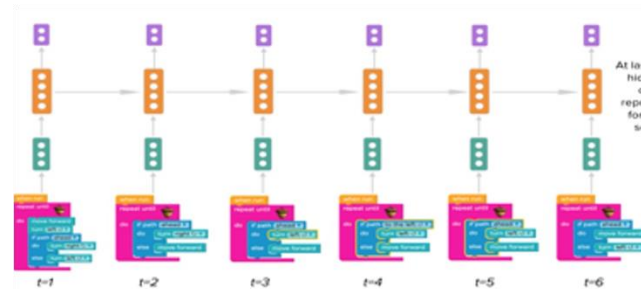
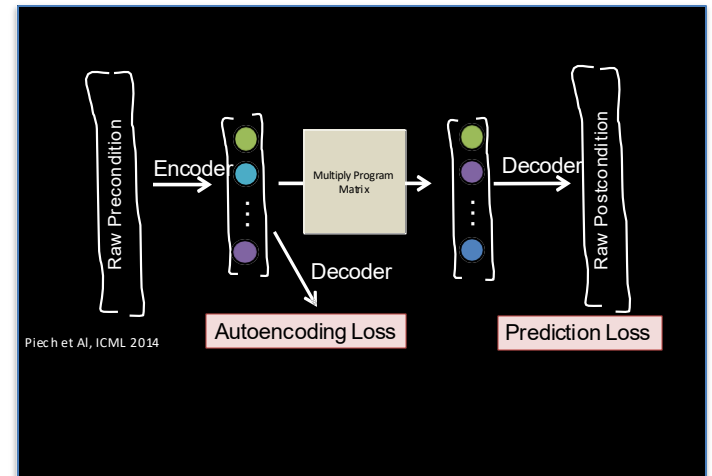
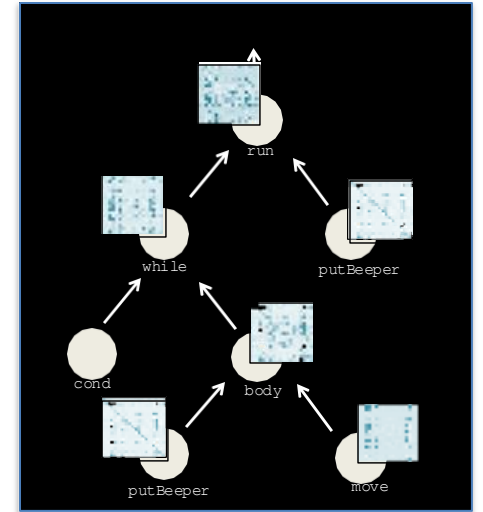
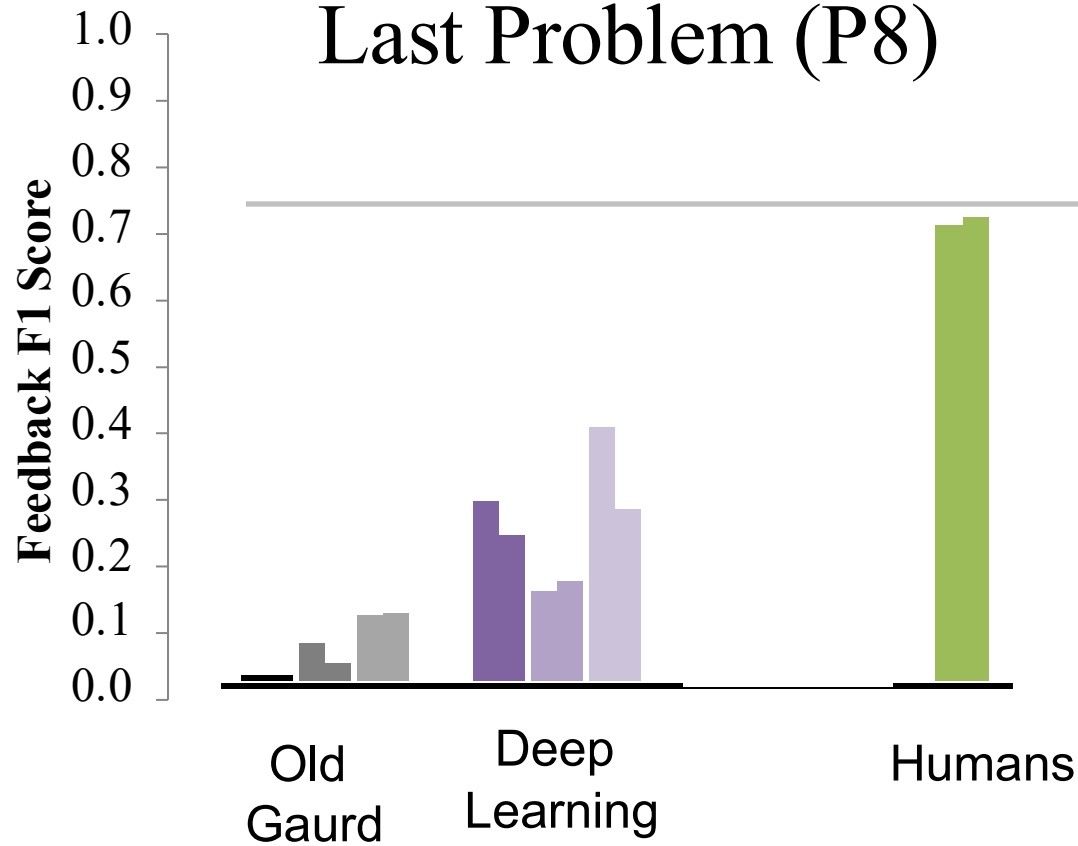


\*coded pre-tensor flow



# Inaccurate, Uninterpretable, and Data Hungry

Label student code



# Humans Don't Need Much Data

Single training example:

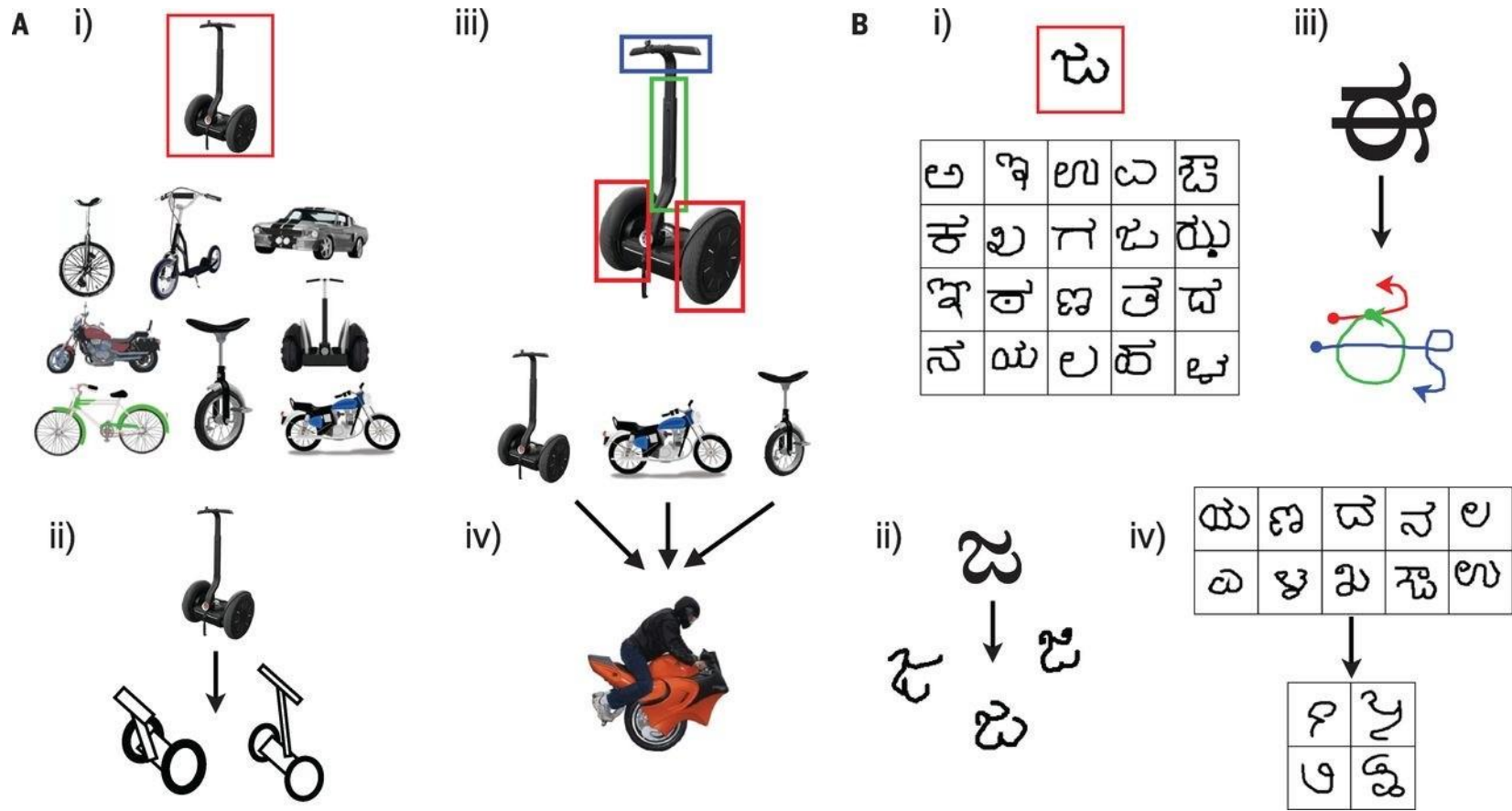
৩

Test set:

৩ ৩ ৩  
৩ ৩ ৩  
৩ ৩ ৩



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



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



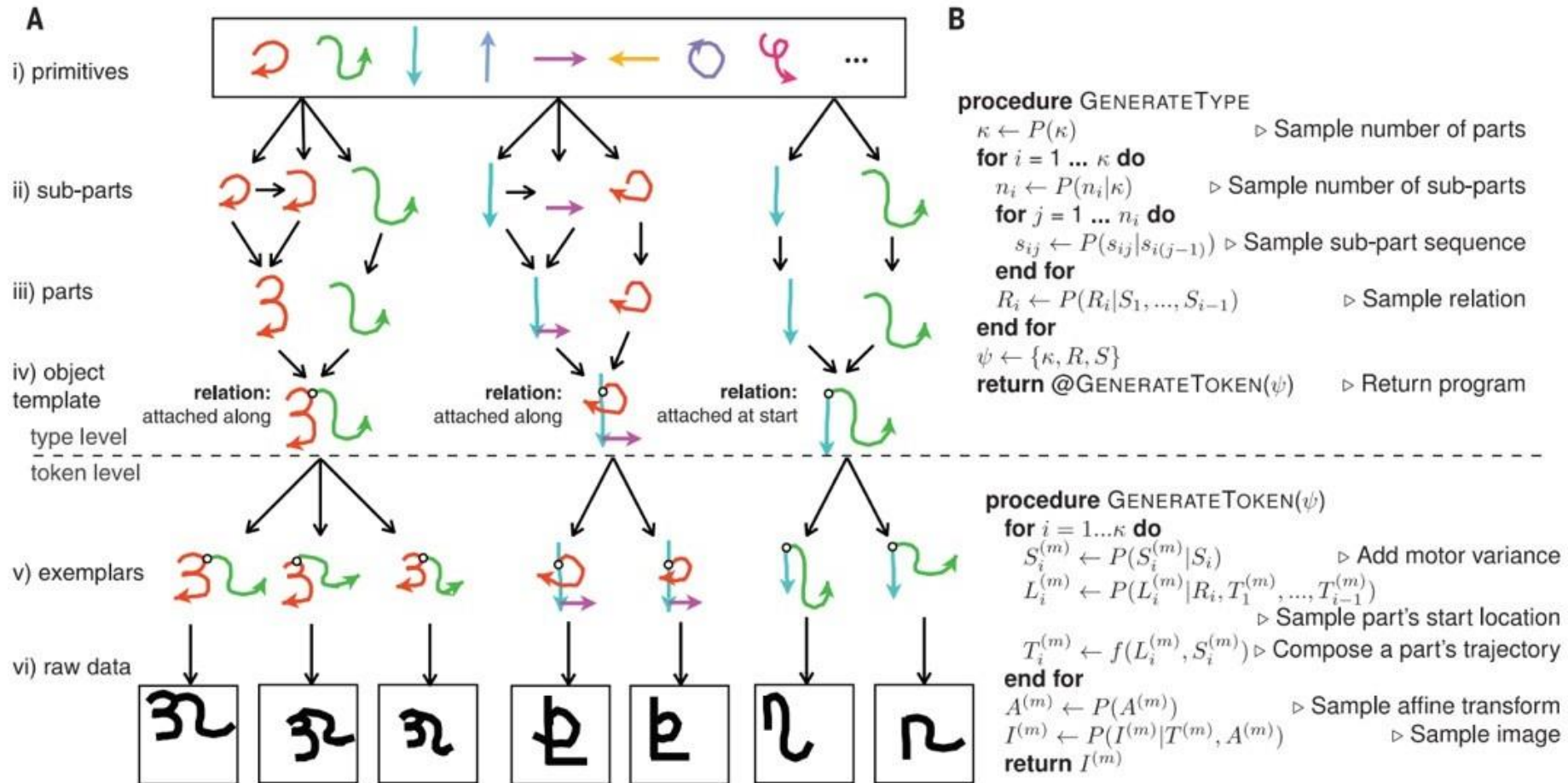
Fig. 2 Simple visual concepts for comparing human and machine learning.



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

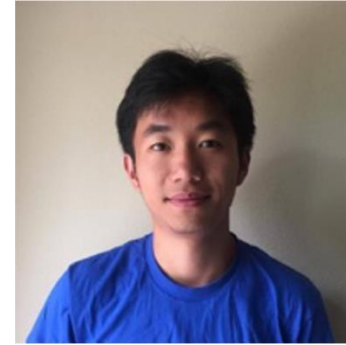
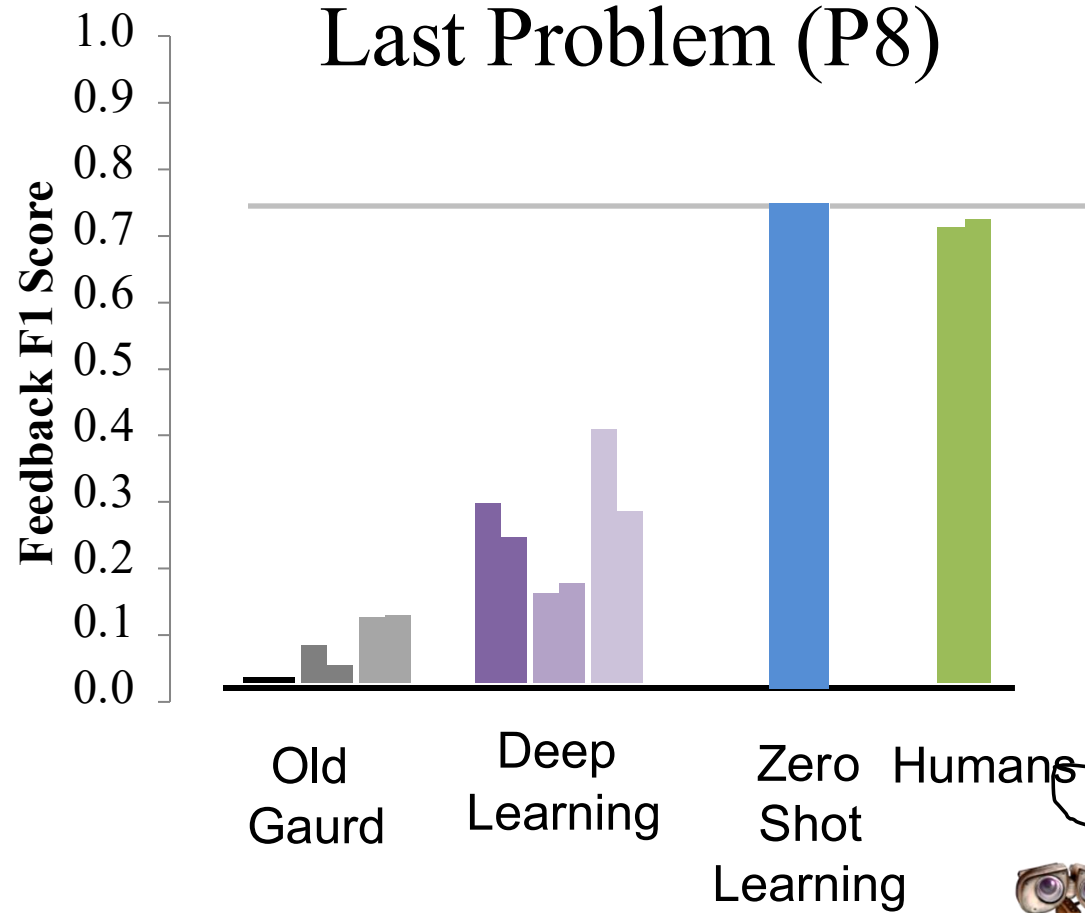


# Bayesian Program Learning

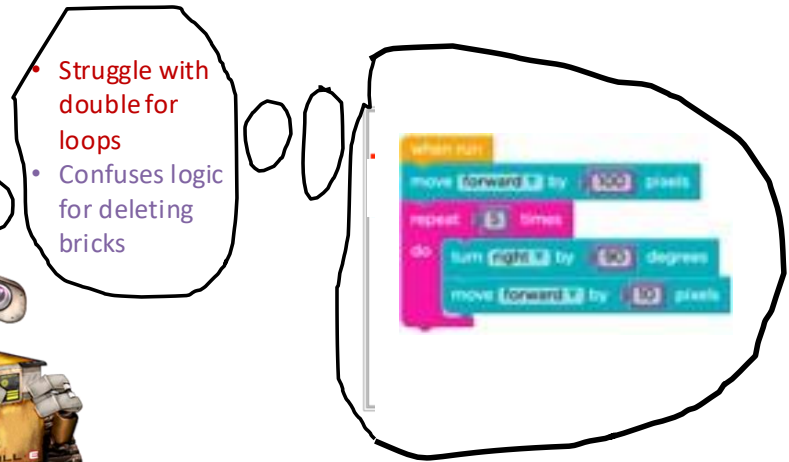


# Generative Understanding

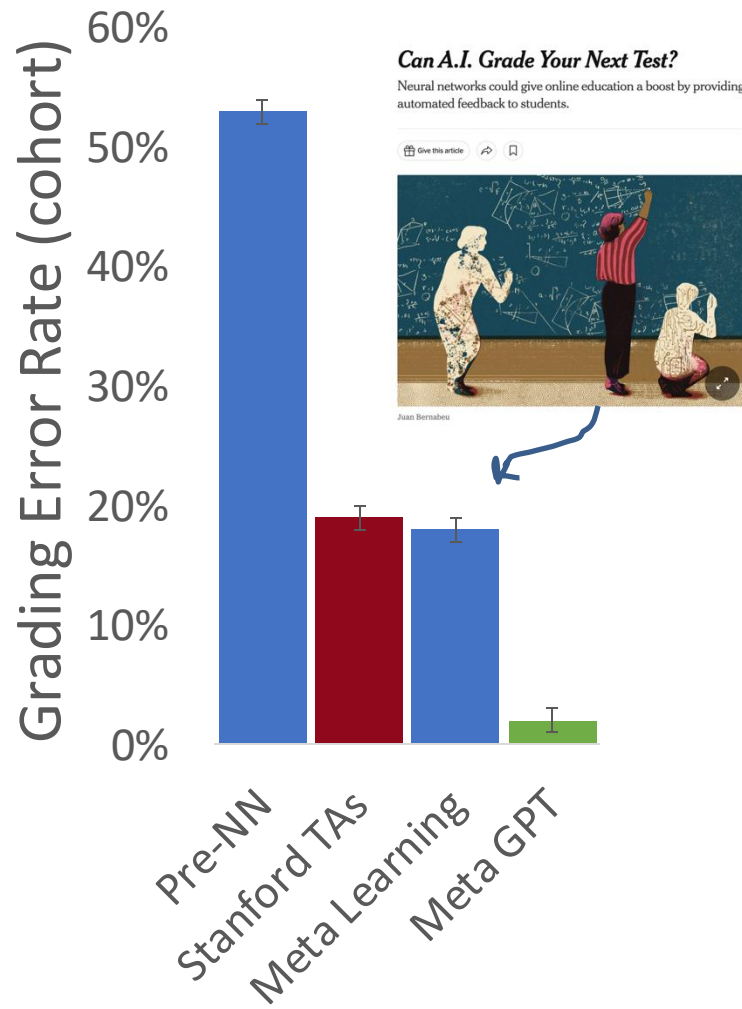
Label student code



*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

**60,000+** 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.

```

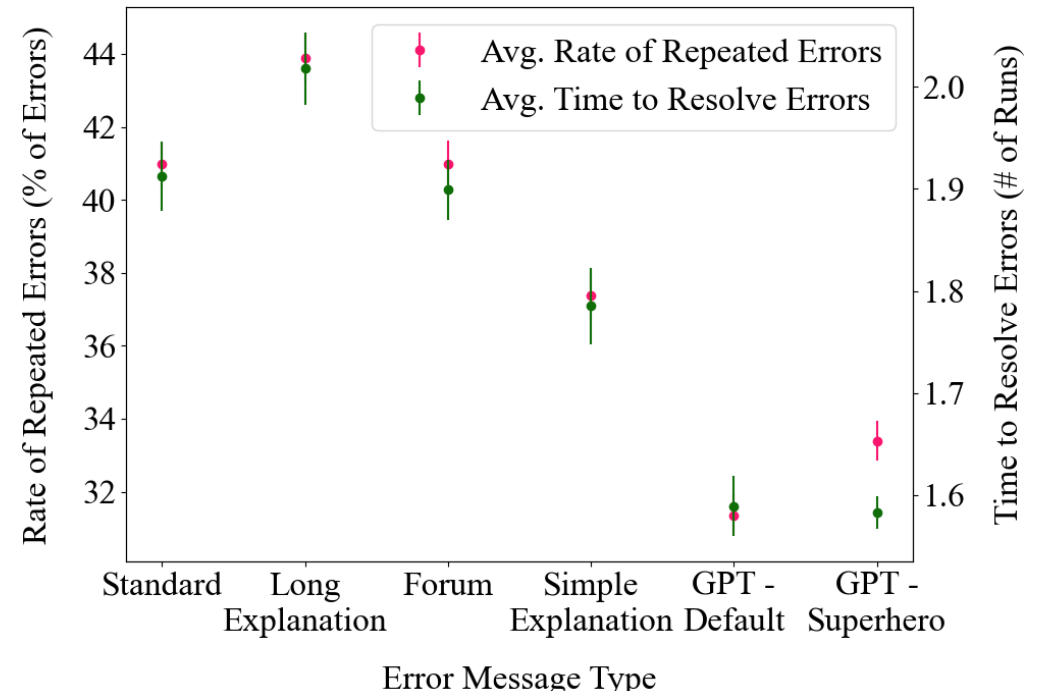
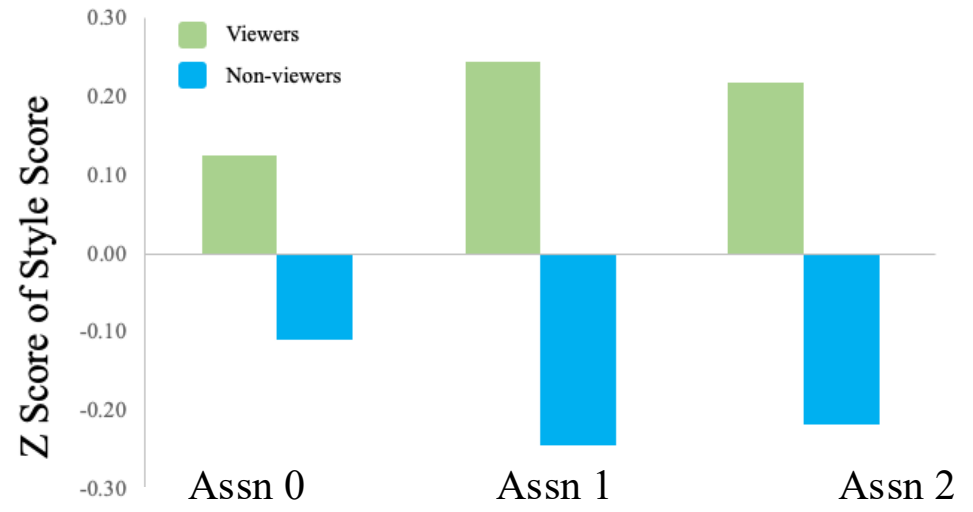
1 import graphics
2 import time
3 import random
4 import math
5
6 canvas_width = 500
7 canvas_height = 600
8 paddle_y = canvas_height - 30
9 paddle_width = 80
10 paddle_height = 15
11 ball_radius = 10
12
13 brick_gap = 5
14 brick_width = (canvas_width-brick_gap*9) / 10
15 brick_height = 10
16
17
18
19 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)
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
    
```

**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 "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>

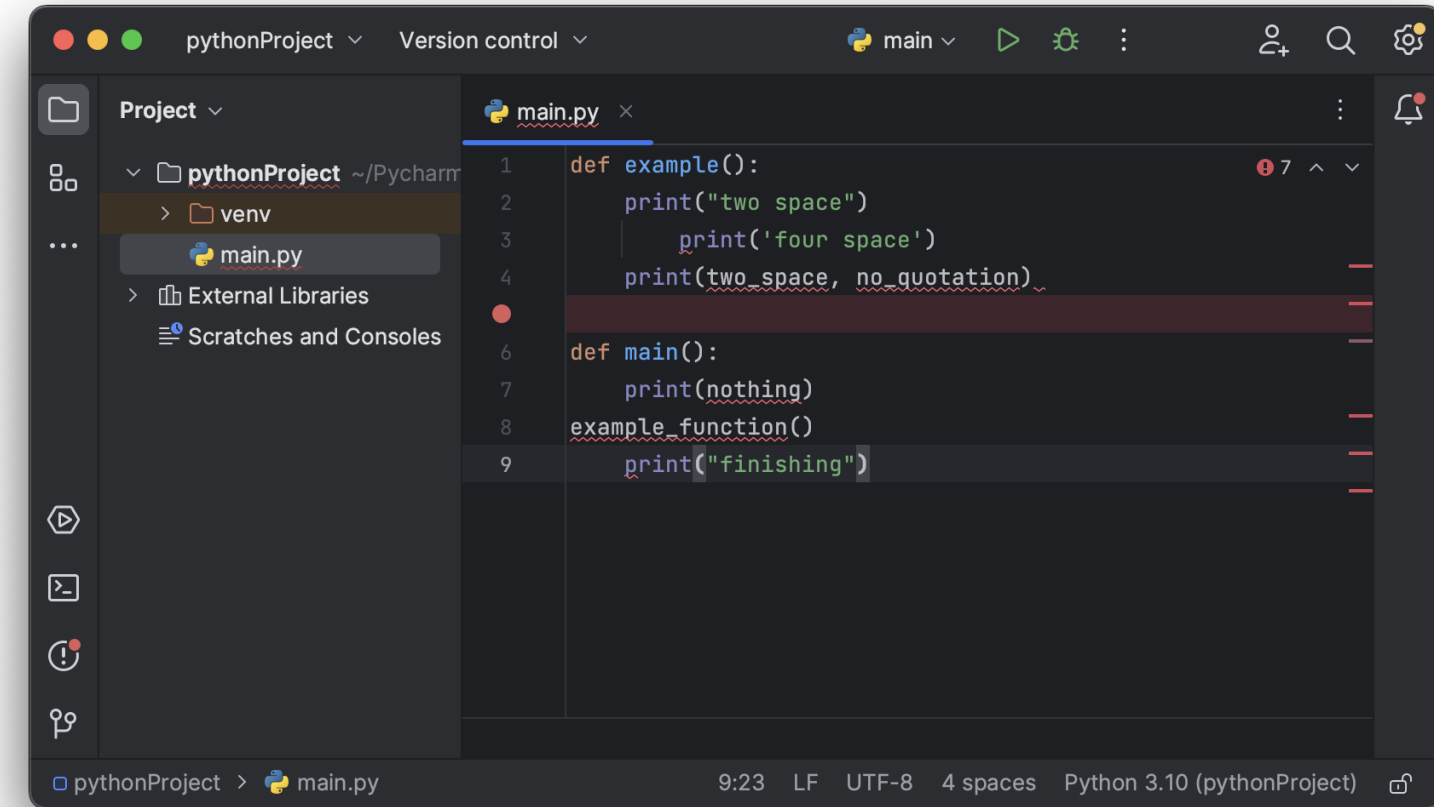
On the right side of the interface, there is a video player showing a colorful grid of bricks (red, orange, yellow, green, blue) and a black ball. Below the video are buttons for "Video", "Code", and "Demo".

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

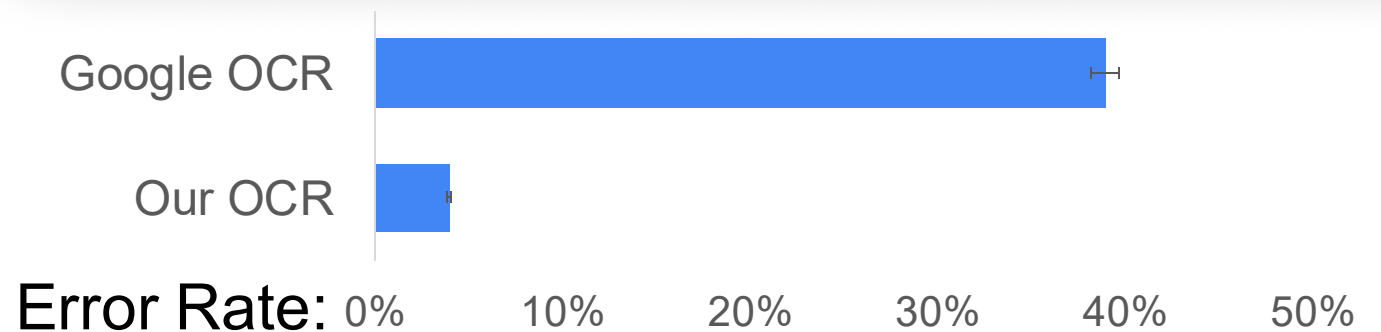
```
def main():  
    → print(rinting)
```

```
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")
```



# Information Theoretic Learning!

youtube.com/watch?v=z2n4e\_oS5jU

Search

Acrobat File Edit View E-Sign Window Help

Home Tools Infolingo\_An\_Inf... x

Table 1. Performance comparison for various vocabulary picking methods with different target percentages using no prior vocabulary knowledge. The accuracy using 0% and 100% of the words is  $6.10\% \pm 0.76$  and  $74.70\% \pm 1.38$ , respectively. The F1-score using 0% and 100% of the words is  $7.80\% \pm 0.82$  and  $83.08\% \pm 1.08$ , respectively.

Target %	10%		25%		50%		75%	
	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)
Random	$10.30 \pm 0.96$	$16.03 \pm 1.06$	$15.60 \pm 1.15$	$25.09 \pm 1.23$	$29.10 \pm 1.44$	$45.80 \pm 1.38$	$45.30 \pm 1.57$	$63.42 \pm 1.33$
Frequent	$11.80 \pm 1.02$	$21.25 \pm 1.14$	$22.60 \pm 1.32$	$34.76 \pm 1.35$	$42.70 \pm 1.56$	$53.30 \pm 1.46$	$60.10 \pm 1.55$	$69.18 \pm 1.36$
Divergence	$15.60 \pm 1.15$	$25.03 \pm 1.23$	$28.40 \pm 1.43$	$41.38 \pm 1.40$	$49.40 \pm 1.58$	$63.95 \pm 1.36$	$69.00 \pm 1.46$	$78.43 \pm 1.19$
Entropy	$16.20 \pm 1.17$	$25.85 \pm 1.24$	$28.80 \pm 1.43$	$41.80 \pm 1.41$	$49.70 \pm 1.58$	$63.09 \pm 1.38$	$69.10 \pm 1.46$	$77.72 \pm 1.21$

Accuracy (%)

Target Percentage of New Vocabulary

Figure 1. Accuracies Using Four Vocabulary Picking Methods.

still produce different vocabularies, as demonstrated in the caption of Figure 2.

**Stemming:** This study treats different forms of the same word, such as "cat" vs. "cats," as distinct. However, a learner with some prior grammar rules would probably suffice to learn the word's stem. Thus, one improvement is to consider only the word stems when selecting vocabulary.

**Incorporating Prior Knowledge:** We can incorporate prior vocabulary knowledge when selecting new vocabulary for intermediate learners at different learning stages. One approach is to leverage the A1-C2 language proficiency scale. Intermediate learners can let the learner self-select their level and assumed vocabulary for each level from consistent sources. Another approach is assigning each word a probability based on the learner's current level. Presenting the uncertainty a learner already knows about a word can help them use inference to update their belief about the word's meaning based on direct input. A1-C2 level, and text comprehension. For example, a known word or a correct guess for a given sentence may increase the belief in understanding.

Infilingo - An Information-Theoretic Approach to Language Learning

Unlisted

Alice Heiman  
1.01K subscribers

Subscribe

1

Share

Download

Clip

All For you Recently uploaded Watched

The 4 things it takes to be an expert  
Veritasium  
12M views · 2 years ago

The Blind Mathematician Who Became the World's Greatest  
Newthink  
406K views · 2 months ago

Lethal Gene Diseases: No More Confusion! (Easy Memorization...  
SOLVE  
40 views · 9 hours ago

Why NBA Players STILL FEAR Trash Talking Steph Curry  
Hoop Reports  
834K views · 1 year ago

Why the US has birthright citizenship  
Vox  
1M views · 6 days ago

The Riemann Hypothesis, Explained  
Quanta Magazine  
5.8M views · 4 years ago

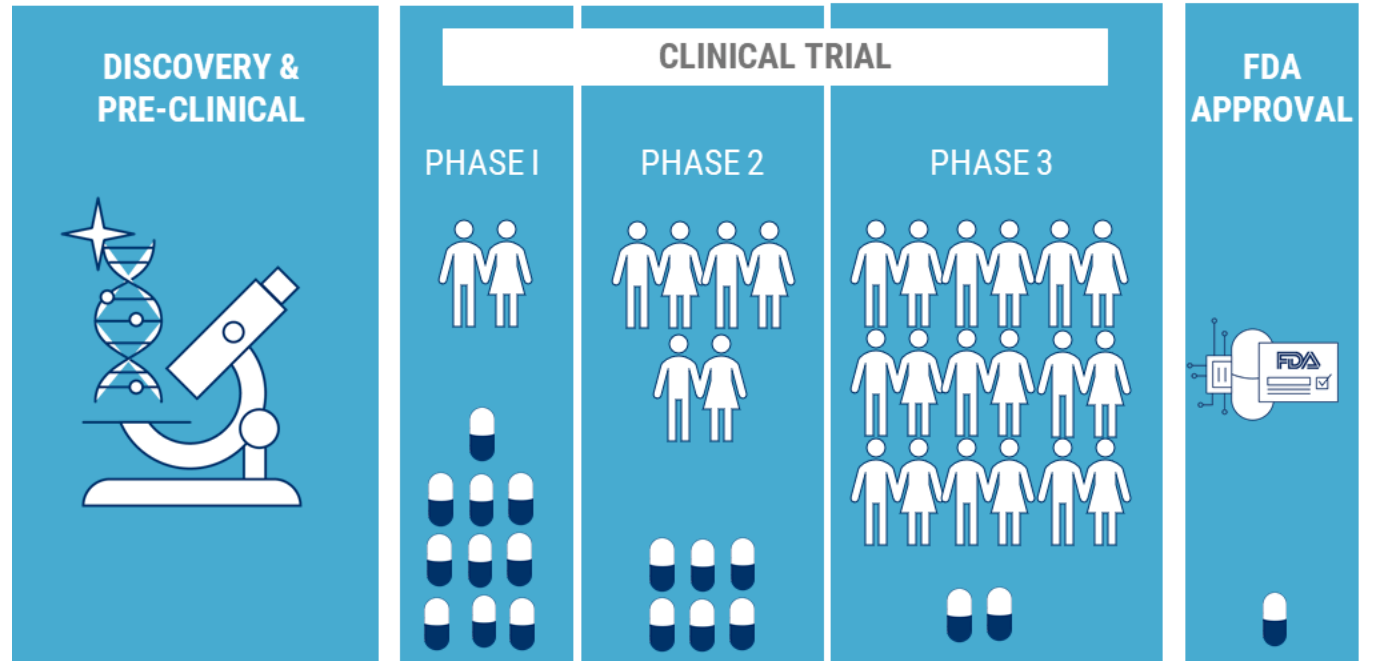
ATTEMPT NO LANDING HERE  
We Might Find Alien Life In 2200 Days  
Veritasium  
9.6M views · 5 months ago

proof by just look at it!

More than Education

# More than education

 **Bringing a drug to market is a drawn-out process**

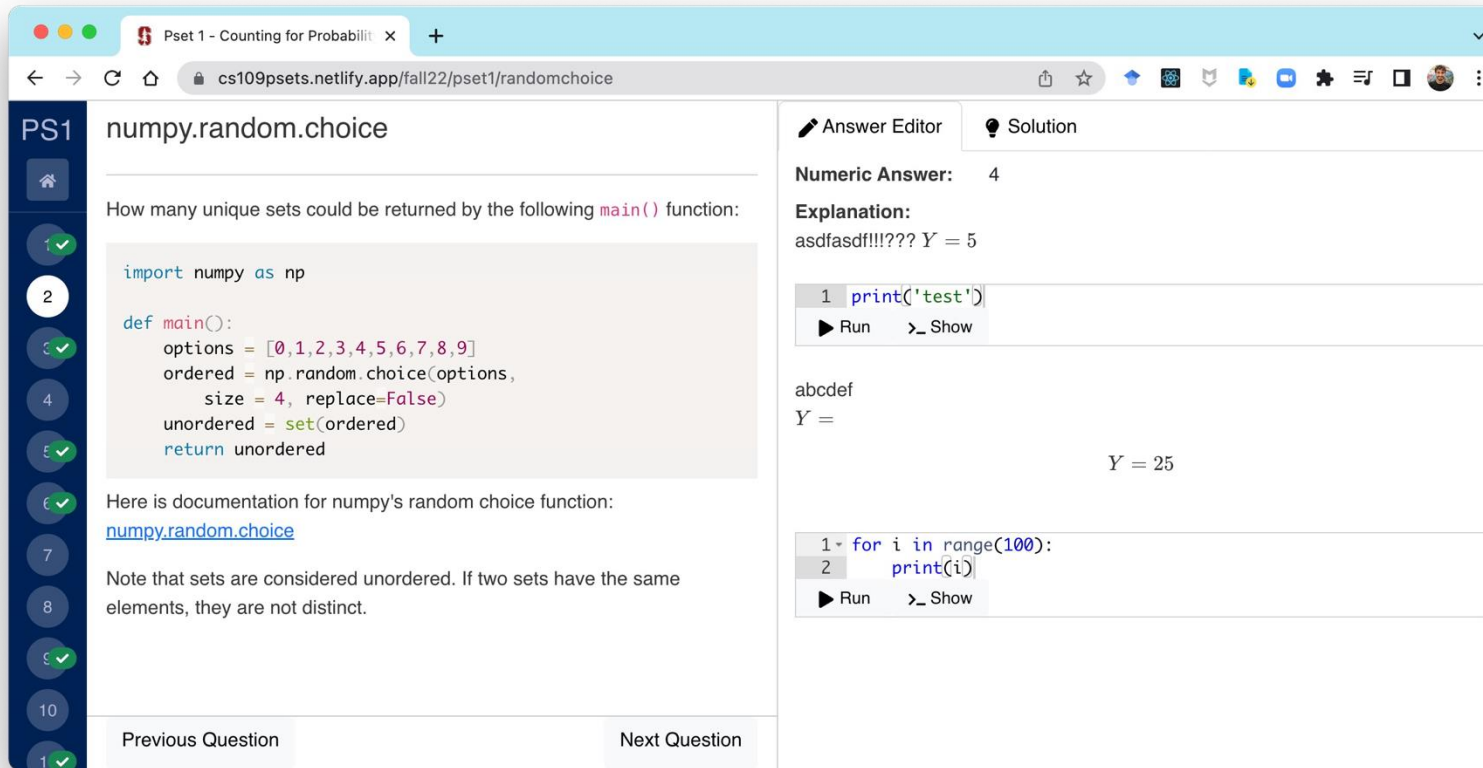


Source: cbinsights.com

 CBINSIGHTS



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



PS1 numpy.random.choice

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
```

Here is documentation for numpy's random choice function:  
[numpy.random.choice](#)

Note that sets are considered unordered. If two sets have the same elements, they are not distinct.

Answer Editor Solution

Numeric Answer: 4

Explanation:  
asdfasdf!!!!?? Y = 5

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

Run Show

abcdef  
Y =

Y = 25

```
1- for i in range(100):
2   print(i)
```

Run Show

Previous Question Next Question

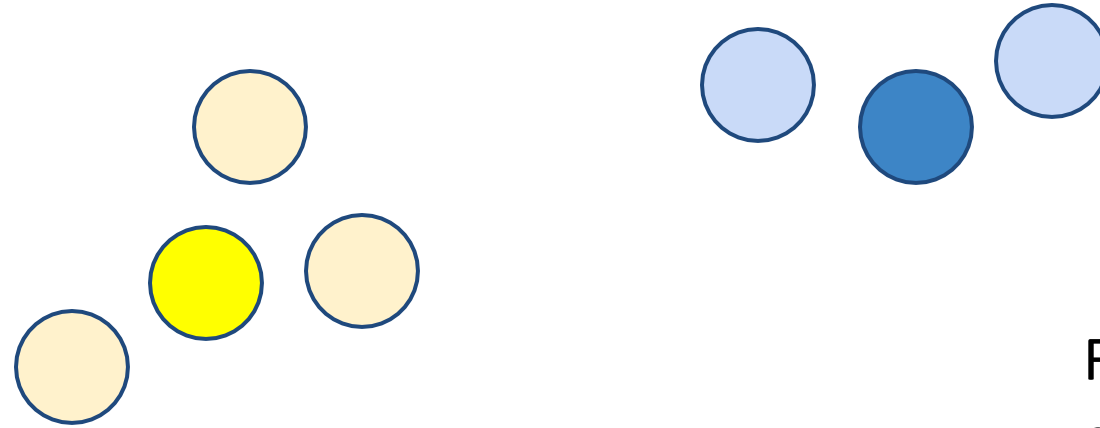
300 solutions

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



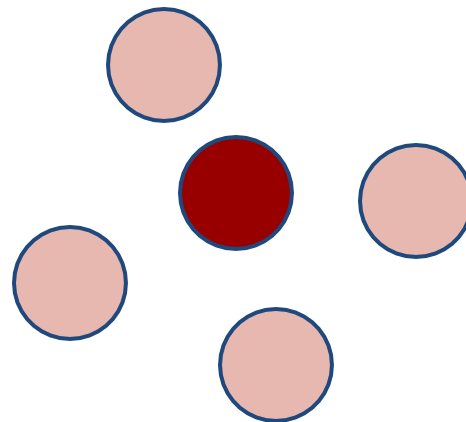
For any two points you can query the distance between them. May be non-Euclidean

Before

After

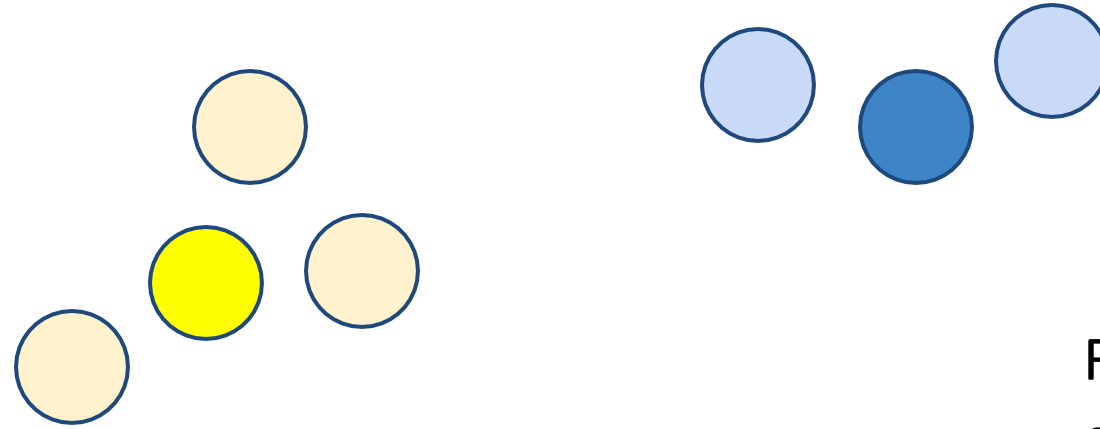
$$O(n^2)$$

$$O(n \log n)$$



# 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



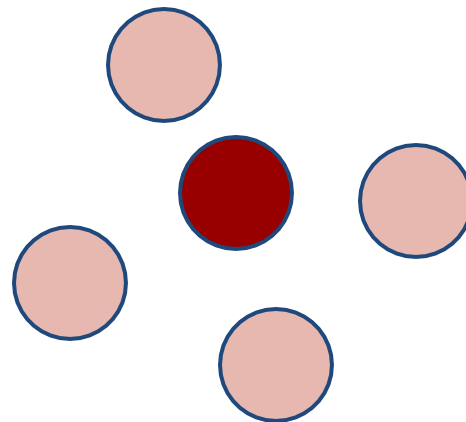
For any two points you can query the distance between them. May be non-Euclidean

Before

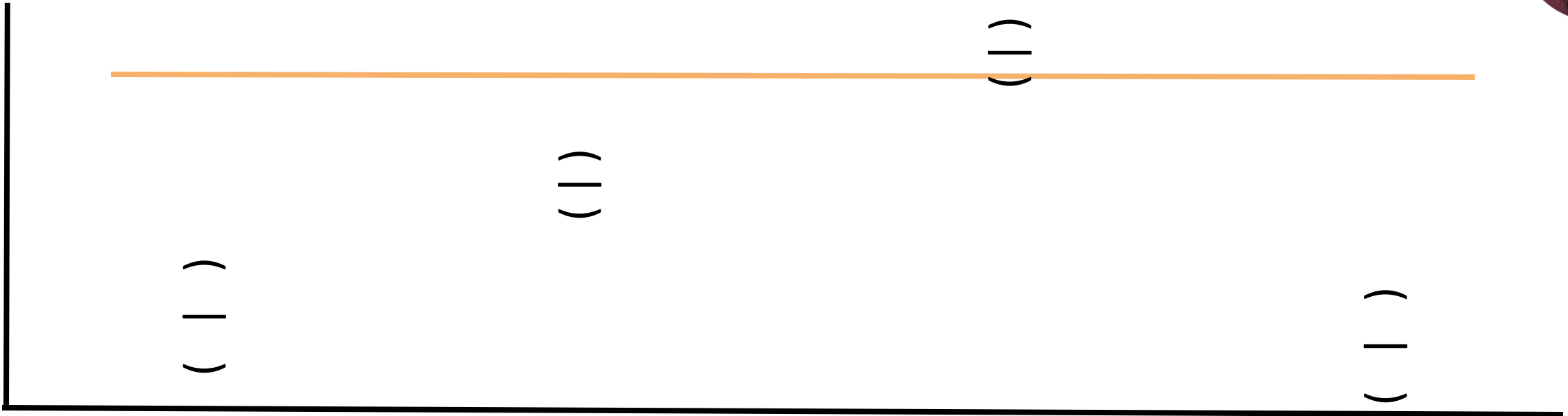
After

$$O(n^2)$$

$$O(n \log n)$$



# 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"*, NeurIPS 2023

CS109 TA



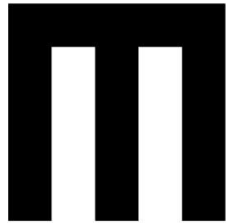
People just miss the random variables

# More than education

Vision Test

myeyes.ai/measure

## Left Eye



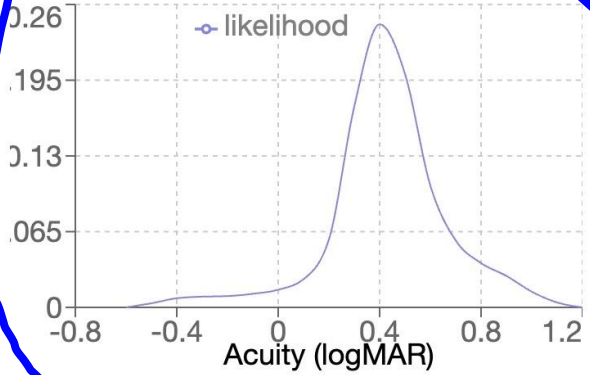
Featured in  
**THE LANCET**

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??

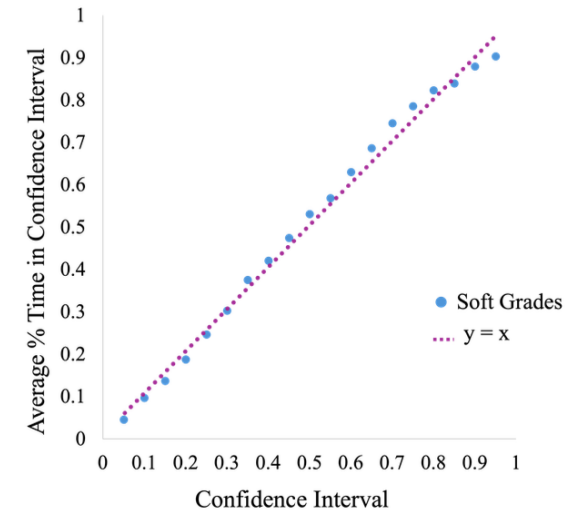
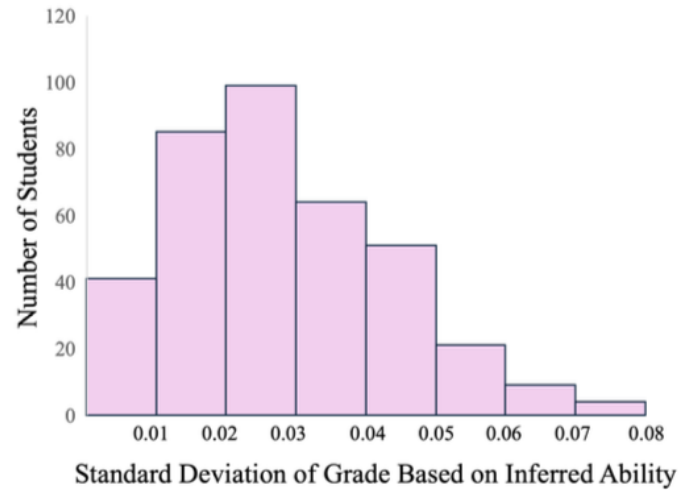
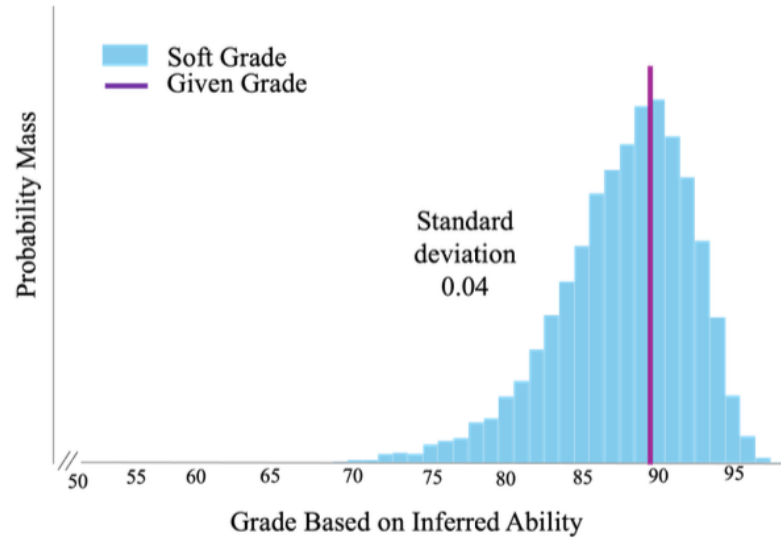
What else should be a **random variable**?

Juliette 2024: Grades!



Our Head TA!

# 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

Attribute its intelligence

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

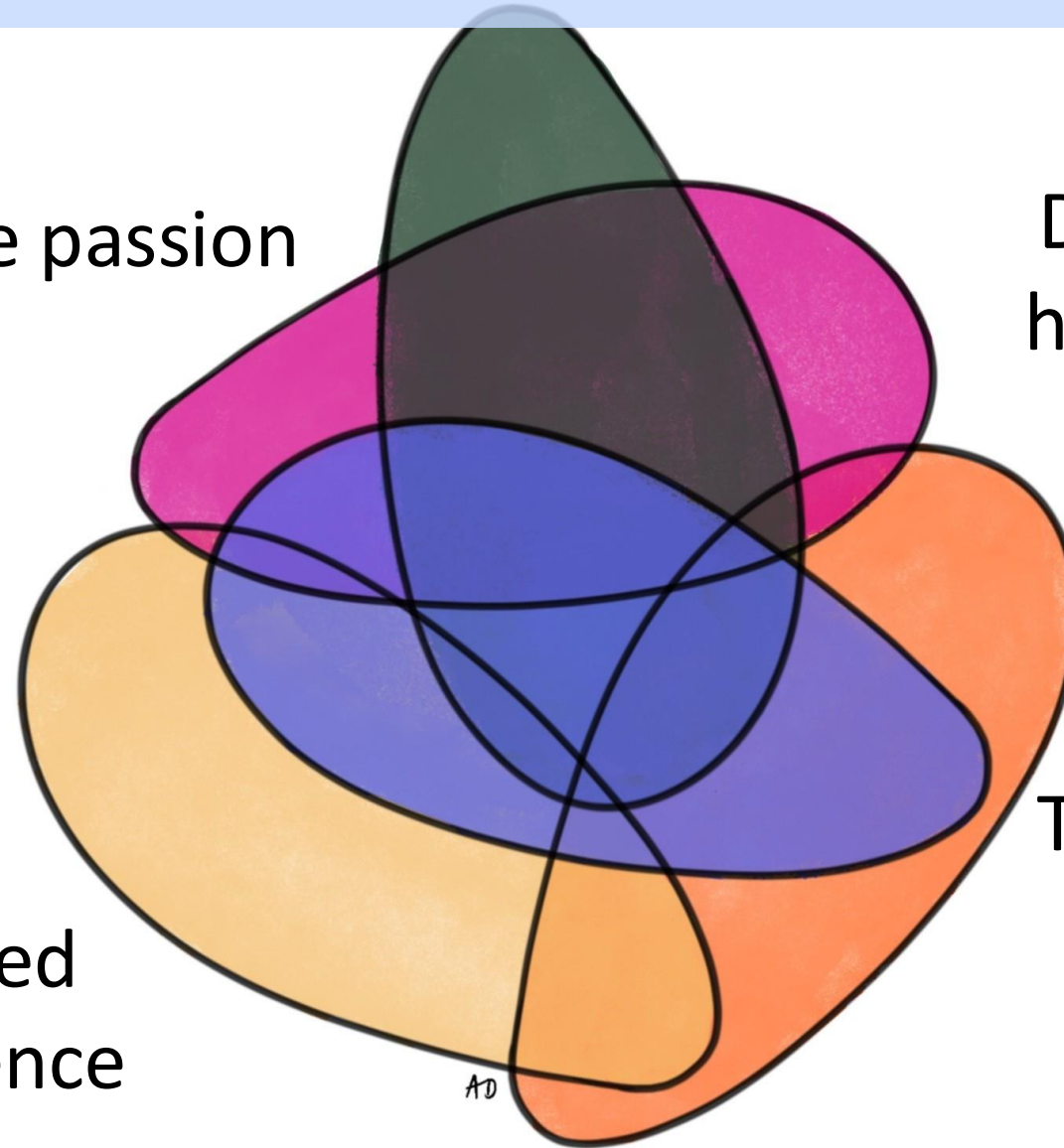
- Home
- Lectures
- Homework
- Grades
- Piazza



# Think about intersectionality

Your side passion

Data that you  
have access to



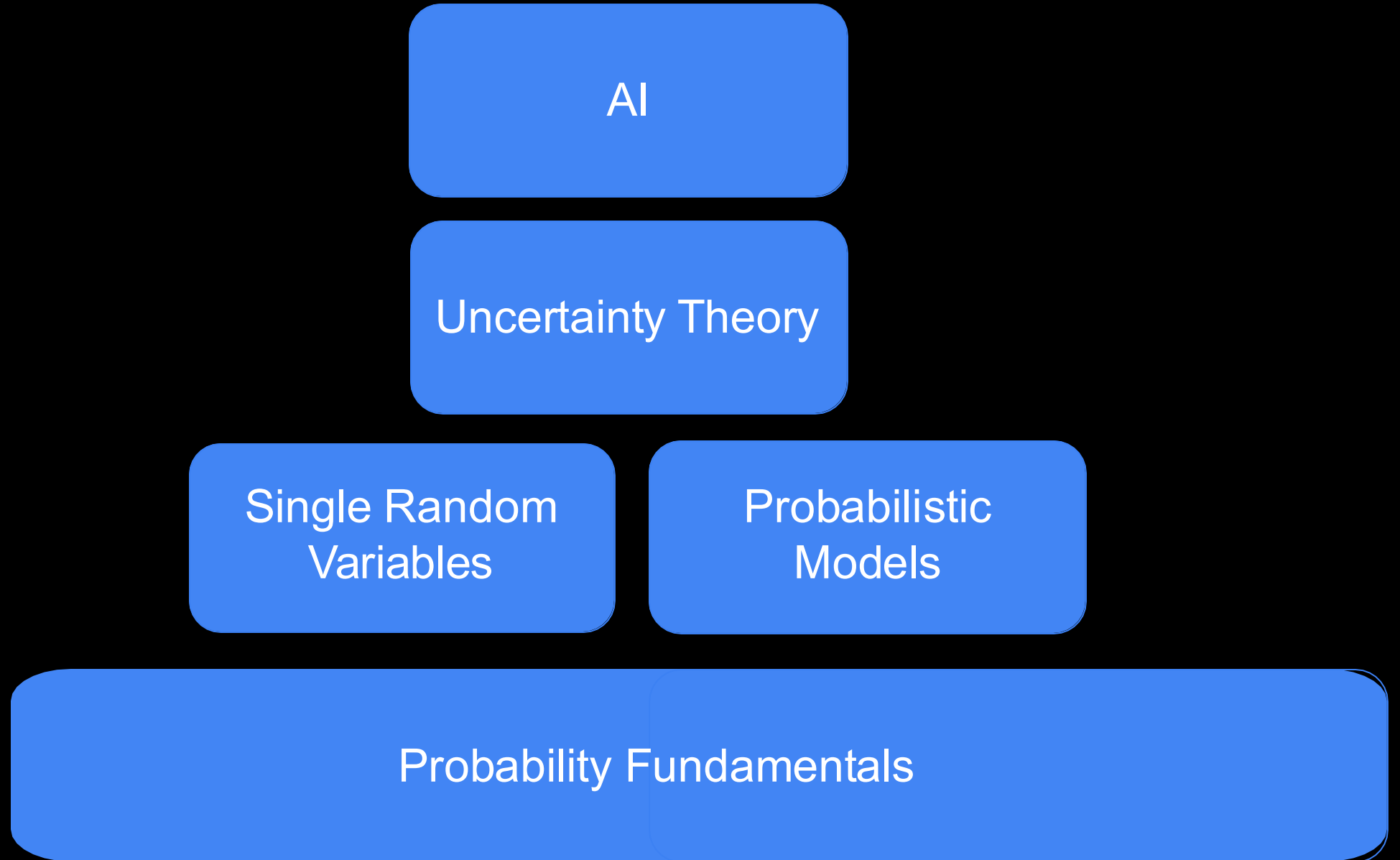
Your lived  
experience

Thompson  
sampling

AD



Last Class...

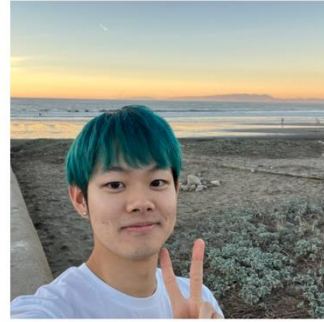




## COURSE VALUES

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

# Fantastic Teaching Team



# 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 "PS1 Sum 20". The problem description is as follows:

1. 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.

2. 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)
```

The right side of the browser shows the "Answer Editor" with the following Python 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)
```

The "Run" button is visible, and the output shows:

```
Running 100000 trials
0.38563
```

At the bottom of the page, there are "Previous Question" and "Next Question" buttons. The user's profile icon shows 95 points.

# Netflix and Learn

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

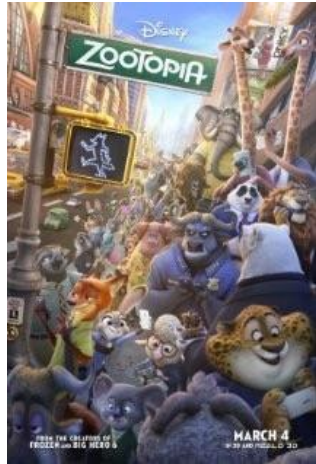
Definition of  
Cond. Probability

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



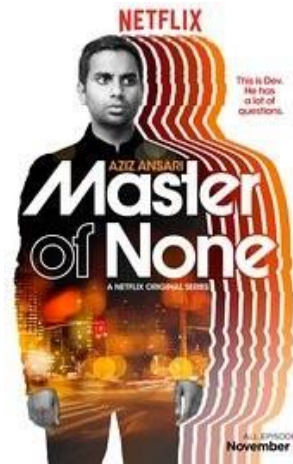
$$P(E) = 0.19$$

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



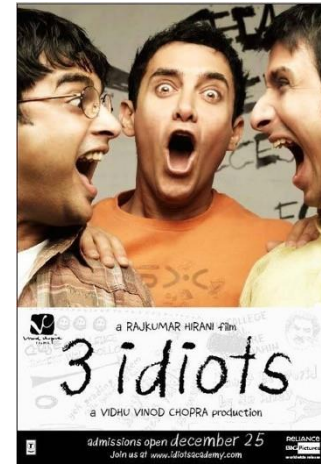
$$P(E) = 0.32$$

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



$$P(E) = 0.20$$

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



$$P(E) = 0.09$$

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



$$P(E) = 0.20$$

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

# Montey Hall Problem



Marilyn discovers the  
Probability Bug



**WHEN YOU MEET YOUR BEST FRIEND**

Somewhere you didn't expect to.



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

# Zika Test



Positive Zika.

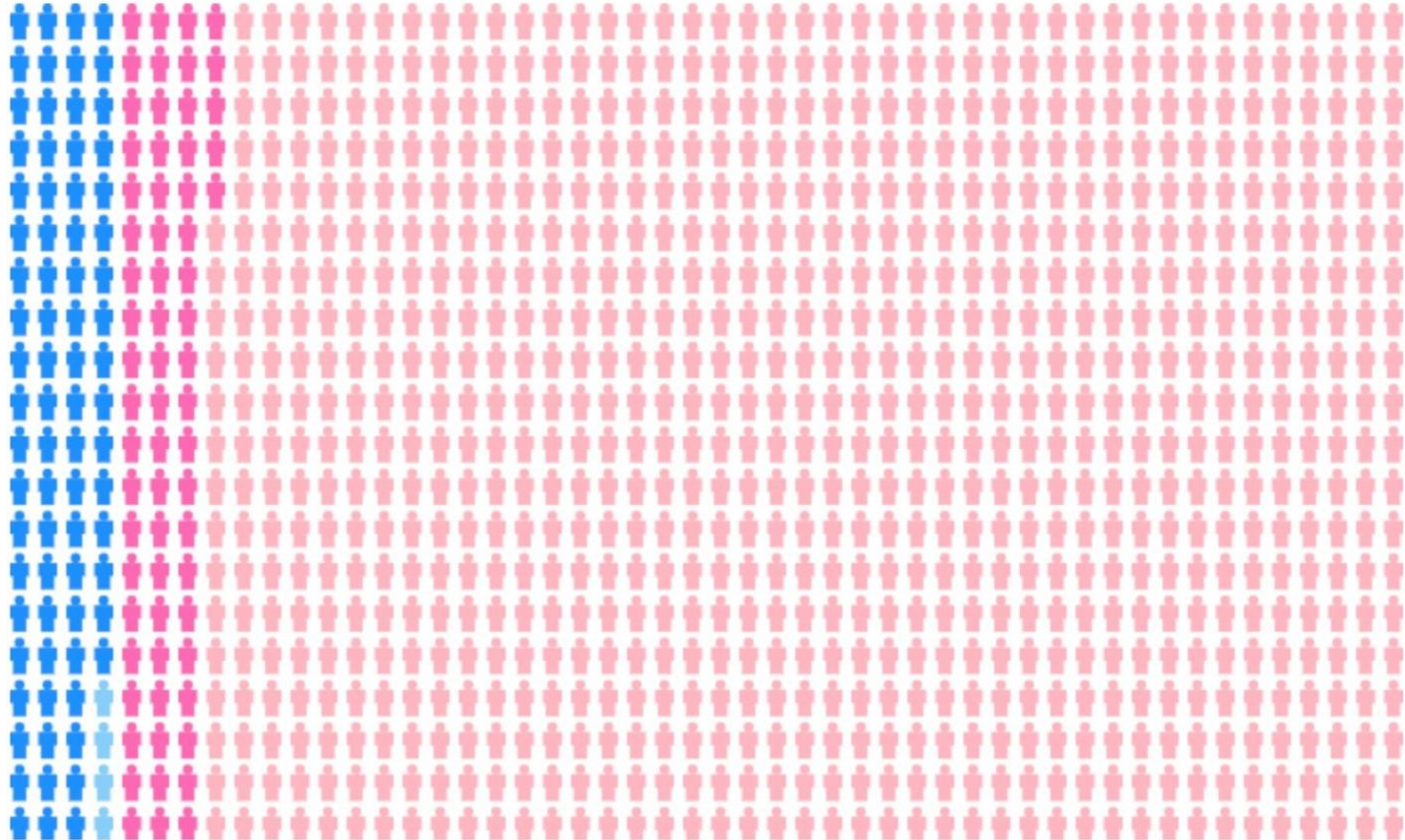
*What is the probability of zika?*

- 
- *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

# Counting Cards

PS1

## Counting Cards

Counting cards refers to when a player keeps track of what cards have already been played during a card-game, in order to have a better estimate of how likely they are to win. Counting cards was successfully used by probability students from MIT to beat casinos worldwide: [MIT Blackjack Team](#) a heist which was popularized by the movie [21](#). The key to counting cards in blackjack is to keep track of the probability of high cards.



In this problem we are going to consider a simpler game called High Card played on a standard 52 card deck. The game works as follows: You decide if you want to play. If you do, the casino deals you a single card. If the card is a high card, (10, Jack, Queen, King or Ace), you win \$20. If it is not, you lose \$20. Another player is playing as well and each game they will play (thus revealing a card). You can play even if you have negative dollars (we assume you will borrow money to pay it back).

If you were given a truly random card out of the deck of 52, your chance of winning would be  $20/52 \approx 0.38$  since 20 of the 52 cards are high. Not very good! But you notice that the casino is only using a single deck of cards. Once a card is played it will not be seen until the dealer deals out all 52 cards. For example the dealer starts a new deck and gives out 2 low cards: 2 of diamonds and 3 of spades. Your chance of winning has just gone up because the proportion of low cards remaining has gone down. Could you beat the casino if you counted cards?

Previous Question      Next Question

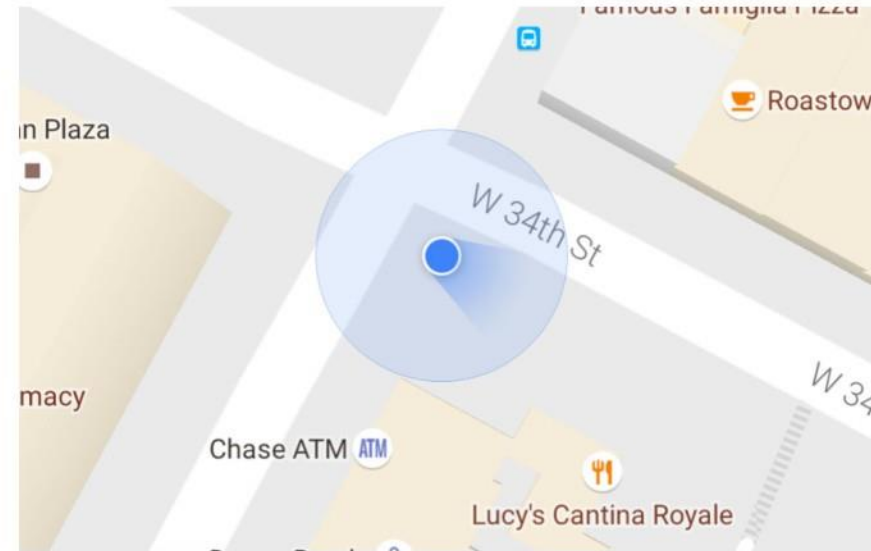
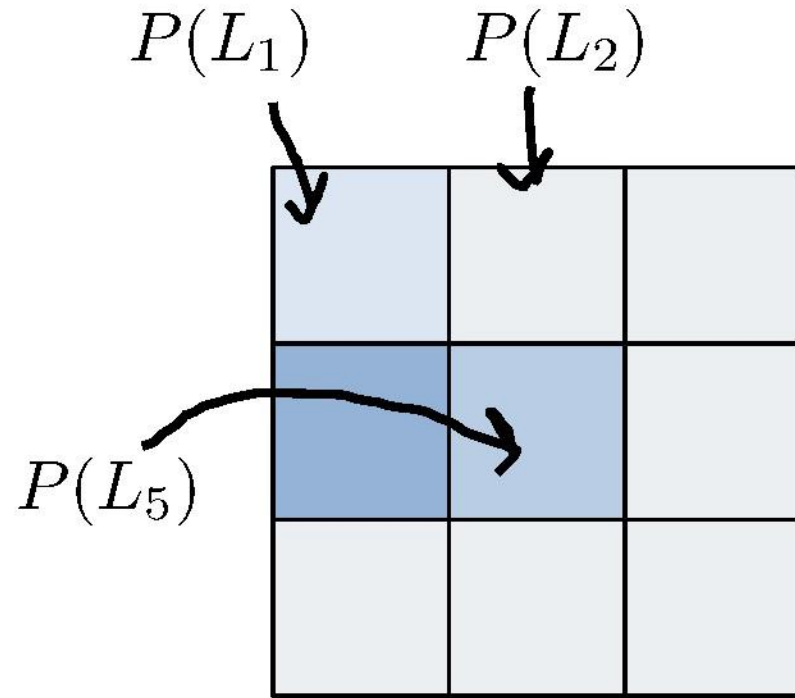
Answer Editor      Solution

Agent:

```
1 """
2 counting_agent.py
3 This file defines an agent "counting_agent" which plays the game
4 High Card. The function gets called each time it is the agent's
5 turn. The cards_played list has all cards which have been played so
6 far.
7
8 def counting_agent(cards_played):
9     # default strategy: always play
10    return 'play'
```

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

# Time to Start Flippin Coins

## Exactly $k$ heads

Next lets try to figure out the probability of exactly  $k$  heads in the  $n$  flips. Importantly we don't care where in the  $n$  flips that we get the heads, as long as there are  $k$  of them. Note that this question is different than the question of first  $k$  heads and then  $n - k$  tails which requires that the  $k$  heads come first! That particular result does generate exactly  $k$  coin flips, but there are others.

There are many others! Let's ask the computer to list the ways we could generate exactly  $k$  heads within  $n$  coin flips.

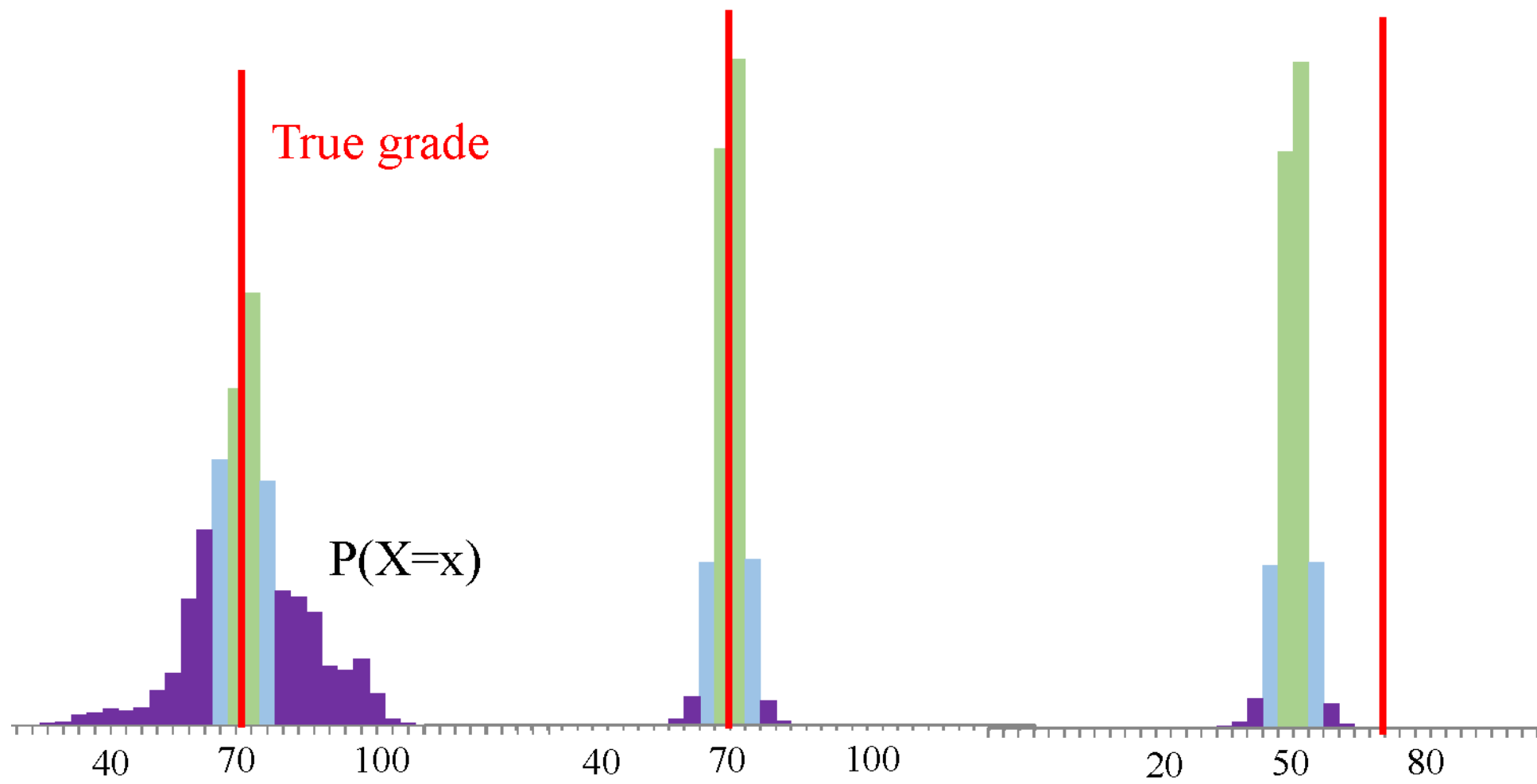
The output region is scrollable:

(H, H, H, H, T, T, T, T, T, T)  
(H, H, H, T, H, T, T, T, T, T)  
(H, H, H, T, T, H, T, T, T, T)  
(H, H, H, T, T, T, H, T, T, T)  
(H, H, H, T, T, T, T, H, T, T)  
(H, H, H, T, T, T, T, T, H, T)  
(H, H, H, T, T, T, T, T, T, H)  
(H, H, T, H, H, T, T, T, T, T)  
(H, H, T, H, T, H, T, T, T, T)  
(H, H, T, H, T, T, H, T, T, T)  
(H, H, T, H, T, T, T, H, T, T)  
(H, H, T, H, T, T, T, T, H, T)  
(H, H, T, H, T, T, T, T, T, H)  
(H, H, T, T, H, H, T, T, T, T)  
(H, H, T, T, H, T, H, T, T, T)  
(H, H, T, T, H, T, T, H, T, T)  
(H, H, T, T, H, T, T, T, H, T)  
(H, H, T, T, H, T, T, T, T, H)  
(H, H, T, T, T, H, H, T, T, T)  
(H, H, T, T, T, H, T, H, T, T)



# Random Variables

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

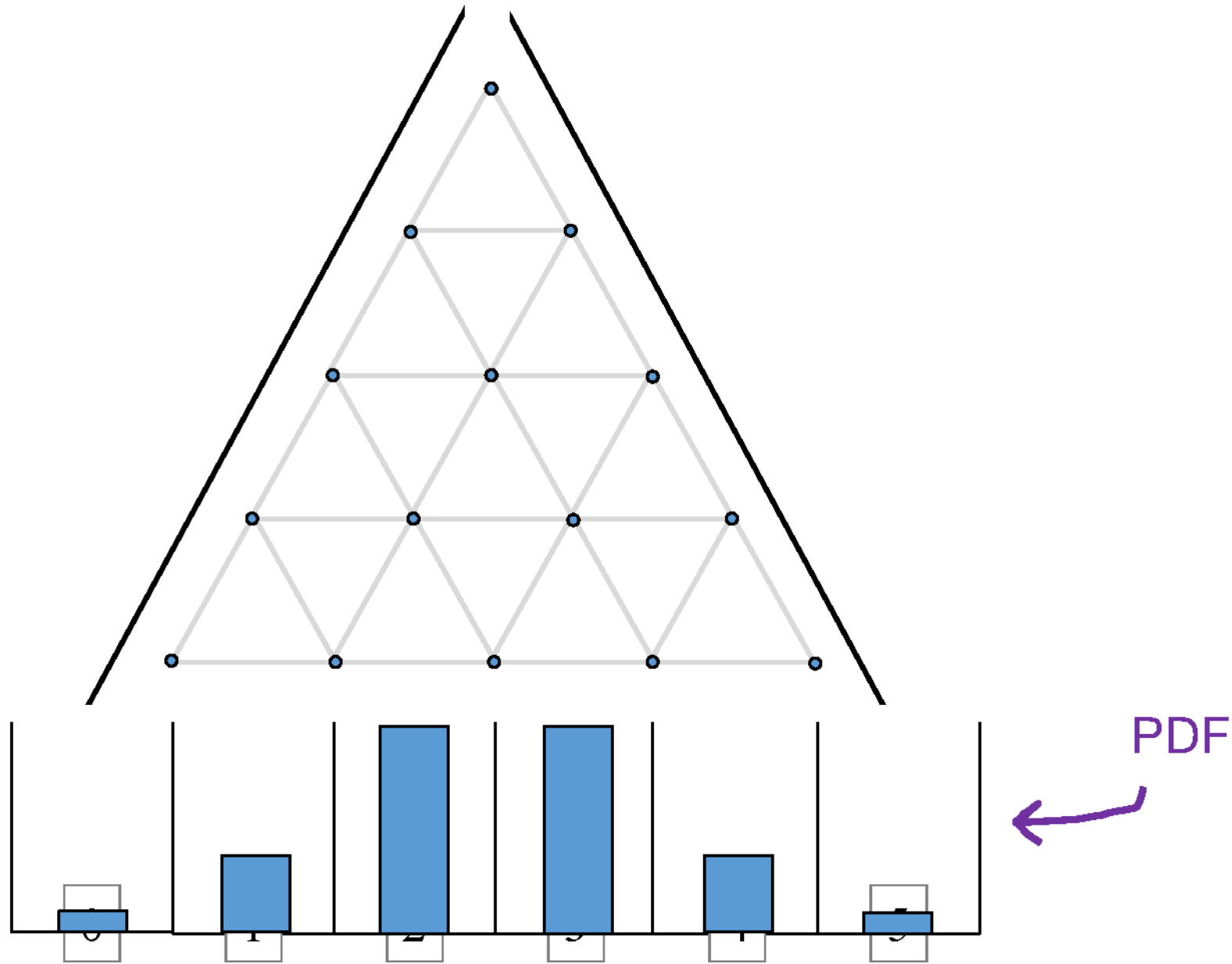


A

B

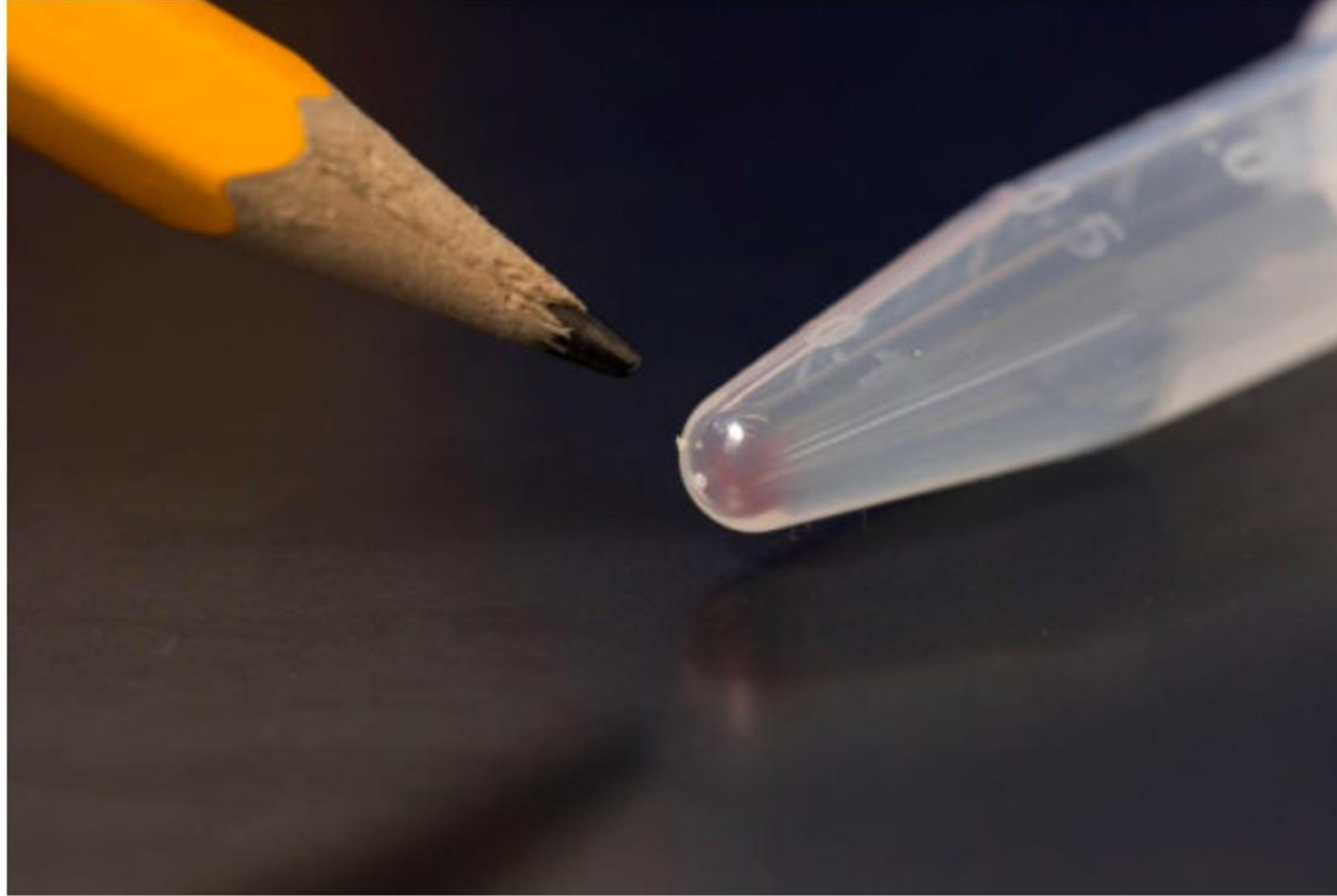
C

# Binomial

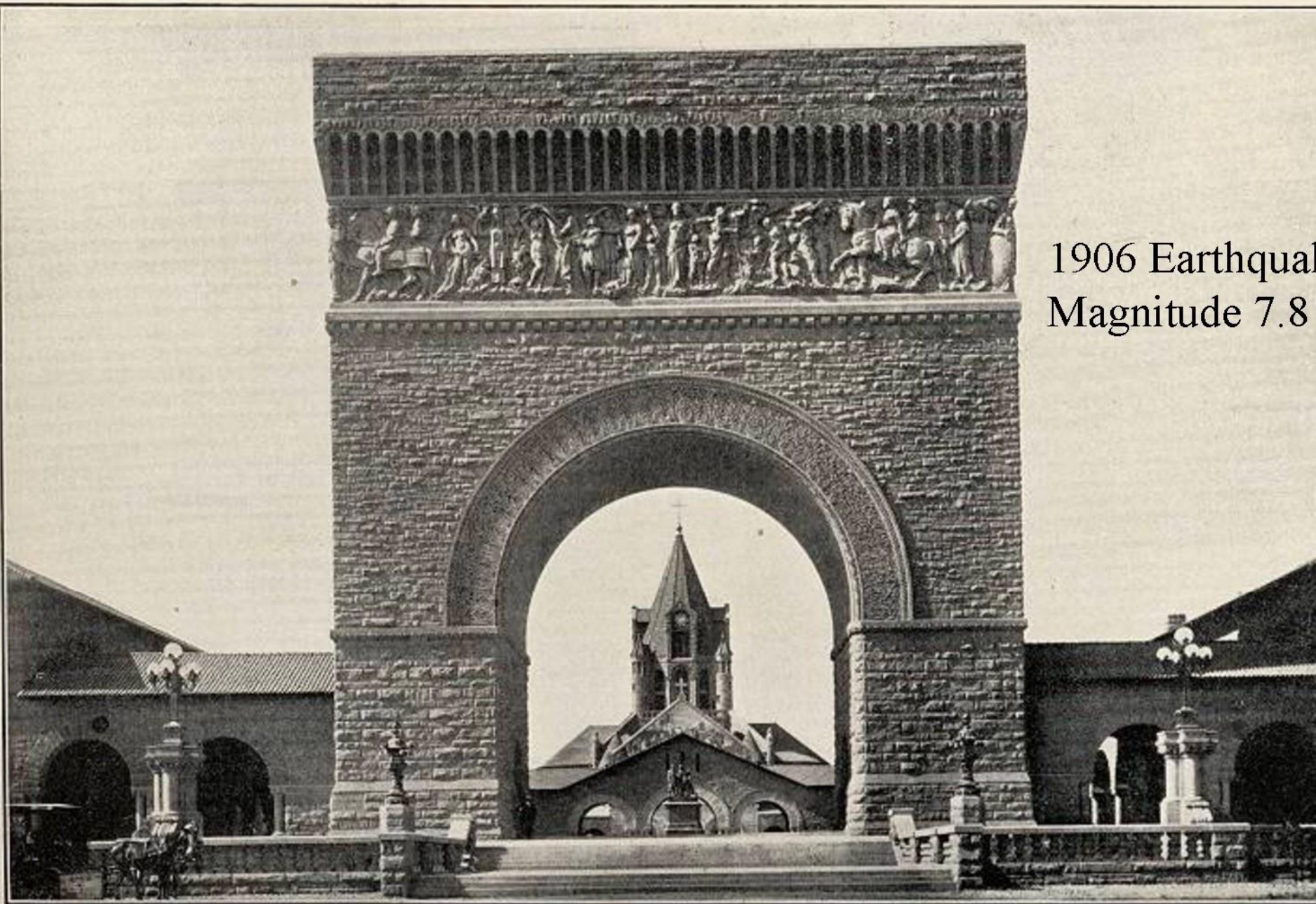




# Storing Data on DNA

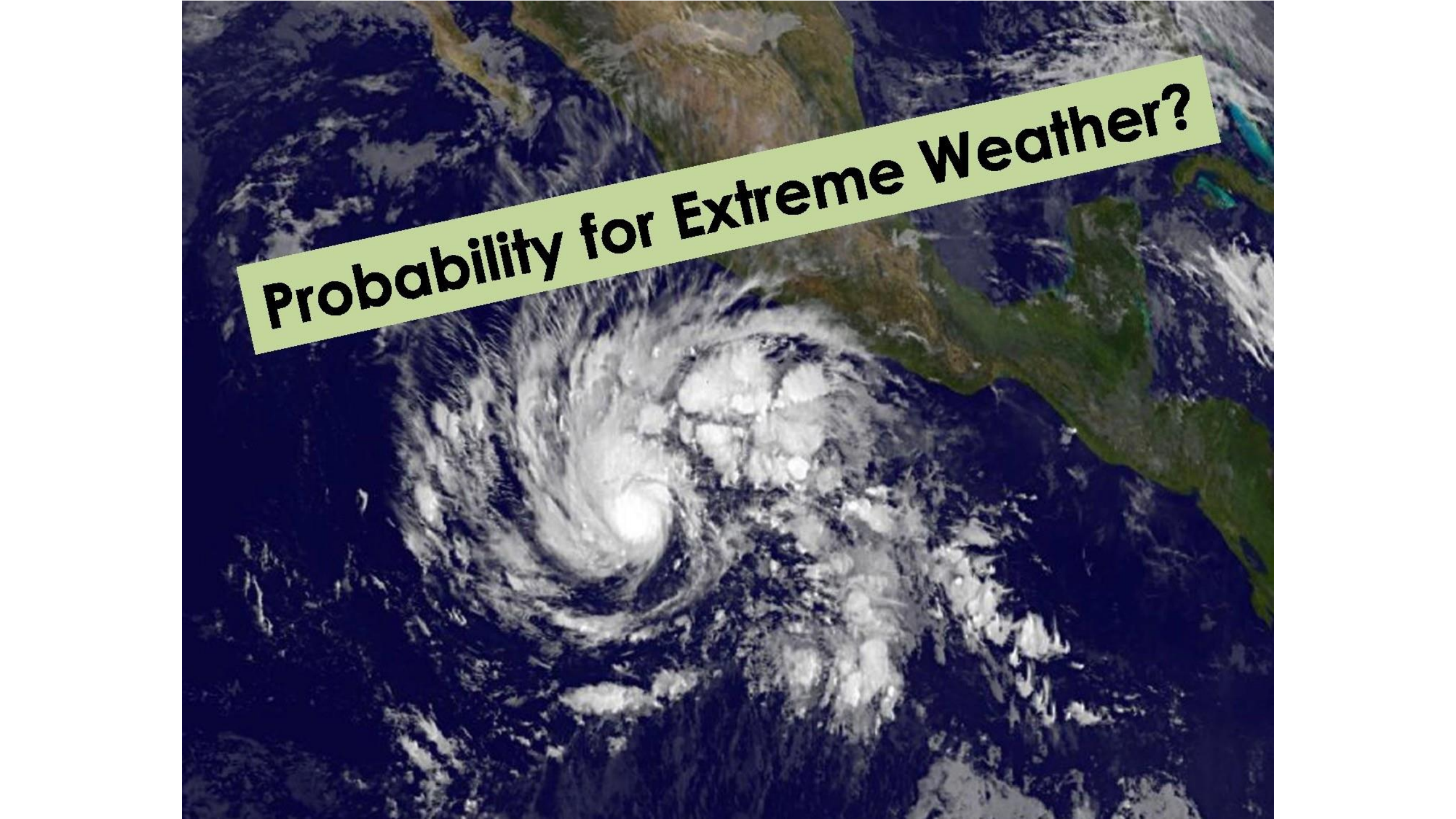


All the movies, images, emails and other digital data from more than 600 smartphones (10,000 gigabytes) can be stored in the faint pink smear of DNA at the end of this test tube.



1906 Earthquake  
Magnitude 7.8

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

A satellite image of a hurricane over the Gulf of Mexico. The hurricane's eye is clearly visible in the center, surrounded by dense, swirling cloud bands. The surrounding ocean is dark blue, and the landmasses of North and Central America are visible in shades of green and brown. A light green banner is overlaid on the top half of the image, containing the text 'Probability for Extreme Weather?'.

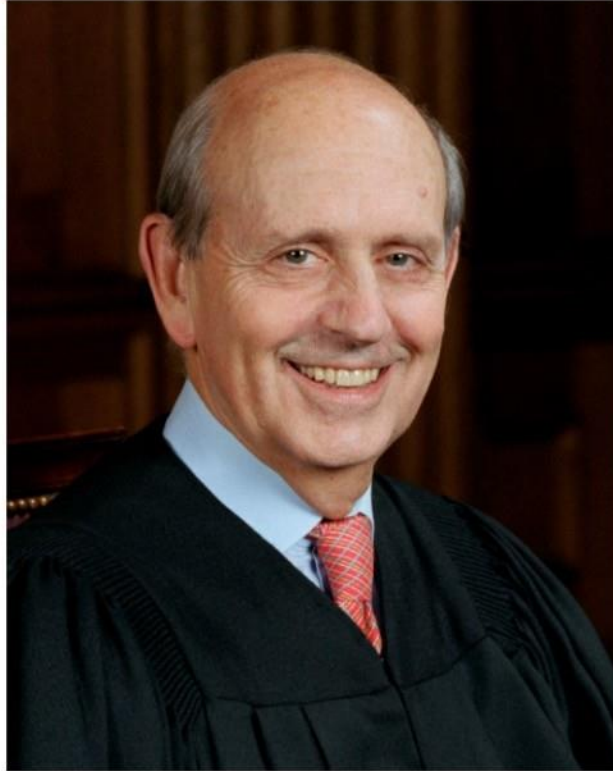
**Probability for Extreme Weather?**

# 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.

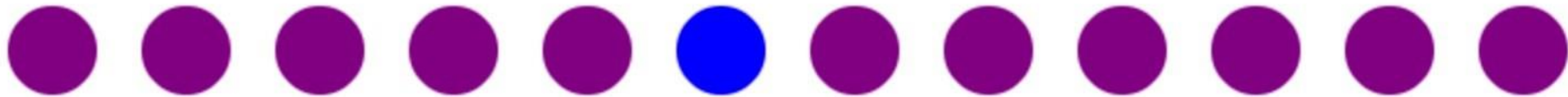


# Representative Juries



Simulate

Simulation:

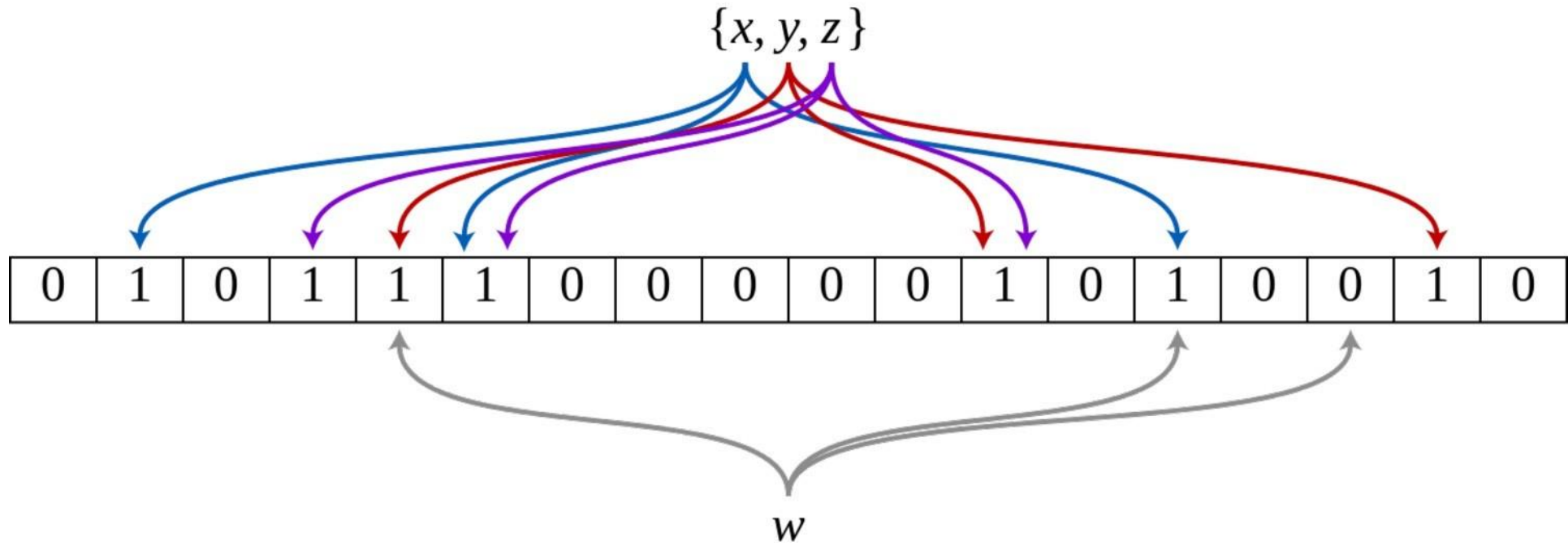


# Dating at Stanford

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



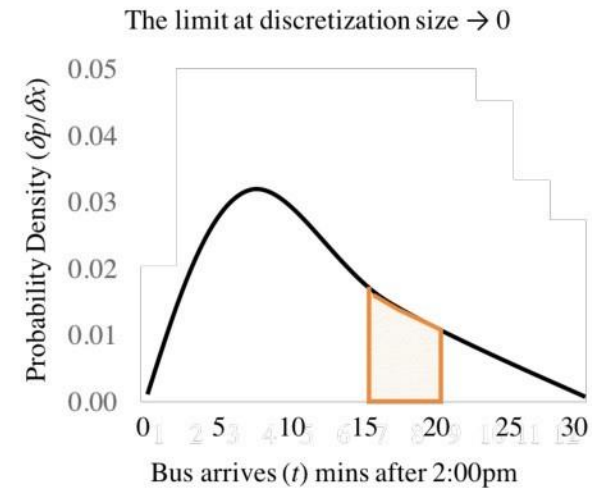
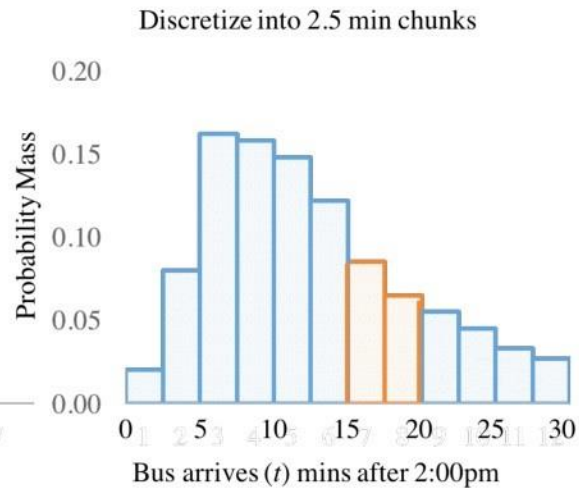
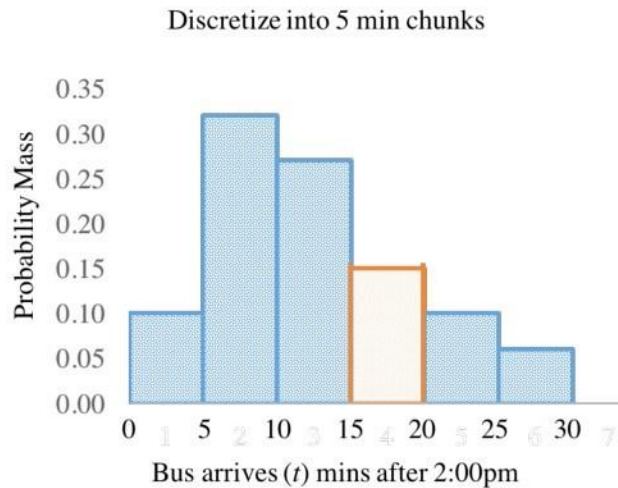
random( ) ?

# Riding the Marguerite

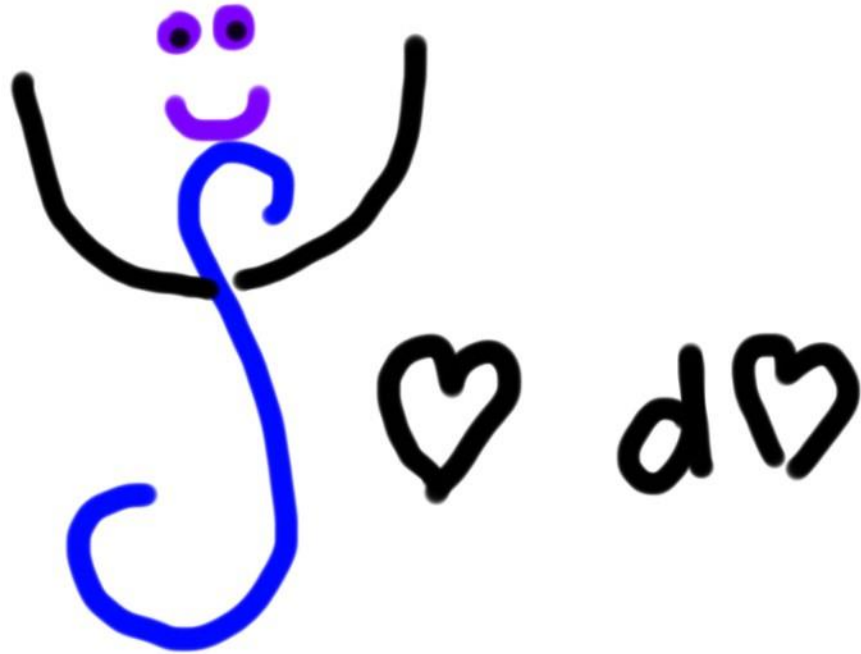


You are running to the bus stop.  
You don't know exactly when  
the bus arrives. You arrive at  
2:20pm.

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



# Integrals

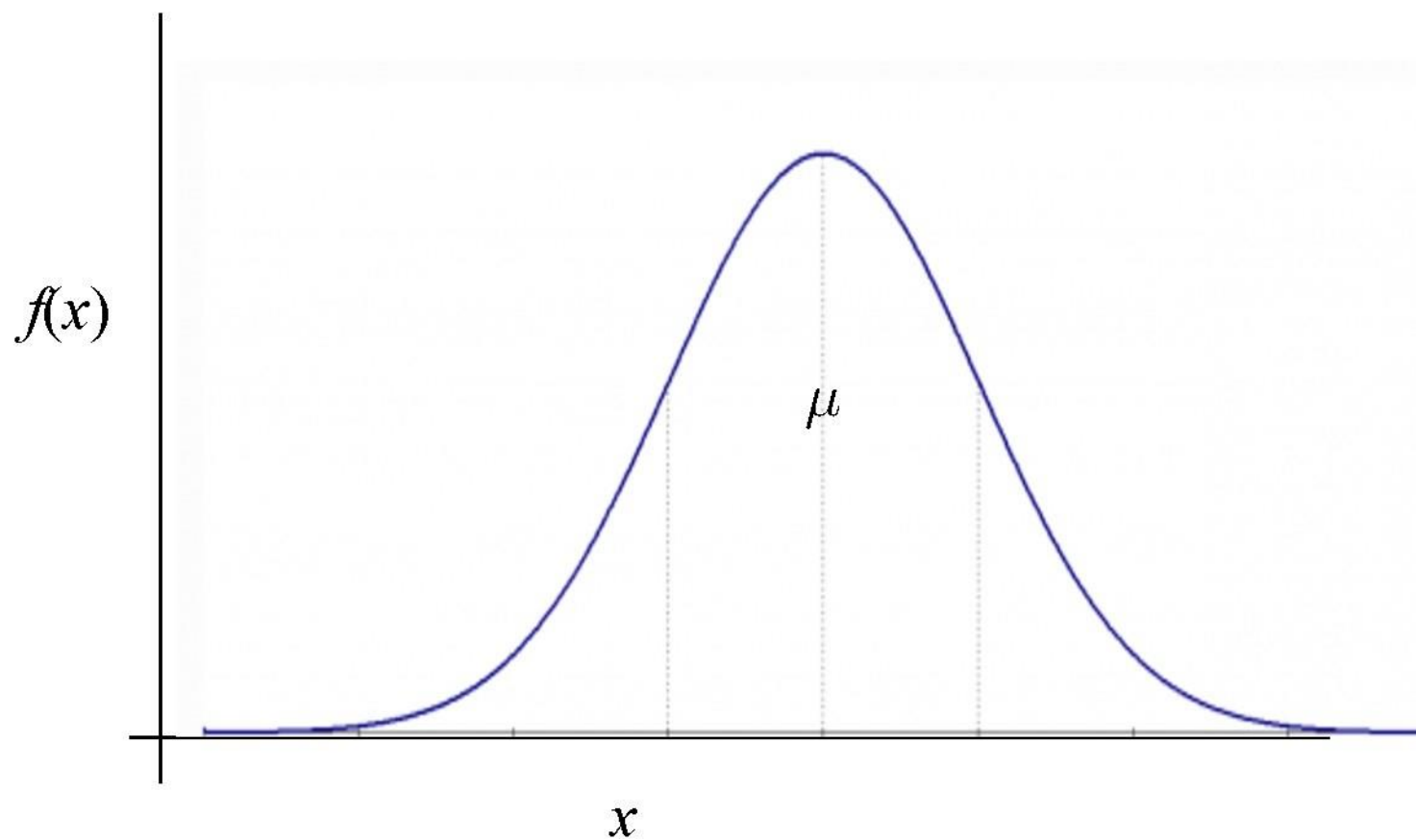


\*loving, not scary

# Probability Density Function

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

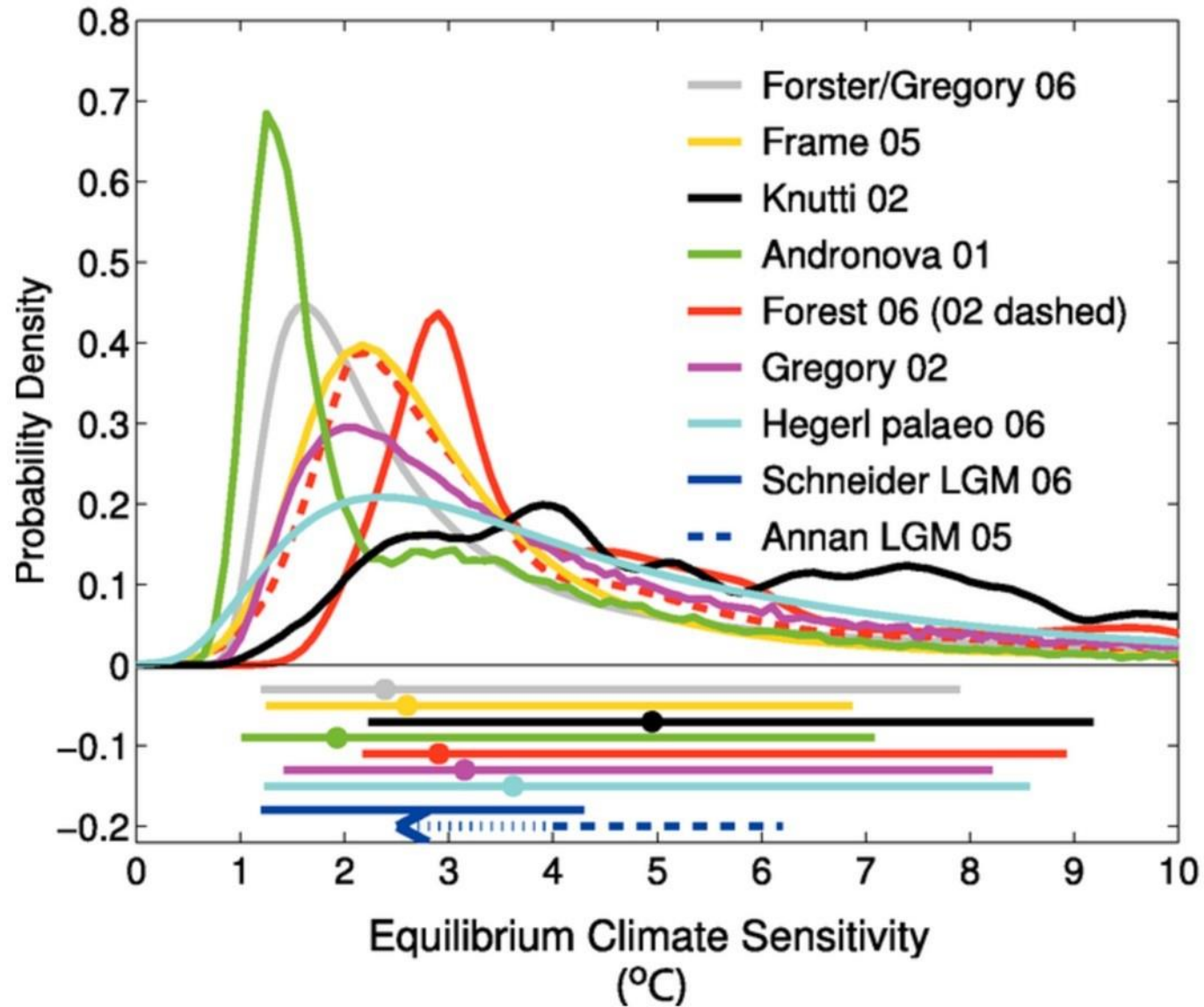
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



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

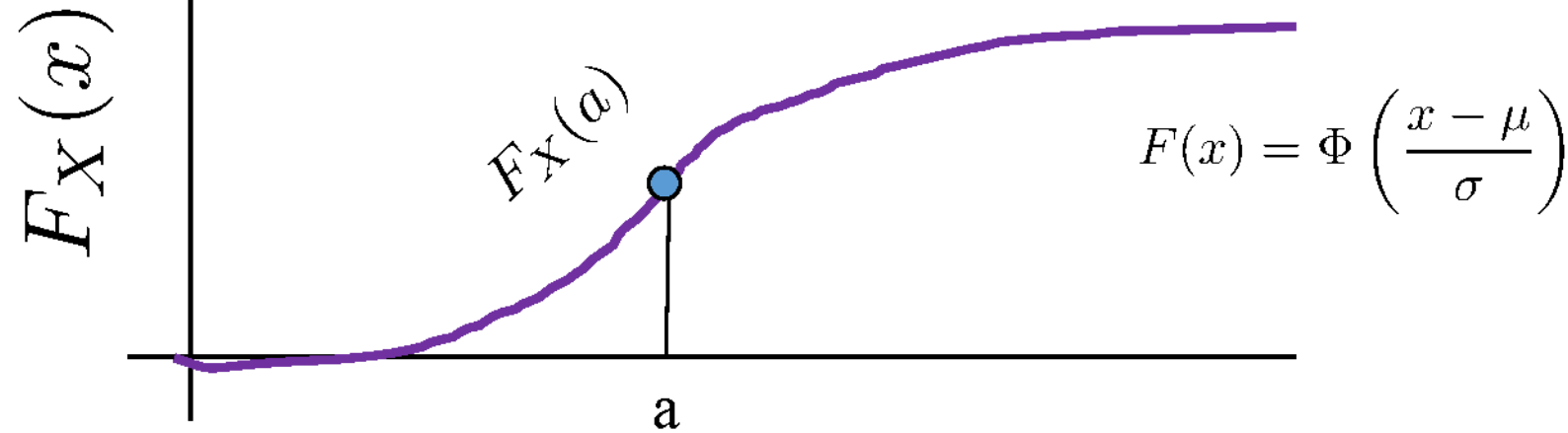
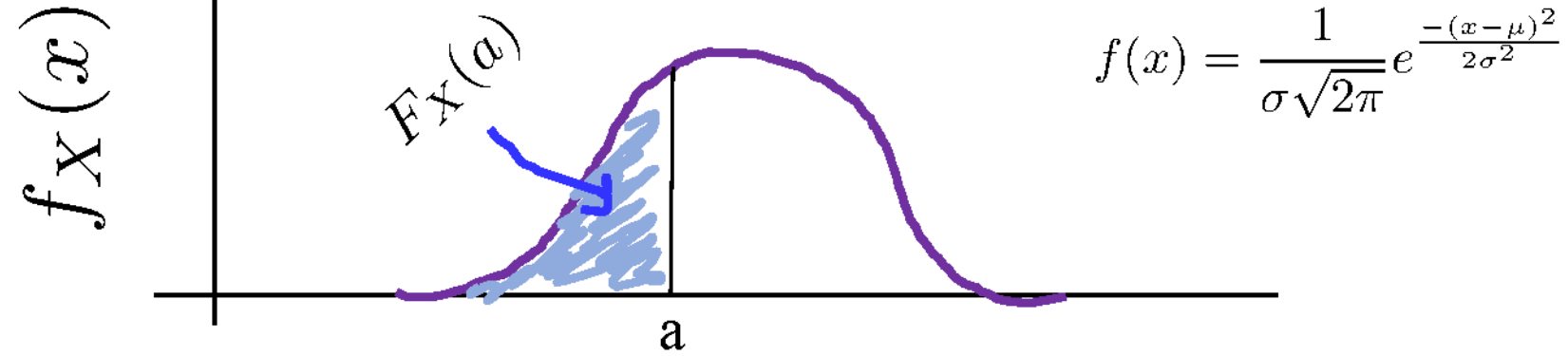
A probability!

# Climate Sensitivity



# PDF and CDF of a Normal

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



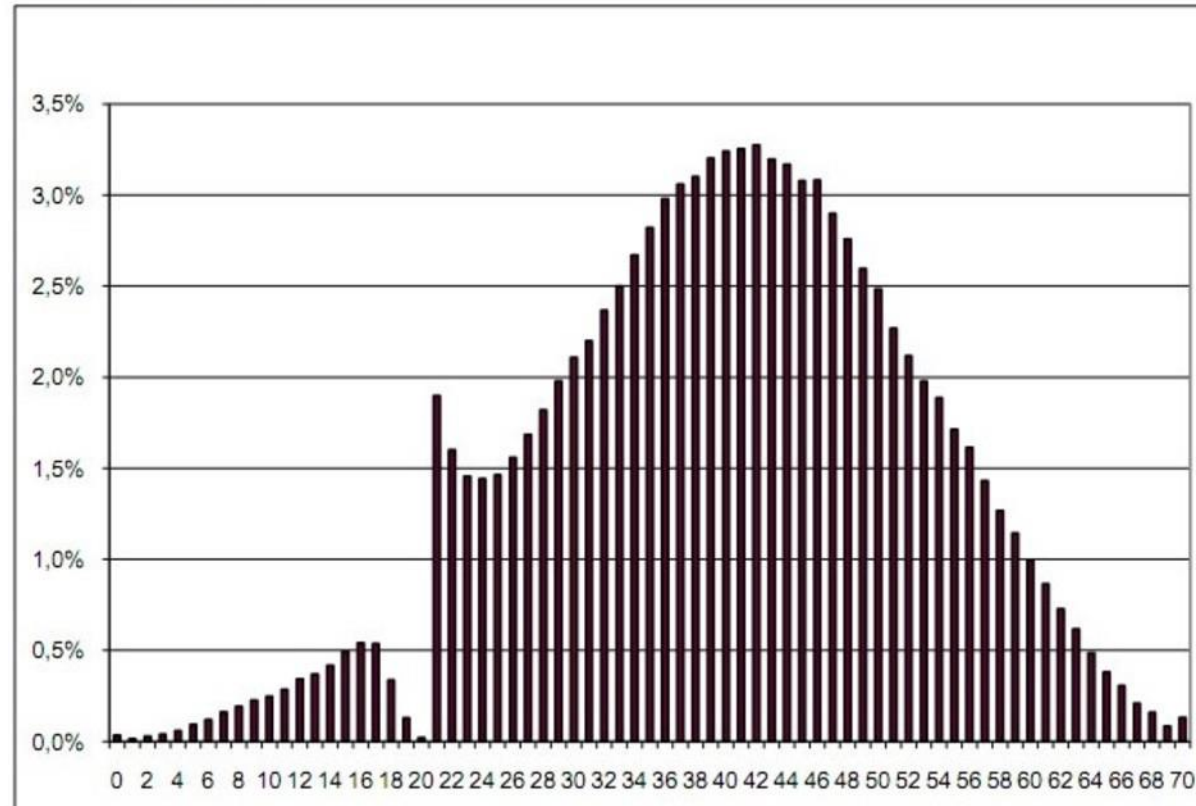
A CDF is the integral from  $-\infty$  to  $x$  of the PDF

# 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

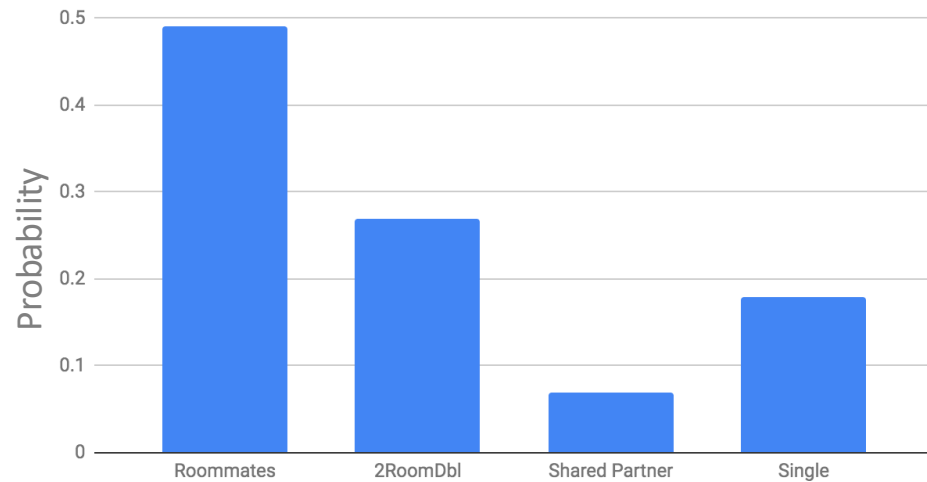
# Probabilistic Models



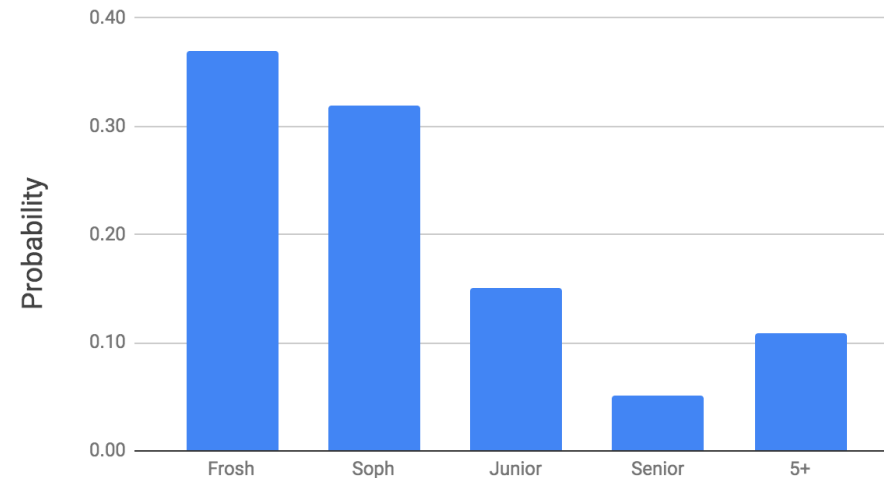
# Joint Probability Table

	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



# Inference

## **Inference** *noun*

Updating one's belief about a random variable (or multiple) based on conditional knowledge regarding another random variable (or multiple) in a probabilistic model.

TLDR: conditional probability with random variables.

# Inference

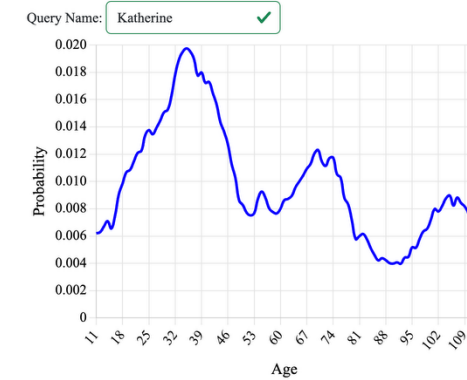
## Age from C14



## Updated Delivery Prob



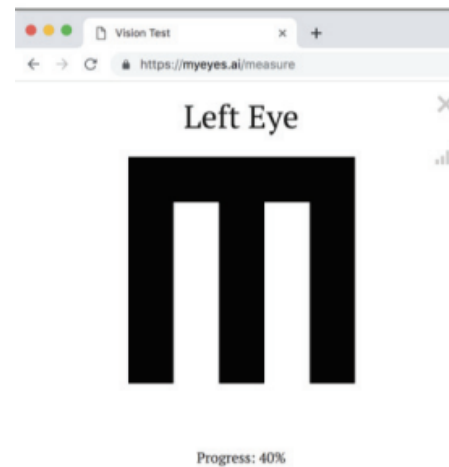
## Age from Name



## Hidden Chambers



## Stanford Eye Test



## Updating Lidar Belief



```

def update_belief_carbon_dating(m = 900):
    # pr_A[i] is P(Age = i | m = 900).
    pr_A = {}
    for i in range(100,10000+1):
        prior = 1 / n_years # P(A = i)
        likelihood = calc_likelihood(m, i) #P(M=m | A=i)
        pr_A[i] = likelihood * prior
    # implicitly computes the normalization constant
    normalize(pr_A)
    return pr_A

```

```

def update_belief_name_to_age(name = 'Laura'):
    # pr_age[i] is P(Age = i | name).
    # prob_name_and_age is just a counting from the US
    # Social Security database.
    pr_age = {}
    for i in range(10,110):
        pr_age[i] = calc_prob_name_and_age(name, i)
    # implicitly computes the normalization constant
    normalize(pr_age)
    return pr_age

```

```

def update_belief_baby(prior, today = 10):
    # pr_D[i] is P(D = i | No Baby Yet).
    pr_D = {}
    for i in range(-50,25):
        # P(NoBaby | D = i)
        likelihood = 0 if i < today else 1
        pr_D[i] = likelihood * prior[i]
    # implicitly computes the LOTP
    normalize(pr_D)
    return pr_D

```

What do you notice  
is the same. What is  
different?

# General “Inference”



# General "Inference"

WebMD Symptom Checker BETA

INFO

SYMPTOMS

QUESTIONS

CONDITIONS

DETAILS

TREATMENT

## Add more symptoms

Type your main symptom here

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**



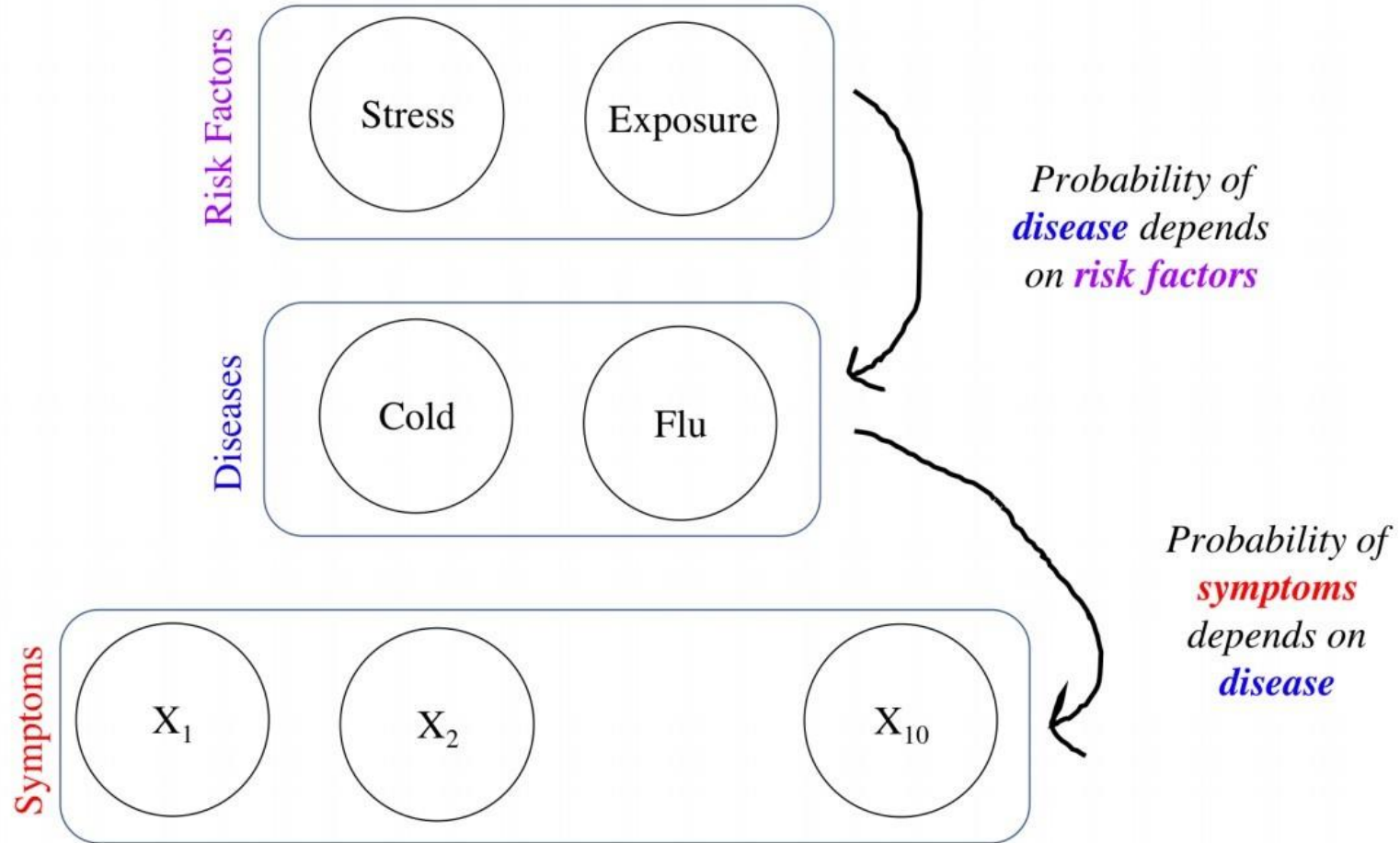
Previous

Info

Continue



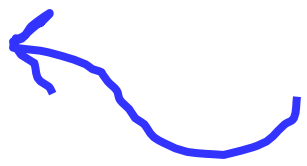
# Huge Joint Models



```
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)
```

```
webMd -- bash -- 38x22
[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]
Observation = [None, None, None, 1]
Pr(Flu | Obs) = 0.140635888502
>
```

Each one of these is one posterior sample:



[Flu, Ugrad, Fever, Tired]

# Multinomial

Example document:

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

$n = 18$

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

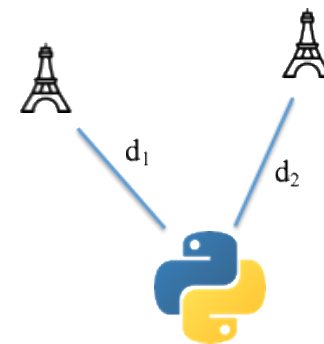
It's a Multinomial!

Probability of seeing  
this document | spam

The probability of a word in  
spam email being viagra



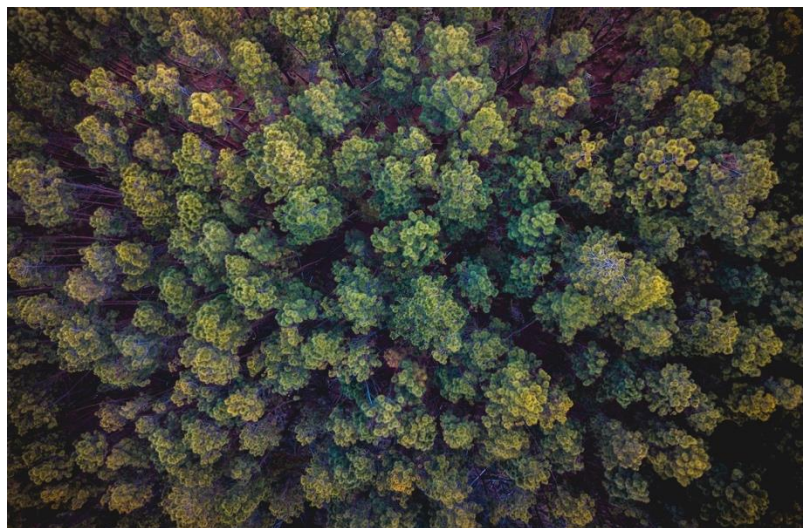
# Hard Midterm, Great Job



11 Sparkle's Outback Tooth Hunt Saltar →



As they traveled, Sparkle grew worried.  
"What if I can't find the tooth? The Tooth Fairy Council will be so disappointed!"  
Sheila comforted **don't** up, little fairy. We'll find it together. ?



```
def simulation_1(p=0.4, k=0.01):  
    X = bern(p)           # sar  
    Y = bern(p+k)        # in  
    return Y - X
```

```
def simulation_2(p=0.4, k=0.01):  
    X = bern(p)  
    if X == 1:  
        Y = 1  
    else:  
        Y = bern(k/(1-p))  
    return Y - X
```



# PEP

New last 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

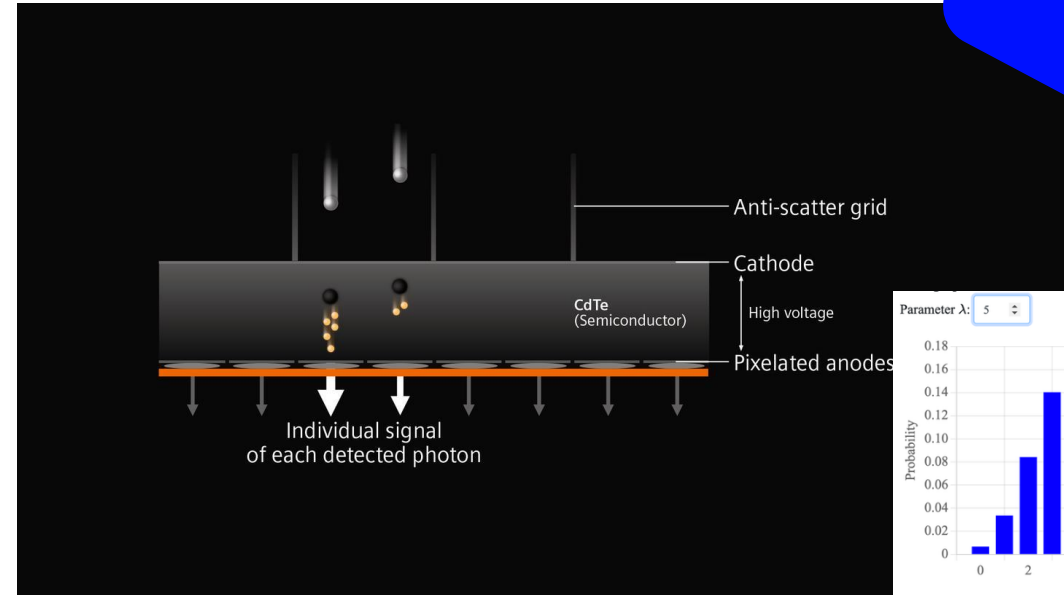
Tuesday, Dec 3

Select a time slot

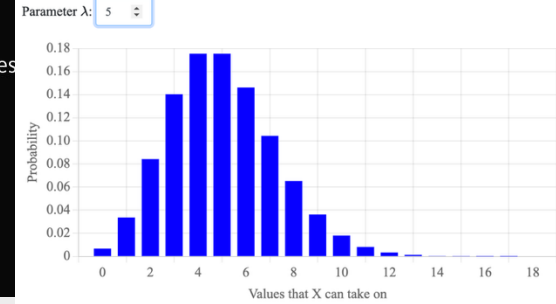
Wednesday, Dec 4

✓ Select a time slot

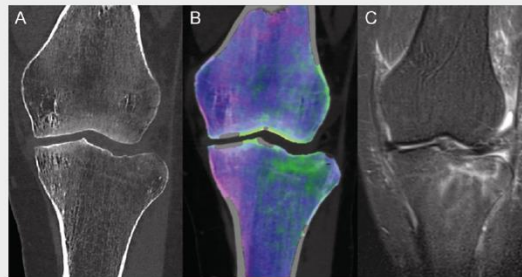
- Wednesday, Dec 4 9:15 AM
- Wednesday, Dec 4 9:30 AM
- Wednesday, Dec 4 12:00 PM
- Wednesday, Dec 4 12:15 PM



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

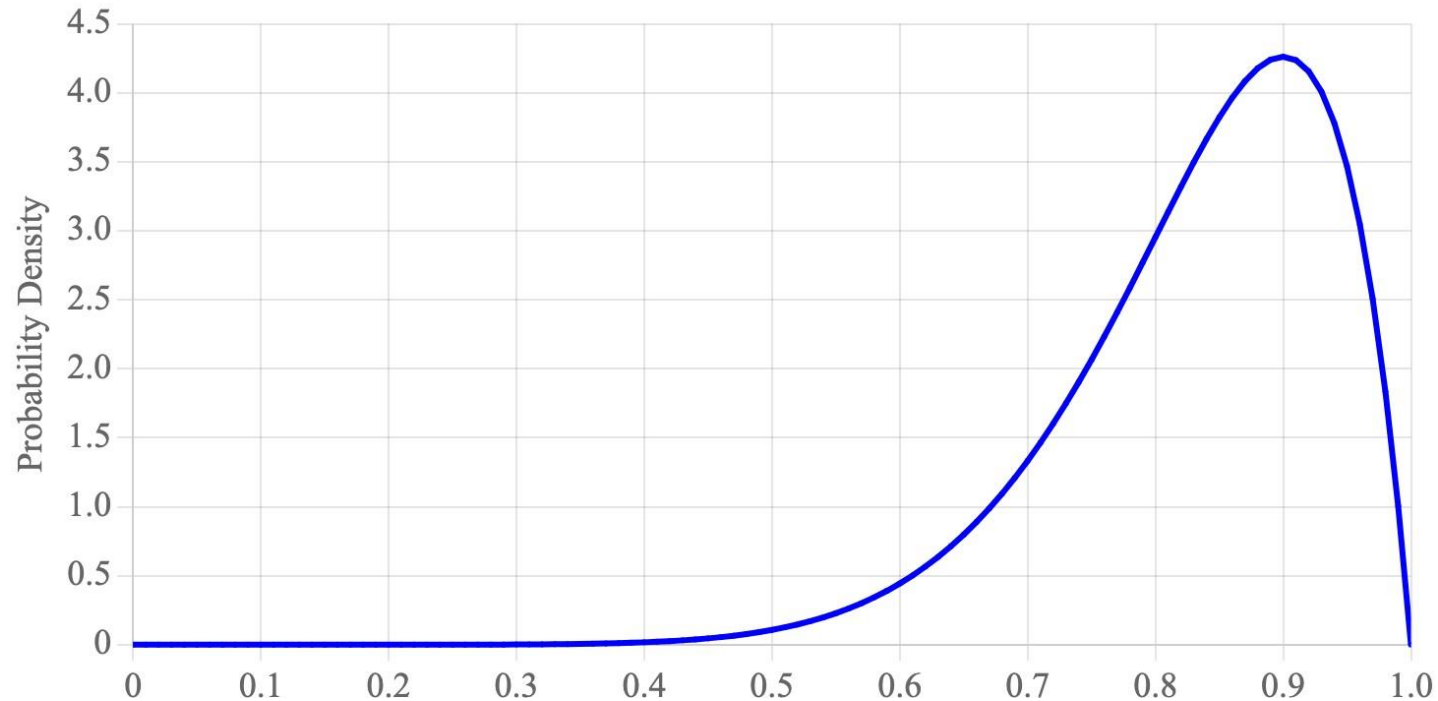
Learning Goal: Be fluent in the language of probability

# 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:



# Let's Play

Drug A

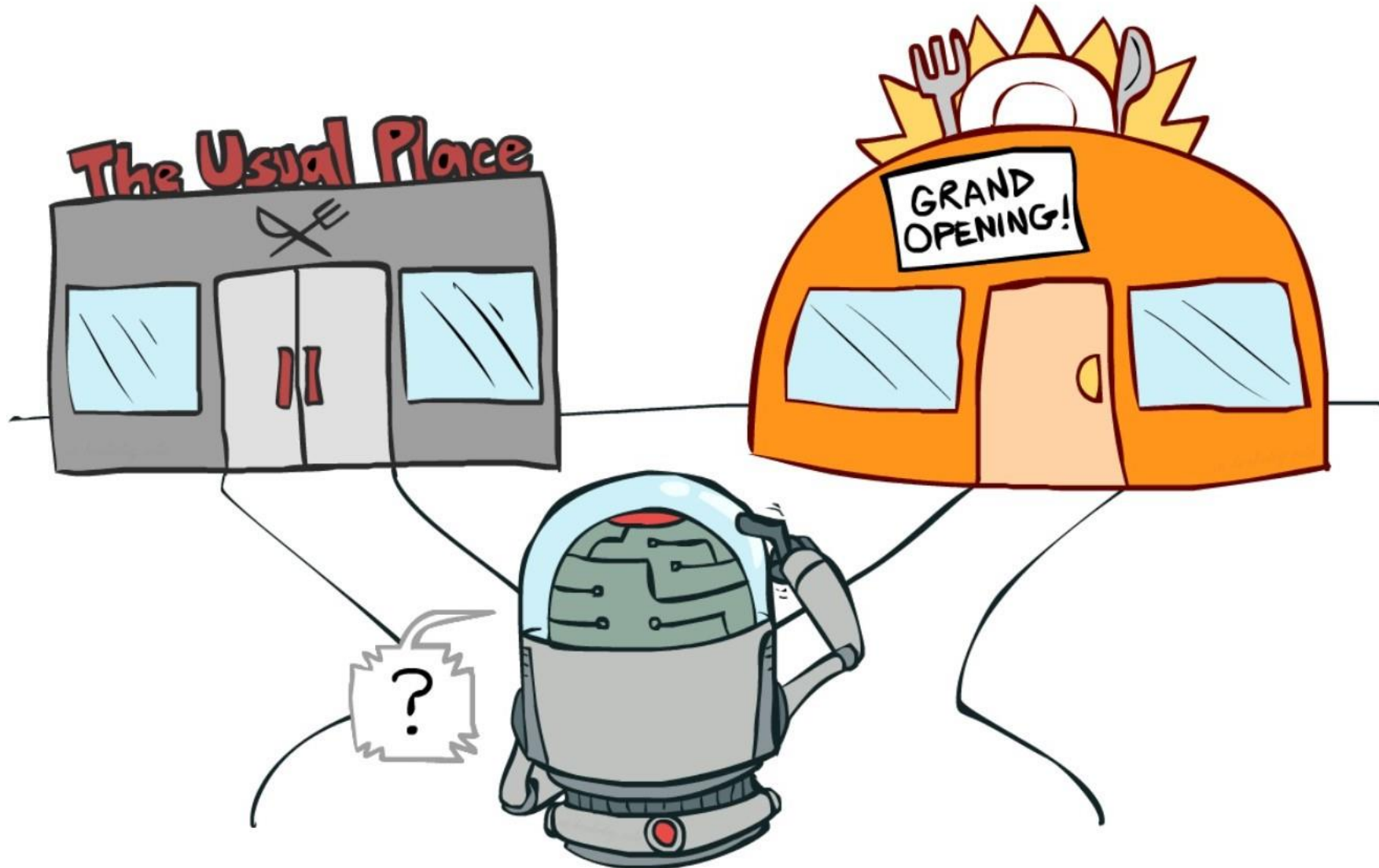


Drug B

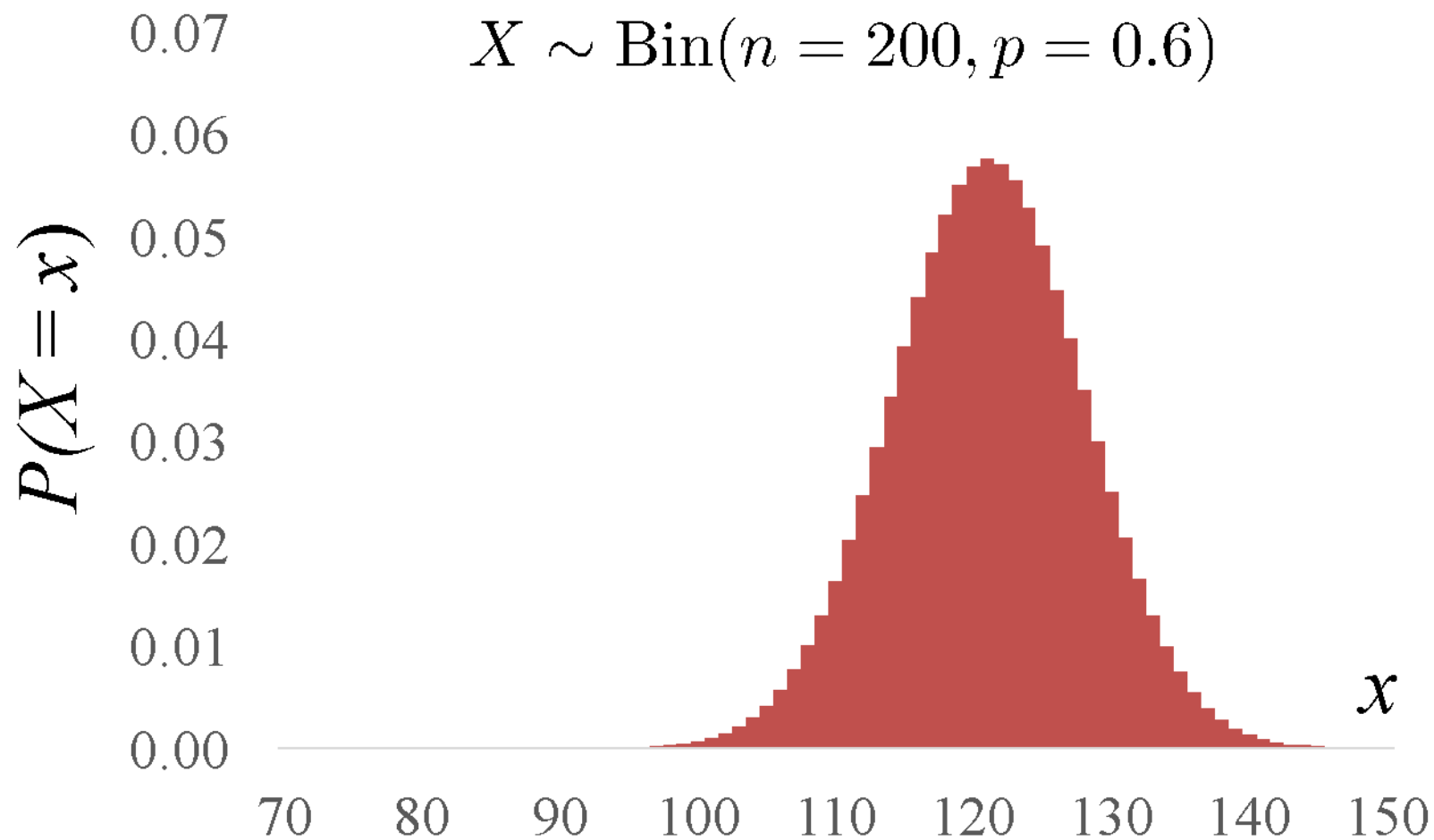


Which one do you give to a patient?

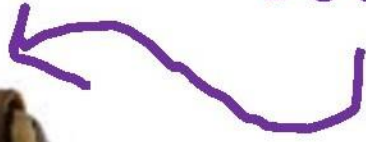
# Thompson Sampling



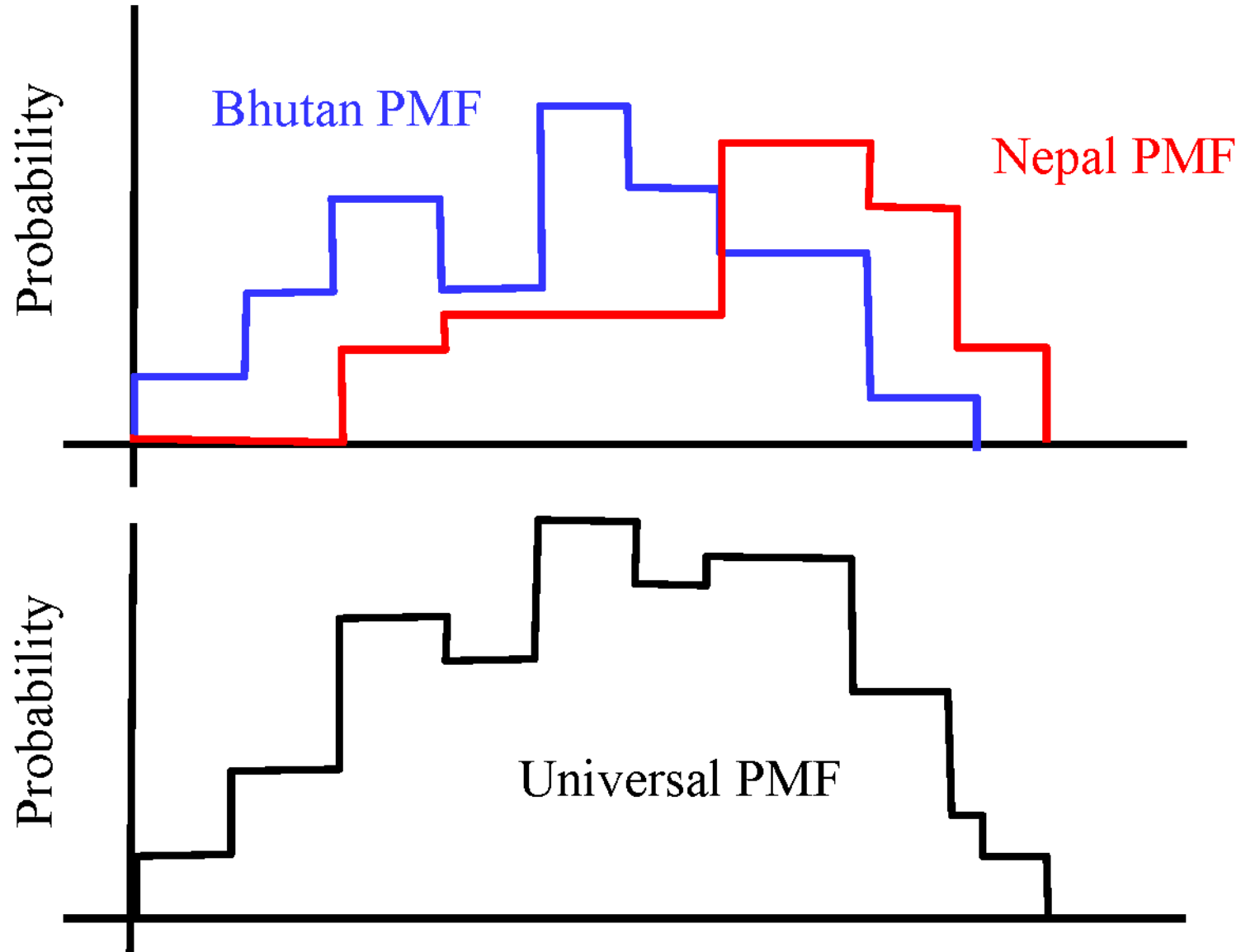
# C.L.T. Explains This



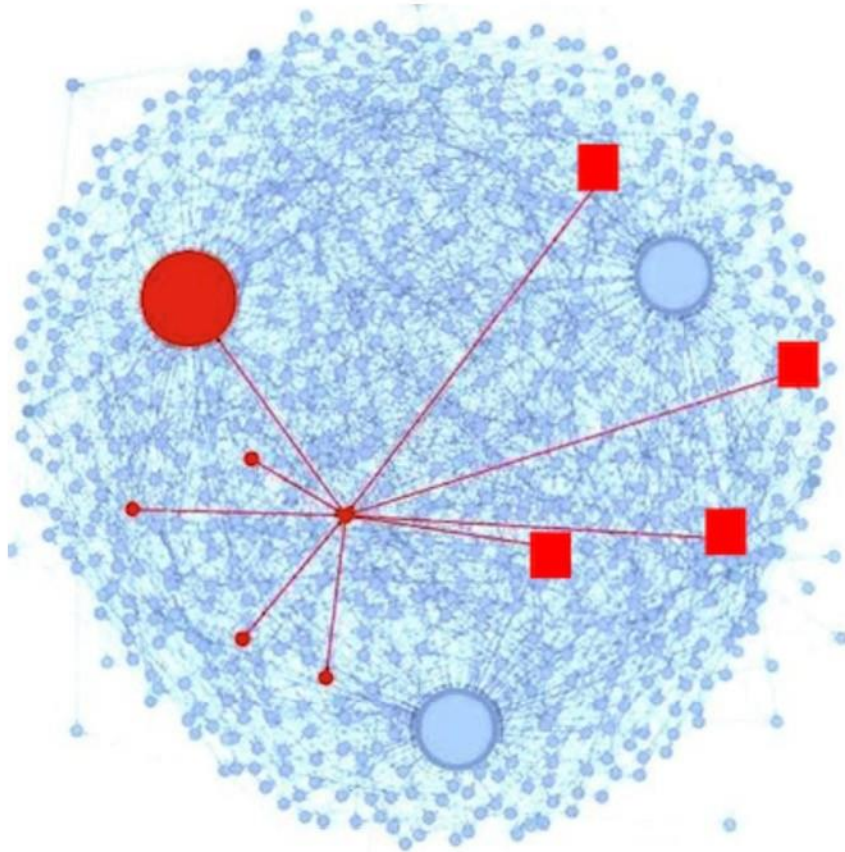
Bootstrap



# 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

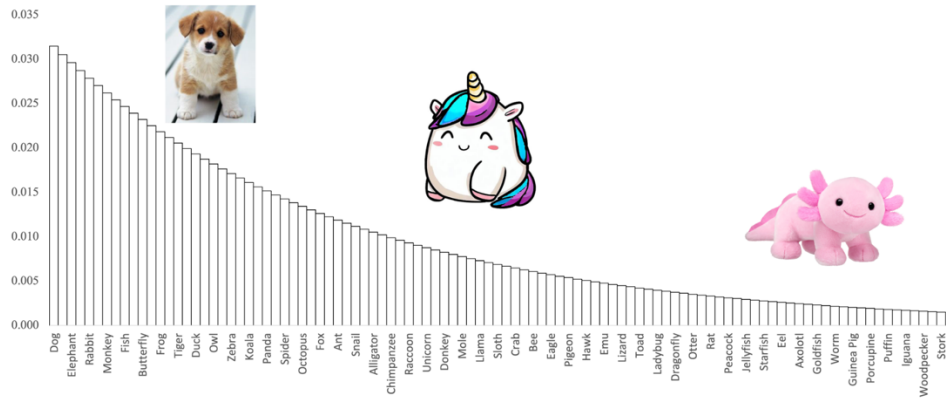
New Last Year

## 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 compression 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?

PS5 Wordle

You are playing a game of Wordle and you have narrowed down the possible words to these seven: bring, girls, storm, tears, rates, grind, agirt. What word should we guess next?

### Background on the game of Wordle

The goal of Wordle is to guess the "daily word" which is always a five letter word. You make a series of guesses, and after each guess you will get feedback (which letters were correct but in the wrong place, or correct and in the correct location). The details of Wordle feedback are not important for this problem, all that you need to know is that feedback can be represented as a string. It is important to be strategic when selecting a next guess!

### State of the game

Not all Wordle words are as likely. For the seven remaining words, here are their corresponding probabilities (note that this is a valid pmf, the probabilities sum to 1)

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

# KL Divergence



PS7 - Review GetMNeVNU

cs109psets.netlify.app/fall25/pset7/score\_guess

## Score a Probability Guess

A have created a game where a user has to guess a probability. You need to provide a score for how good that guess is. Write a scoring function

```
def score_guess(true_p, guess_p):
```

that takes:

- `true_p`: the true probability that a Bernoulli random variable equals 1
- `guess_p`: the user's guessed probability that it equals 1

How should we score their guess?

Let  $X \sim \text{Bern}(p = \text{true\_p})$  be the Bernoulli the user is trying to estimate

Let  $Y \sim \text{Bern}(p = \text{guess\_p})$  be the Bernoulli for their guess.

KL divergence between  $X$  and  $Y$  is a principled way to calculate how wrong the guess is (how much more surprised would you be if you used the guess probability instead of the true probability when observing Bernoullis from this distribution). However, KL Divergence is small when the guess is good, and large when the guess is bad. To reverse this property, return

$$\text{Score}(\text{true\_p}, \text{guess\_p}) = e^{-10 \cdot \text{KL}(X, Y)}$$

Where  $\text{KL}(X, Y)$  is the KL divergence between the two bernoullis. Each time you hit run, it will choose a value for true probability, and then plot the score you gave (and what we expected)

### Base for Log

For this problem you should use  $\log_e$  instead of  $\log_2$  for your KL divergence calculation. For all entropy related calculations (Entropy, KL Divergence, Mutual Information) either base is generally accepted, as long as you are consistent. What is standard? In Shannon information theory, base 2 (bits) is standard. In statistics and machine learning, natural logs (nats) are more common because they simplify calculus and align with maximum-likelihood estimation.

### Curiosities

Why use  $e^{-x}$  to transform kl divergence into a score? It turns high divergence into a low score, and low divergence into a high score. See [Graph of  \$e^{-x}\$](#) . Why the -10 in the score function? It penalizes bad guesses more harshly (play around with the constant to see how the function changes).

Previous Question      Next Question

Answer Editor      Solution

Python:

```
1 def score_guess(true_probability, guess_probability):
2     # TODO: your code here
3     return 0
4
```

Run

True probability (p): 0.777  
distance to solution: 31.2103

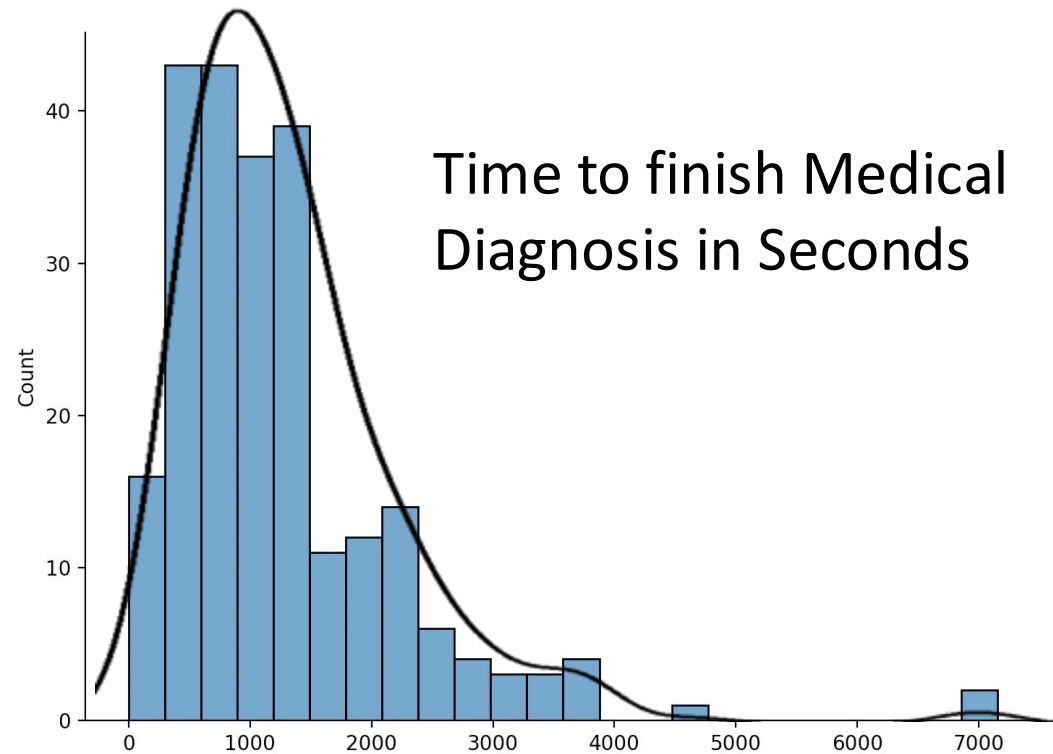
True Probability (p)	Guess Probability (Y)	Your score_guess function	exp(-10 * KL(X, Y))
0.777	0.777	0.0	1.0
0.777	0.7	0.0	0.8
0.777	0.8	0.0	0.8
0.777	0.6	0.0	0.4
0.777	0.9	0.0	0.4
0.777	0.5	0.0	0.1
0.777	0.95	0.0	0.05
0.777	0.4	0.0	0.05
0.777	0.99	0.0	0.01
0.777	0.01	0.0	0.01

# Machine Learning



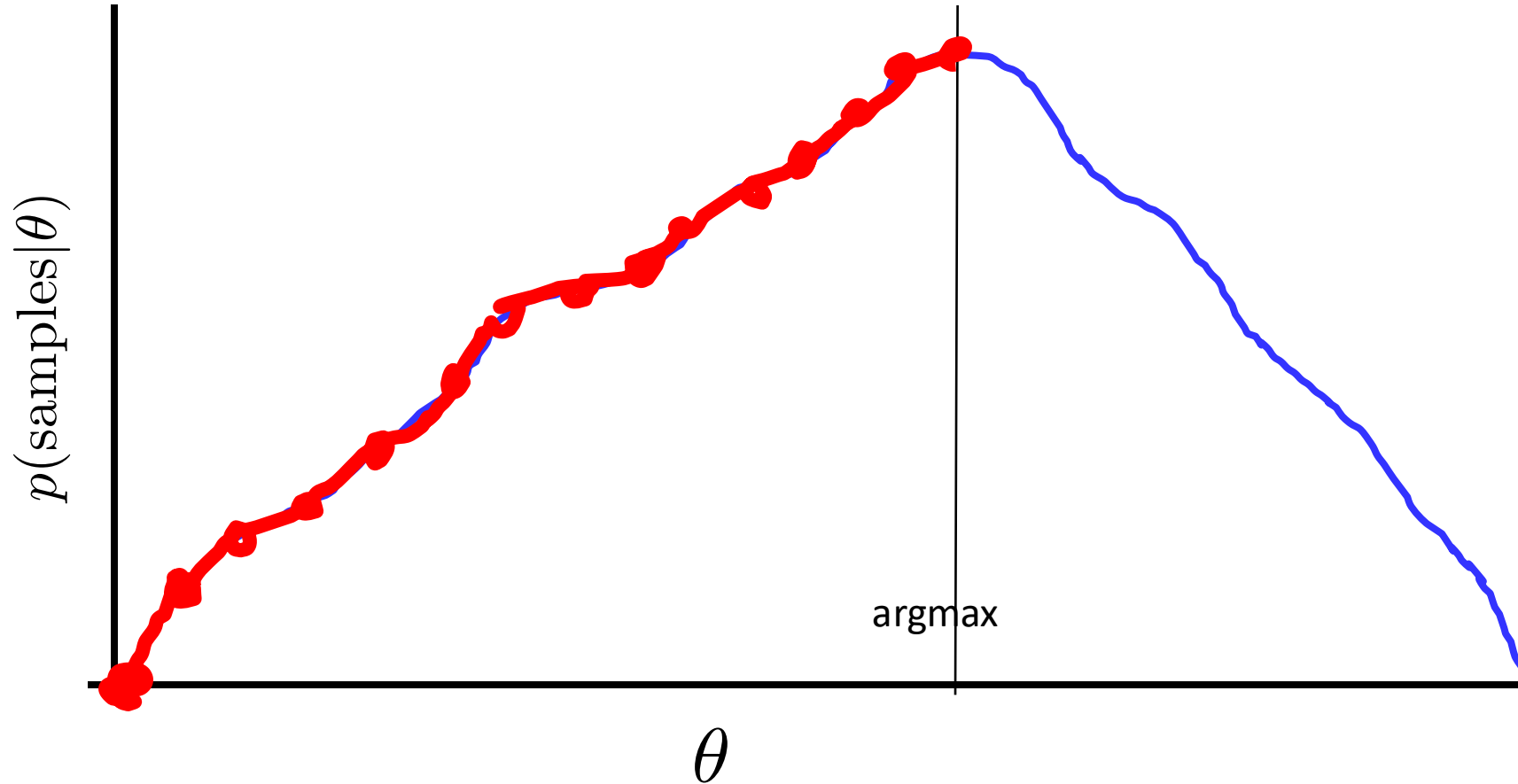
# MLE for Choosing Params

```
[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]
```



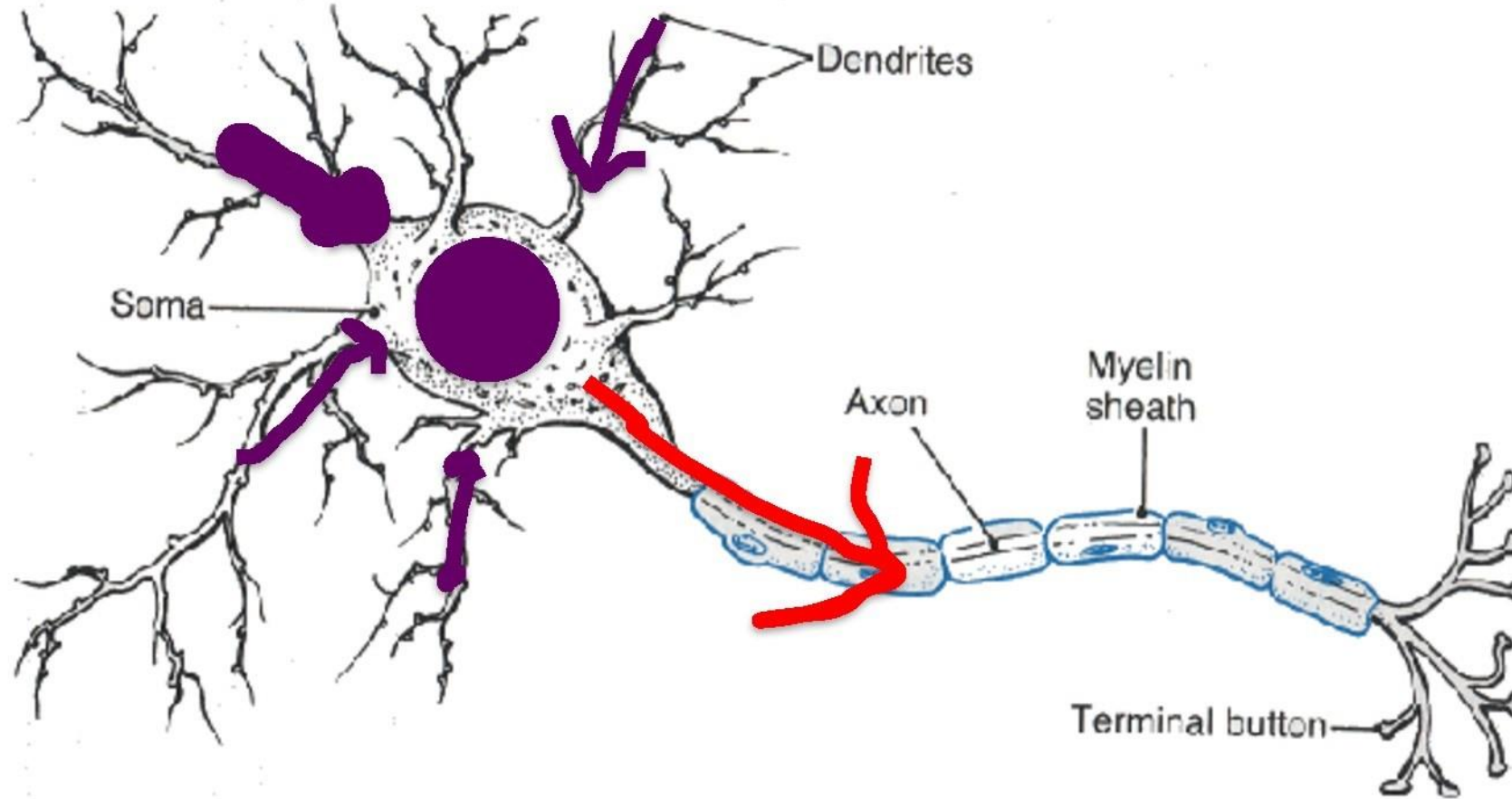
$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}$$

# Gradient Ascent



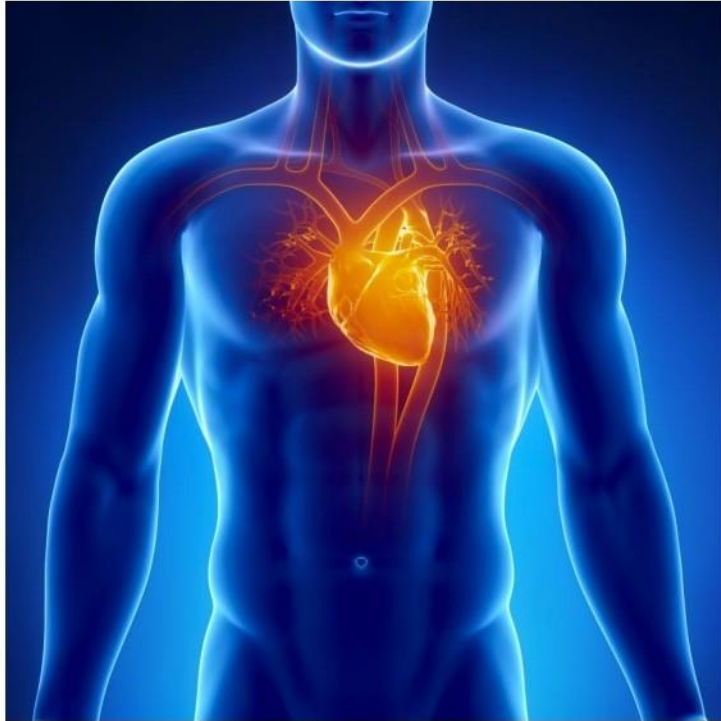
Walk uphill and you will find a local maxima  
(if your step size is small enough)

# Logistic Regression



# Machine Learning

Heart



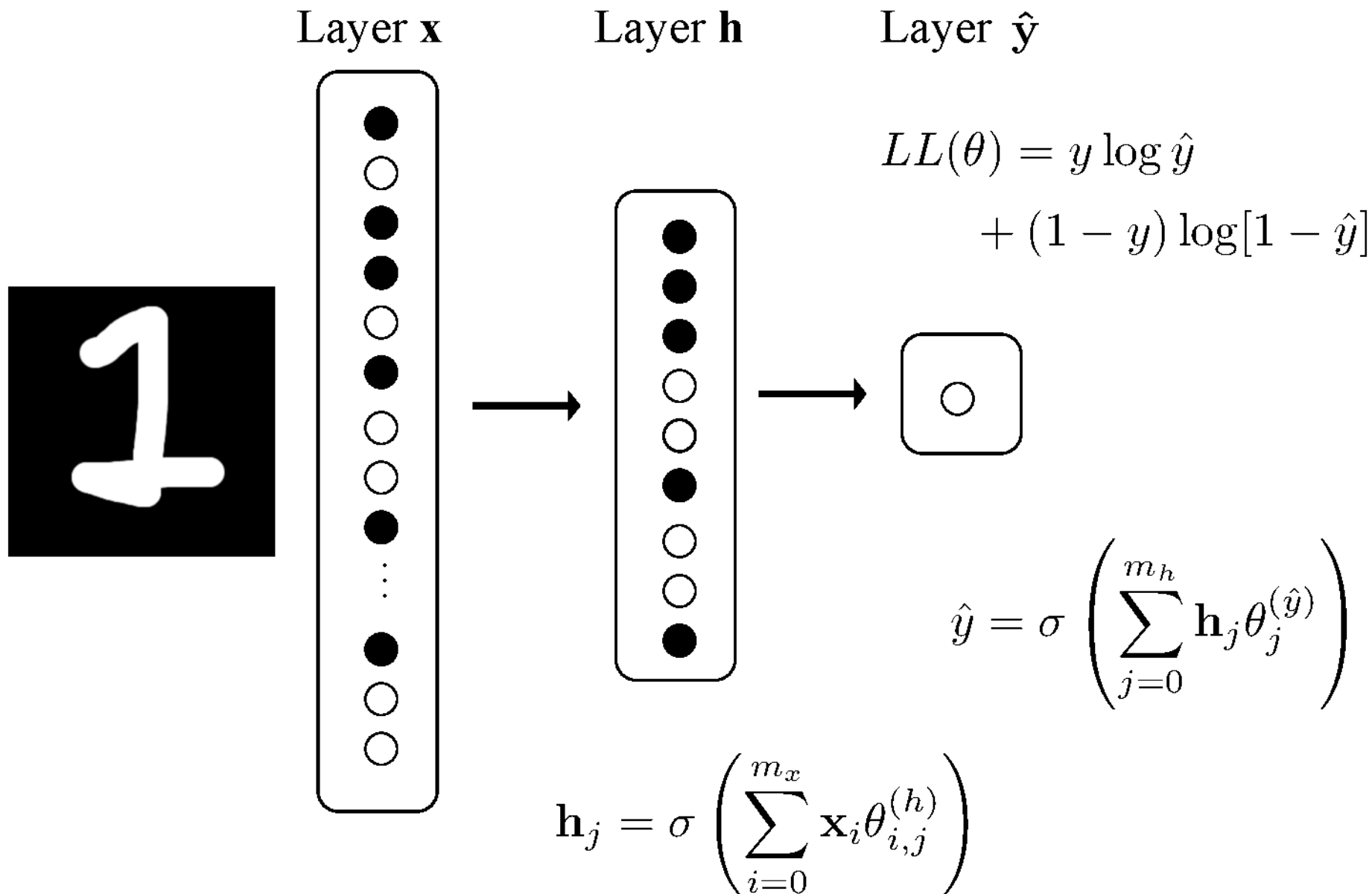
Ancestry



Netflix

**NETFLIX**

# Deep Learning



# Calibration

New!

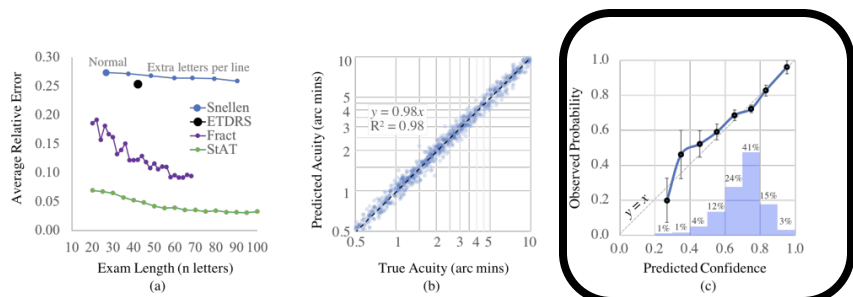


Figure 4: (a) The tradeoff between length of exam and error for the different algorithms. (b) Visualization of the predictions made by StACT. (c) Calibration test: StACT confidences correspond to how often it is correct.

## 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

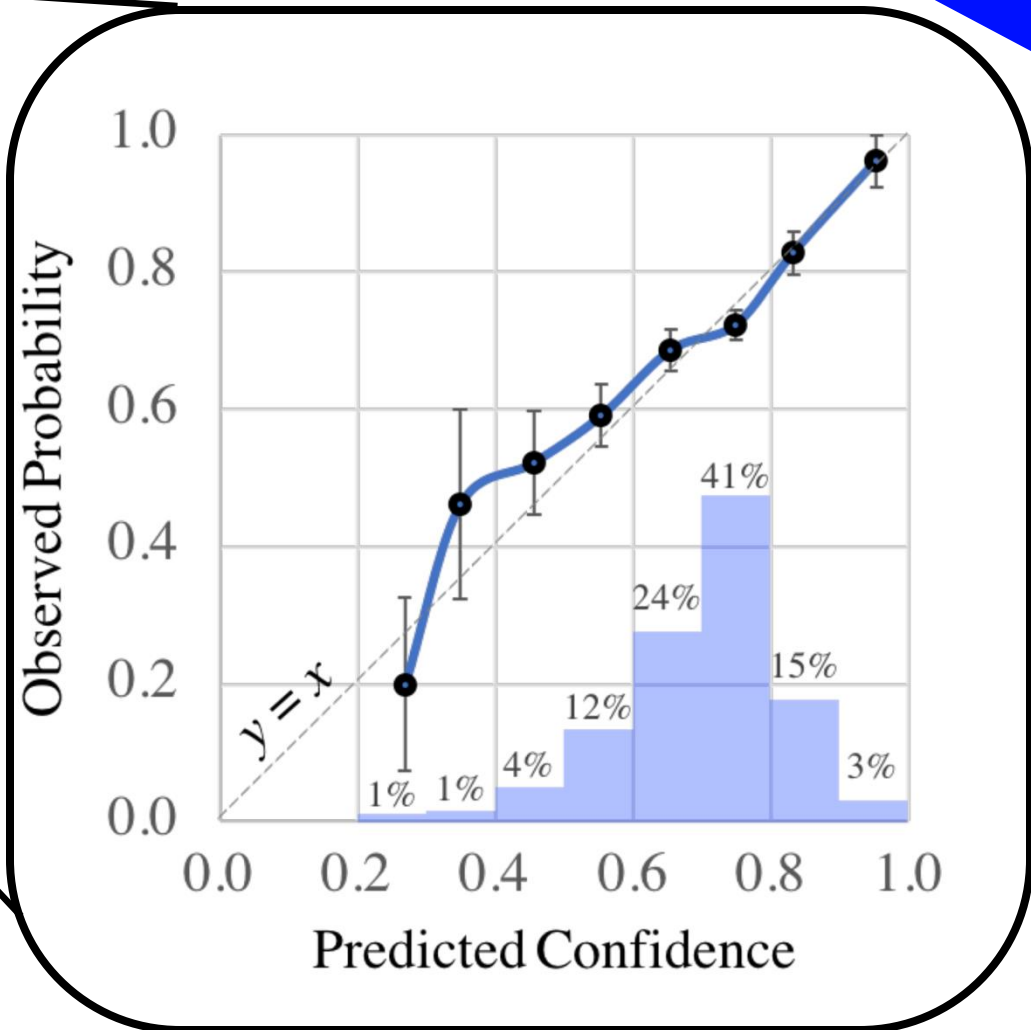
Table 1: Average relative error for each algorithm. Except for Snellen each test was allowed 20 letters. Results are average relative error after 1000 tests. <sup>†</sup> Snellen and ETDRS used 19 unique optotypes.

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.

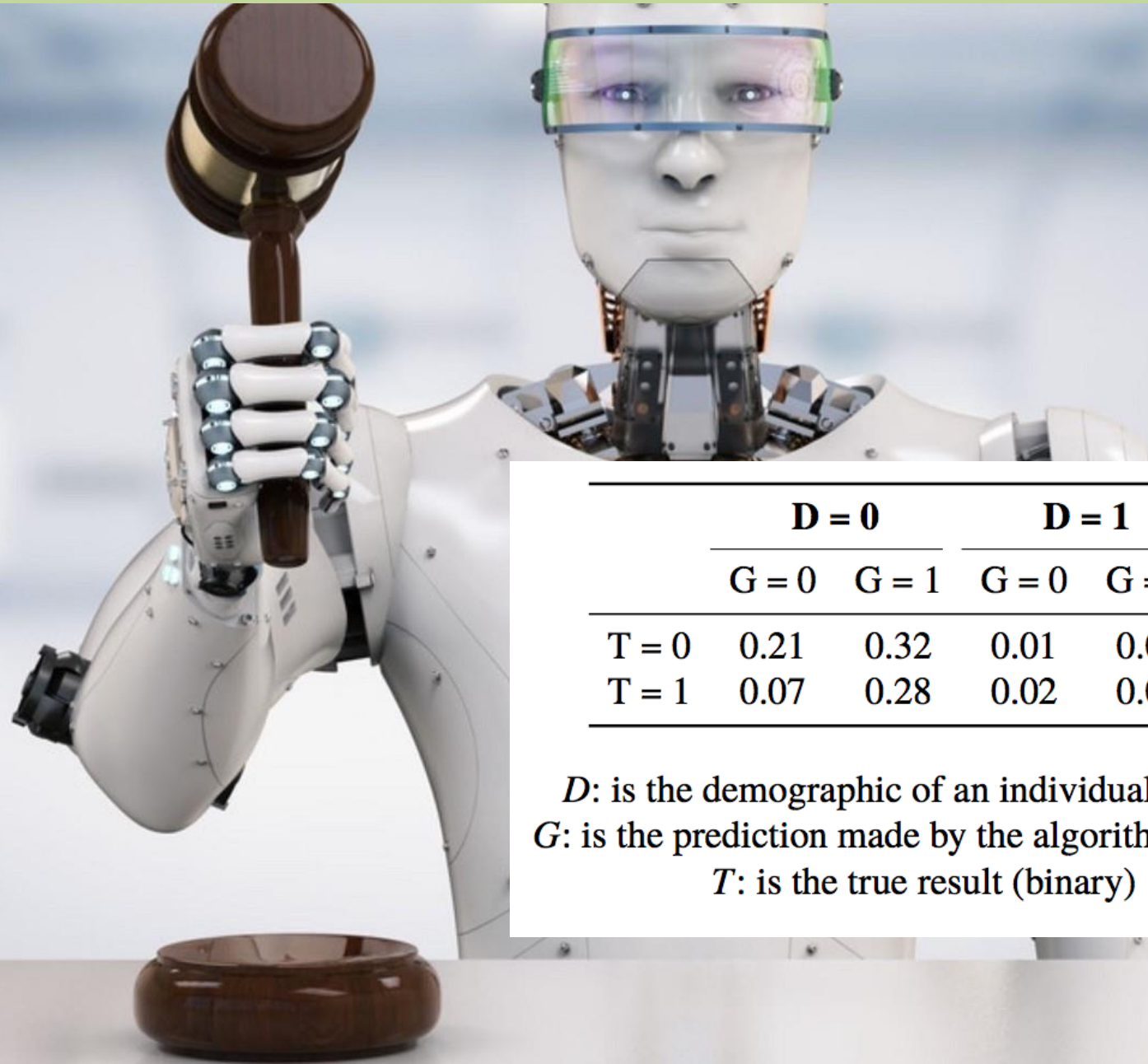
Figure 4 (b) visualizes what StACT's small relative error means in terms of predictions. Each point in the plot is a single patient. The x-axis is the true acuity of the patient and the y-axis is the predicted accuracy. We can qualitatively observe that the predictions are often accurate, there are no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

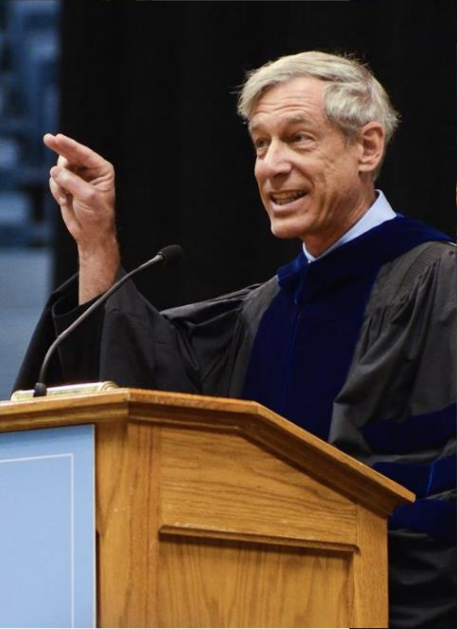
The Kaggle Champion  
←



Reinforcement!

# Night Sight

New!

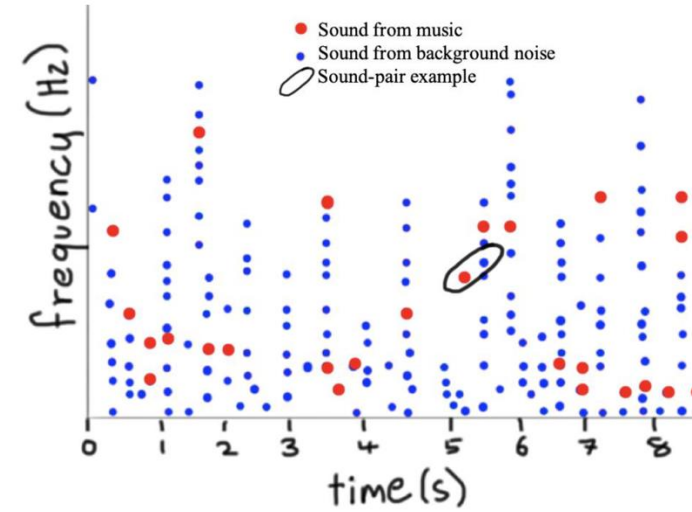


Mark Levoy, Stanford Emeritus Professor

<https://static.googleusercontent.com/media/hdrplusdata.org/en//hdrplus.pdf>

# Wisdom of the Crowds

New!



In 1999, what animal was taken off the U.S. Endangered species list after 29 years?

A:

B: Peregrine Falcon

C: Humpback Whale

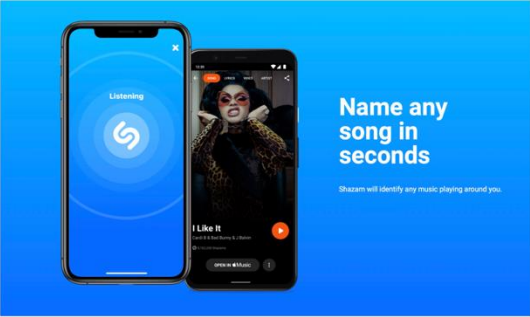
D:

# Review PSet

New!

PS7 Shazam

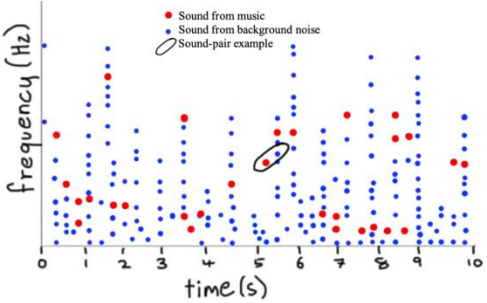
Shazam is trying to detect which song is playing in a noisy bar. We will explore how "wisdom of the crowds" effects allow it to accurately identify the song even when there is far more noise than music.



When Shazam runs its acoustic fingerprinting algorithm on an audio recording, it extracts many short "notes", represented as (frequency, time) tuples, from the recording.

You are in a really loud restaurant. In the current audio recording, there are 5000 notes from background noise and **only** 25 notes from the song. Is it still possible to identify the song?

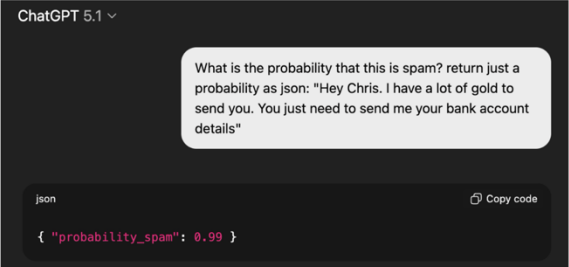
Shazam lets every pair of notes vote for which song it thinks is playing. There are (5025 choose 2) such pairs.



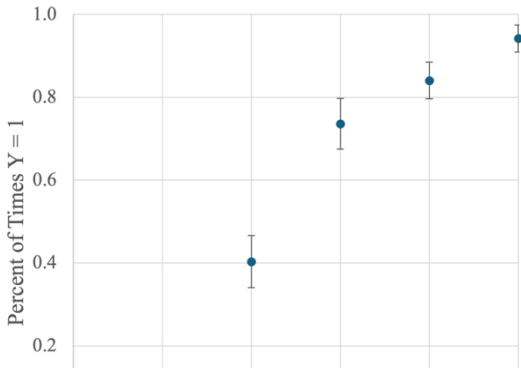
53

PS7 Calibrating ChatGPT

Imagine you use ChatGPT (though you could use any prediction for that matter) to make a prediction for a datapoint with features  $x$  and ChatGPT responds to your prompt with a probability that the label is a 1,  $P(Y = 1|X = x)$ . You hesitate. Are those probabilities trustworthy? In a situation like this, it might be really important to also know how *calibrated* the probabilities are. For example, if ChatGPT says that  $P(Y = 1|X = x) = 0.8$  does that mean there really is an 80% chance the label is a 1?




More formally, how can we check if the probabilities output from a model are "calibrated"? To evaluate if a model is calibrated we first group datapoints together based on the  $P(Y = 1|X = x)$  probability output given by the model. Then, among those groups we check how often the label is 1. Here is an example of calibration curve which plots probability groupings on the x-axis and fraction of times that the label was 1 on the y-axis:



PS7 Bayesian Flo

As we saw in section Flo models each user's cycle lengths! In this problem we are going to revisit the same task, but this time we are going to approach parameter estimation from a Bayesian perspective.



We will slightly simplify the formula to make the math less long. Assume that the length of someones period  $X$  is given by the following pdf which is based off a single parameter  $\beta$ :

$$f(X = x | B = \beta) = \beta(x - 27)^{\beta-1}e^{-(x-27)^\beta}$$

For a new user, you only observe **two** cycle lengths  $X_1 = 28$  and  $X_2 = 29$ , which means that you don't have enough data to estimate  $\beta$  using MLE (in section we used MLE, but we had many more cycles)! How can we estimate  $\beta$  in this low data situation? Inference. Just like in the Beta distribution derivation.

Good news! Based on historical data, we have a strong prior on the values that  $\beta$  can take! For thousands of past users, we have learned individual values for  $\beta$ . We then treat those values as samples from a random variable  $B$ . After analysing all the data, we found that  $B$  is a Gamma. We then used MLE to fit the Gamma distribution which resulted in the following nice equation for our prior belief in  $B$ :

$$P(B = \beta) = \frac{1}{2} \beta^2 e^{-\beta}$$

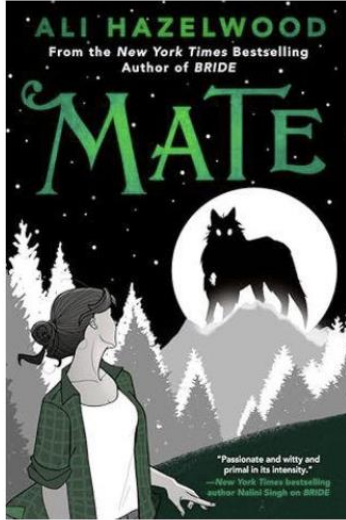
Given the two observations, use inference to compute your updated belief in  $B$

$$P(B = \beta | X_1 = 28, X_2 = 29)$$

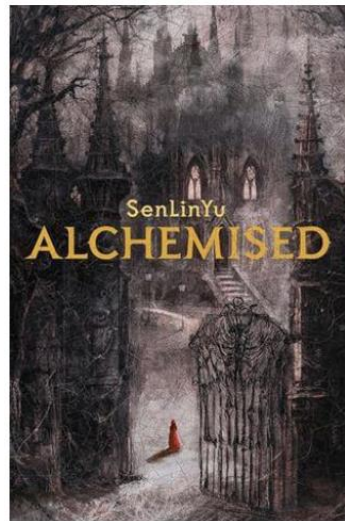
Allow for a normalization constant in your answer. You can assume that the two observations ( $X_1$  and  $X_2$ ) are independent if you know the value  $B$ .

Answer Editor  
Numeric Answer:  
Explanation:  
Block LaTeX

# Which Book?



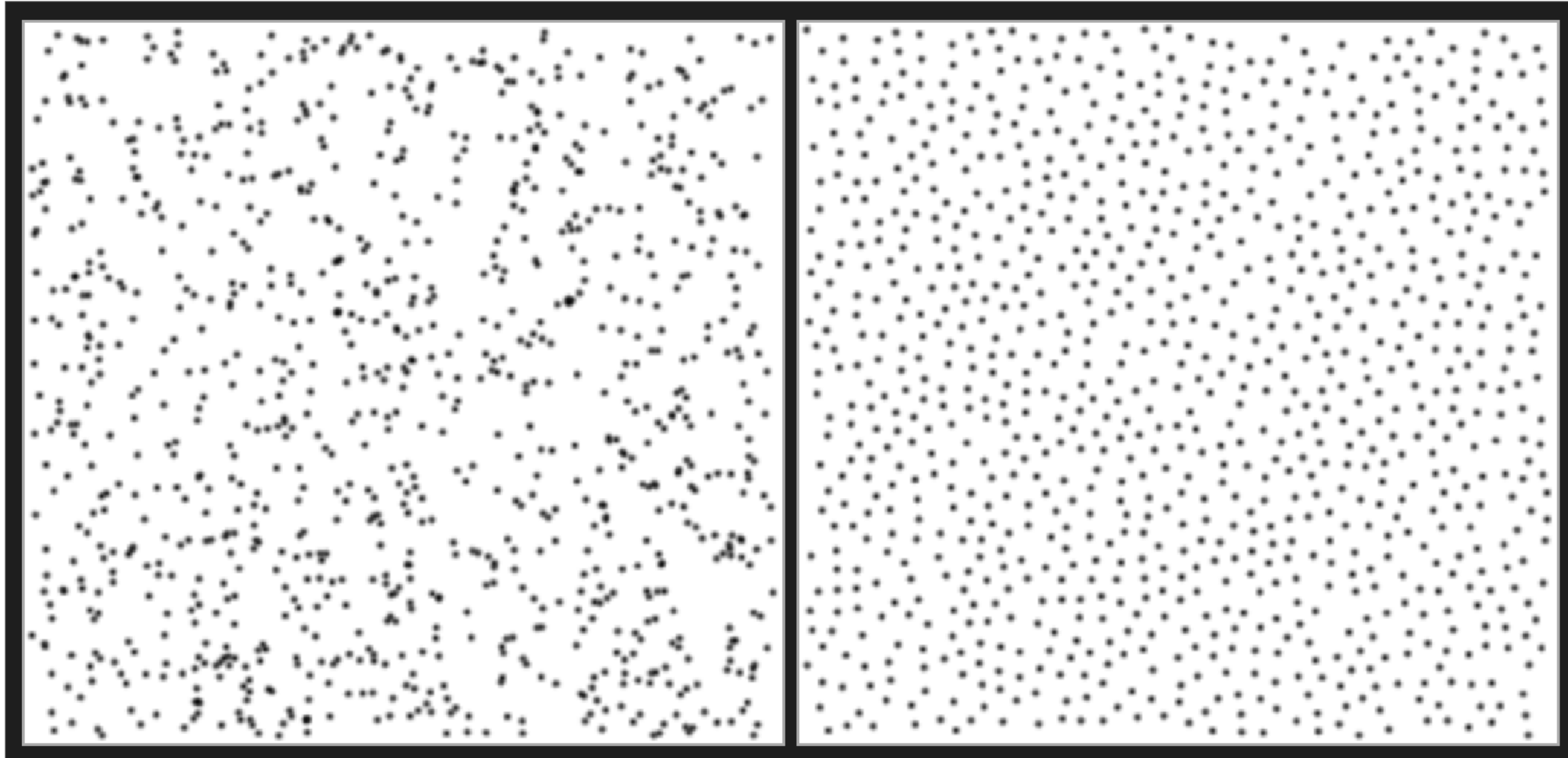
Rating	1	2	3	4	5
Count	0	0	0	1	4



Rating	1	2	3	4	5
Count	0	1	2	20	200

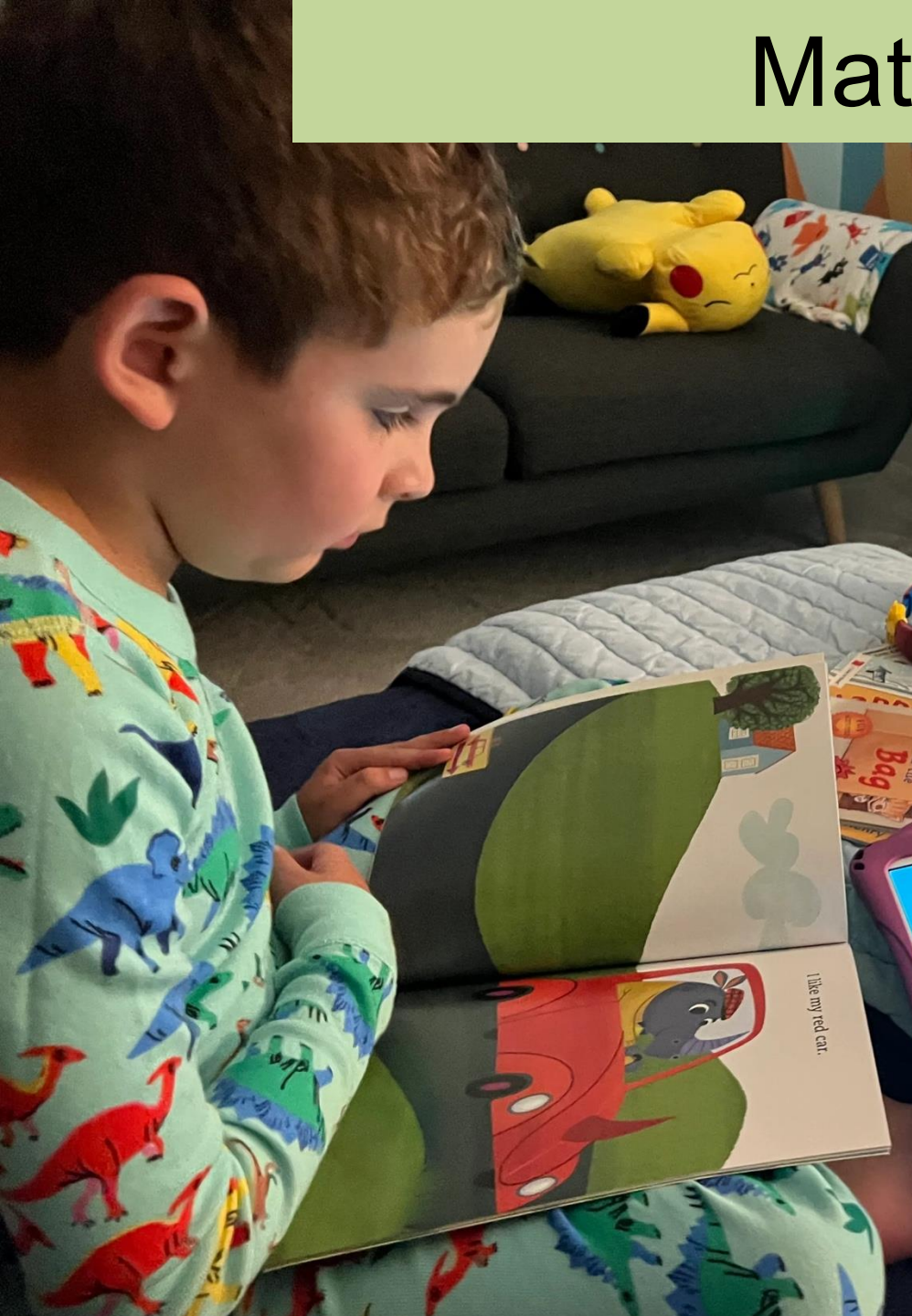
# Which is Poisson?

New!



# Math -> Prompt

New!



## 2. What an Ideal Classifier Looks Like

We compute the true conditional probability  $P(I = i | x, y_1, \dots, y_N)$ . Using conditional probability,

$$P(I = i | x, y_1, \dots, y_N) = \frac{P(x, y_1, \dots, y_N, i)}{P(x, y_1, \dots, y_N)}.$$

Because the distractors  $Y_j$  for  $j \neq i$  are drawn independently from  $P(Y)$  and independently of  $X$ ,

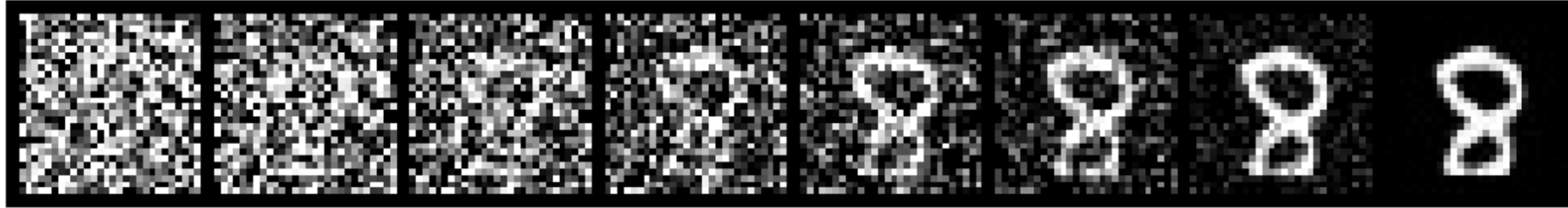
$$P(x, y_1, \dots, y_N, i) = \frac{1}{N} P(x, y_i) \prod_{j \neq i} P(y_j).$$

The denominator is a sum over all possible positions of the true record:

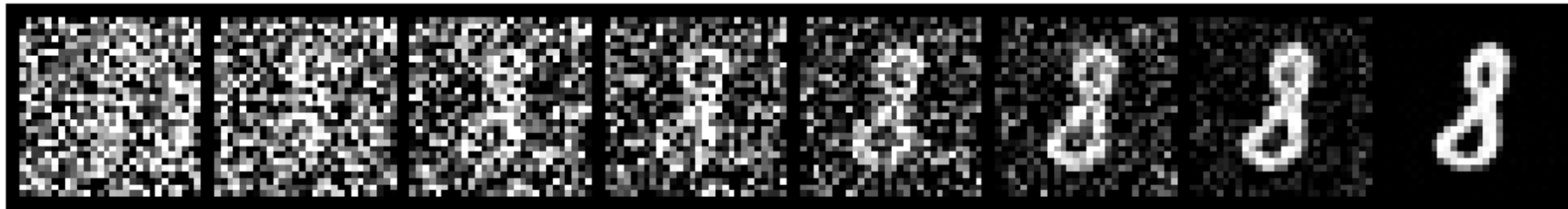
$$P(x, y_1, \dots, y_N) = \sum_{k=1}^N \frac{1}{N} P(x, y_k) \prod_{j \neq k} P(y_j).$$

# Diffusion!

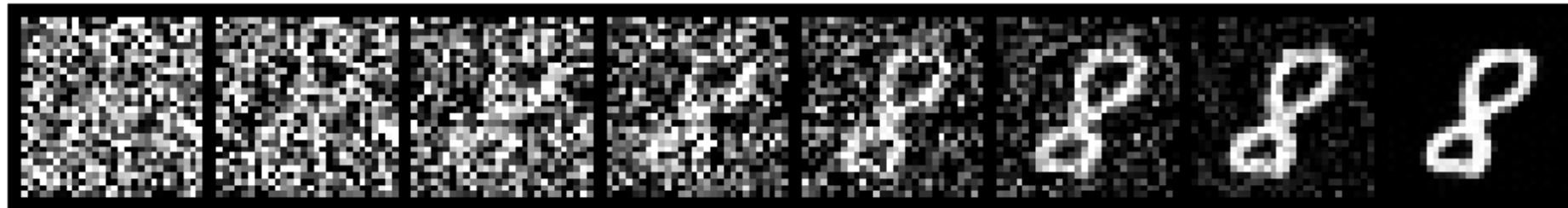
Reverse denoising (class 8)



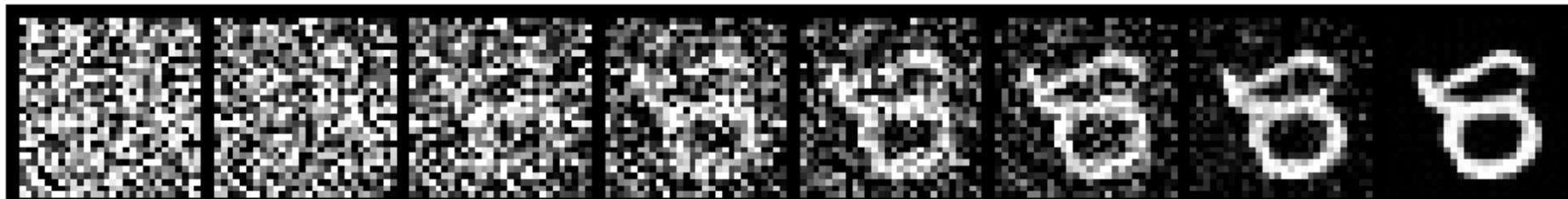
Reverse denoising (class 8)



Reverse denoising (class 8)



Reverse denoising (class 8)



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



# 1173 EdStem Answers

The screenshot shows a web browser window with the URL `edstem.org/us/courses/67646/discussion/5828924`. The page title is "CS 109 – Ed Discussion". The main content area is titled "Bernoulli MLE #745" and was posted 17 hours ago in the "Lectures" category. It has 68 views. The post content includes:

- Consider I.I.D. random variables  $X_1, X_2, \dots, X_n$
- $X_i \sim \text{Ber}(p)$
- Probability mass function,  $f(X_i = x_i | P = p)$

Two graphs are shown: "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 for the PMF are given as:

$$f(x_i | p) = p^{x_i} (1 - p)^{1 - x_i}$$
$$f(x_i | p = 0.2) = 0.2^{x_i} (1 - 0.2)^{1 - x_i}$$

The user's text says: "Committing this to memory since we went over it in class makes sense; I'm slightly worried though, I'm not sure if I would necessarily come up with this PMF function (on say, an exam) if I just saw 'here's a Bernoulli' and 'Do MLE on it'... would this be, and thus other possibly derivable yet slightly hairy PMFs that we haven't gone over, be provided to us?"

There is 1 answer from Anna Mattinger (TA) posted 4 hours ago:

Hi!

2 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?

Please feel free to follow-up if I'm not understanding your question correctly, but to answer what I think you're asking:

You'll definitely want to know the PMFs/PDFs of RV's that we've covered [i.e., ones on the RV Reference in the Course Reader]. If you don't have those memorized, that's okay—that's what the note sheet on the

# 15 New Course Reader Chapters

New Chapters

- any Binomial Problems
- inning Series
- pproximate Counting
- y Selection
- ading Eye Inflammation
- ades are Not Normal
- urse of Dimensionality
- gorithmic Art
- me of Ur
- andom Variables Practice
- Part 3: Probabilistic Models
- at Probability
- rginalization
- ltinomial
- ntinuous Joint
- erence
- yesian Networks
- ependence in Variables
- relation
- neral Inference
- me to Age
- ries
- firmness in Artificial Intelligence
- ederalist Paper Authorship
- obability of Baby Delivery
- yesian Carbon Dating
- ramid Hidden Chambers
- igital Vision Test
- idge Distribution
- pectation of Sum Proof
- yesian Viral Load Test
- 109 Logo
- acking in 2D
- Probabilistic Models Practice
- Part 4: Uncertainty Theory
- a Distribution
- ding Random Variables
- tral Limit Theorem
- ampling
- otstrapping
- orithmic Analysis
- ormation Theory
- tance Between Distributions
- ries
- ompson Sampling
- ght Sight
- Hacking
- fferential Privacy
- ncertainty Theory Practice
- Part 5: Machine Learning
- ameter Estimation
- ximum Likelihood Estimation
- ximum A Posteriori
- chine Learning
- stic Regression
- ve Bayes
- luating Classifiers
- ear Regression
- chine Learning in Python
- fusion
- ries
- LE Demos
- LE Pareto Distribution
- LE Mixture Model
- Machine Learning Practice
- Drafts
- udo Random Numbers
- n Response Theory
- ny Dice Rolls
- Central Limit Theorem Proof

## CS109 Logo

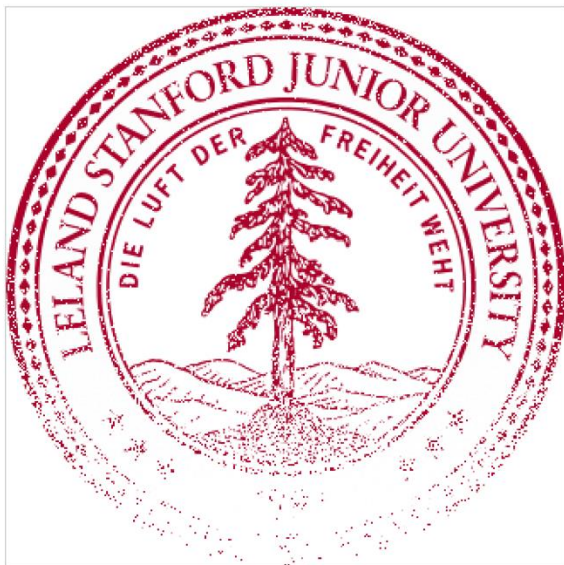
Did you know that we used a stochastic process to generate the CS109 logo? Throw 300,000 darts at a picture of the Stanford seal and only keep the pixels that are hit by at least one dart. Each dart has its x-pixel and y-pixel chosen independently at random from gaussian distributions:

Let  $S$  and be a constant that equals the size of the logo (its width is equal to its height). For the Stanford seal  $S = 300$ .

Let  $X$  be a random variable which represent the x-pixel.  $X \sim \mathcal{N}(\mu = \frac{S}{2}, \sigma^2 = (\frac{S}{2})^2)$

Let  $Y$  be a random variable which represents the y-pixel.  $Y \sim \mathcal{N}(\mu = \frac{S}{2}, \sigma^2 = (\frac{S}{2})^2)$

You can re-run the process by pressing the Play animation button:



Play animation

Throw 5,000 darts

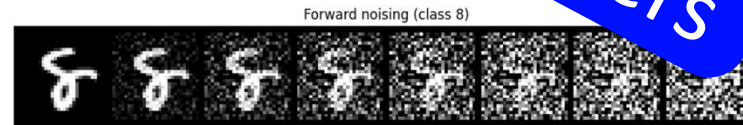
Reset animation

Darts thrown: 300,000 / 300,000

Here is the probability density function of  $X$  (left) and  $Y$  (right) on their own:



Here is what adding noise looked like for my hand draw 8s:



## The Reverse Process

We have now redefined our task. Could we learn to take a square image with randomly selected pixels, and learn to reverse the noise adding task until we are left with a picture of a tree? To reverse this process, we need to learn the conditional distribution  $P(x_{t-1}|x_t)$ . Here's the surprising part:

### Diffusion Critical Fact: Reverse Process is also Gaussian

If the noise variance  $\sigma^2$  is small enough, the distribution of  $X_t|X_{t+1}$  can be approximated as:

$$X_t|X_{t+1} \sim N(\mu, \sigma^2),$$

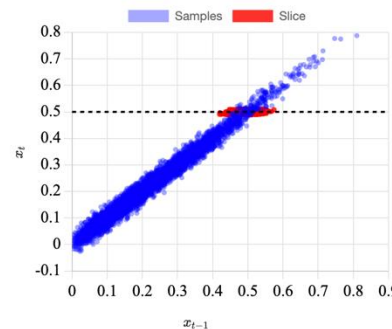
where:

- $\mu$  is the mean of the Gaussian, which is not known (but we assume is a complex function of the image)
- $\sigma^2$  is the variance of the noise that we selected when adding in Gaussian noise to our images

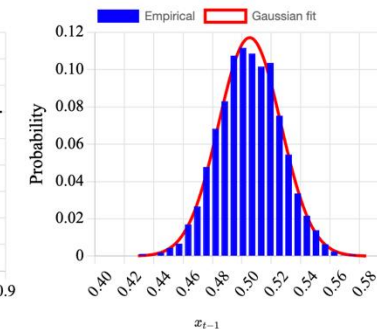
Here is a demo of the Diffusion Critical Fact. In this demo I first sample half a million values from  $X_t \sim \text{Beta}(2, 7)$  (the critical fact doesn't depend on the distribution of  $X_t$ ). For each point I then chose a value  $X_{t+1}$  using the diffusion process:  $X_{t+1} = X_t + N_t$  where  $N_t \sim N(0, \sigma^2)$ .

On the left is the joint of  $X_t$  and  $X_{t+1}$ . On the right is the posterior of  $X_t$  after observing that  $X_{t+1} = k$ . Notice how Normal it looks when  $\sigma$  is small. Then try setting  $\sigma$  to be 2 and see that the Normal shape goes away.

Joint  $P(X_{t-1} = x_{t-1}, X_t = x_t)$



Posterior  $P(X_{t-1} = x_{t-1} | X_t \approx k)$



k: 0.5     $\sigma$ : 0.02    Slice width: 0.01    Samples in slice: 3600

A proof sketch of the Diffusion Critical Fact is provided later on in this chapter. It uses inference (and Taylor Series

# 52+ Personal Challenges



# 1 New Counter on the Way



I had an important job!  
I hope you think I did it justice

Wild time. Incredible school at  
which to study probability

thank you

The image features the words "thank you" spelled out using ten light-colored wooden blocks, each with a single lowercase letter in a bold, black, sans-serif font. The blocks are arranged in a single horizontal line on a dark wooden surface. The background is a soft-focus bokeh of numerous warm, golden-yellow circular lights, creating a warm and inviting atmosphere.