

# Beyond CS109

CS109, Stanford University

# Final Exam Logistics

**Where ?? Hewlett 200**  
(this exact room !!! )

**When ?? 8:30am next Tuesday.**

- **No phones** (honor code violation to have a phone on or near your person during exam)
- **Assigned seating**
- **6 pages** of notes
- **Cumulative**
- If you have **OAE** you should have already contacted us!
- **Review on Friday** (thank you Isabel!)

Probability gives you a new  
lens on the world

Can I tell you the story  
about how it influenced my  
life?

# My Life Goal is to Be a Teacher 😊

In Computer Science



**Juliette Woodrow**  
Ph.D Candidate at Stanford University  
jwoodrow@stanford.edu

I have been teaching in some capacity for over 15 quarters at Stanford.

I was the instructor for CS106A in 2021.

I have been a TA and a head TA for CS109.



## CS106A: Programming Methodologies

Stanford University | Summer 2021  
Monday, Tuesday, Wednesday, Thursday | Live Lectures 1:30pm - 2:30pm PT

**TEACHING TEAM**

**Juliette Woodrow**



 Lecturer  
 jwoodrow@  
 Tues 9-10:30am  
 Thurs 9-10:30am

**ANNOUNCEMENTS**

**Quiz 2 Grades and Solutions**  
4 years ago

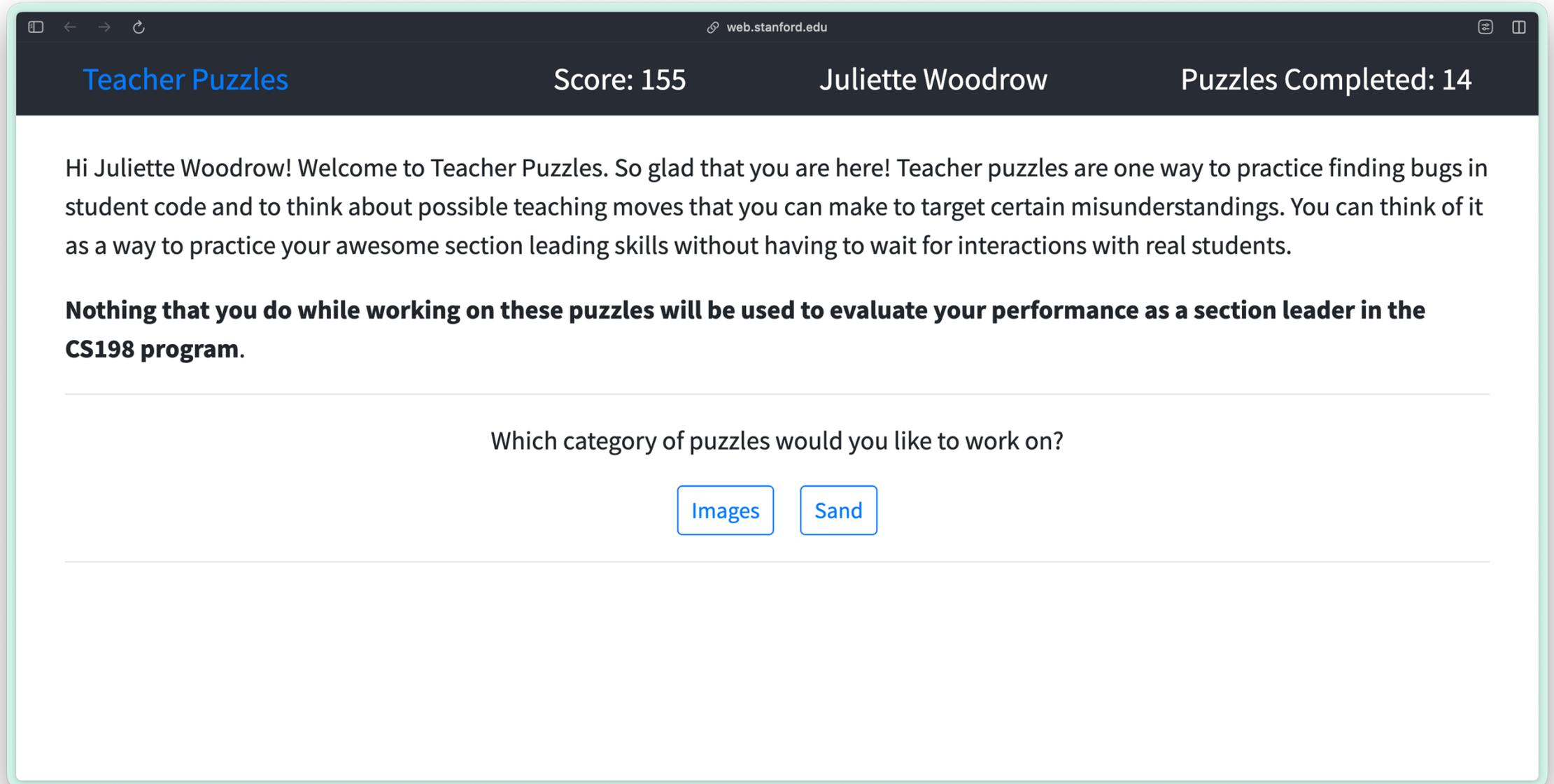
We have released grades for the second quiz on [Gradescope](#). Solutions as well as statistics about the exam can be found on the [Quiz 2 Solutions handout](#).

**Submitting Quiz 2**  
4 years ago

We are so sorry for the troubles submitting quiz 2. We are still not sure what



# My First Research Project



The screenshot shows a web browser window with the URL `web.stanford.edu`. The page has a dark header with the following information: **Teacher Puzzles** (in blue), **Score: 155**, **Juliette Woodrow**, and **Puzzles Completed: 14**. Below the header, there is a white content area with a welcome message and a question. The question asks which category of puzzles the user would like to work on, with two buttons: **Images** and **Sand**.

Teacher Puzzles

Score: 155

Juliette Woodrow

Puzzles Completed: 14

Hi Juliette Woodrow! Welcome to Teacher Puzzles. So glad that you are here! Teacher puzzles are one way to practice finding bugs in student code and to think about possible teaching moves that you can make to target certain misunderstandings. You can think of it as a way to practice your awesome section leading skills without having to wait for interactions with real students.

**Nothing that you do while working on these puzzles will be used to evaluate your performance as a section leader in the CS198 program.**

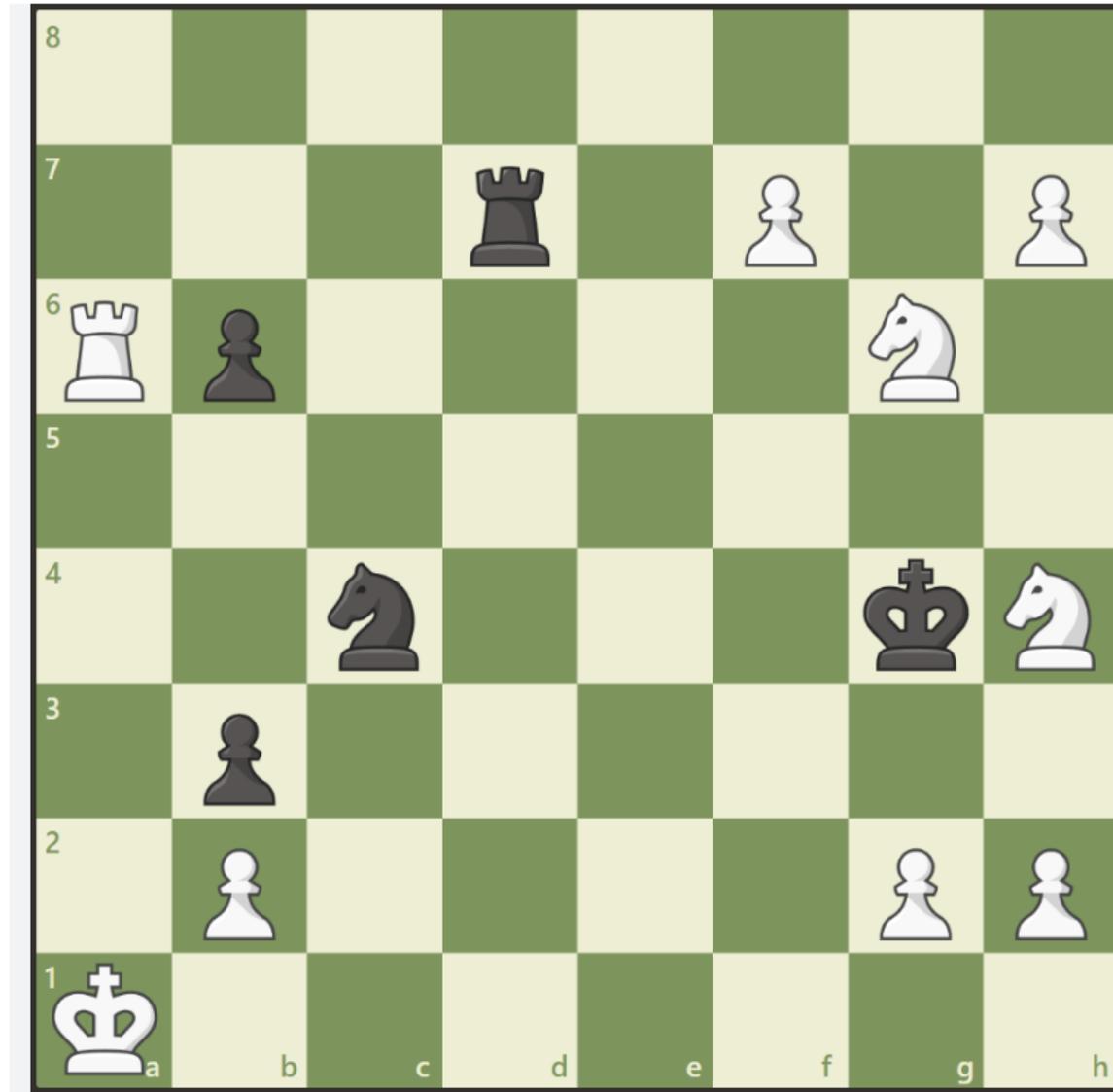
---

Which category of puzzles would you like to work on?

Images Sand

---

# Inspired by Chess.com Puzzles



# My First Research Project

(1) Teacher in training would identify the bugs.

Teacher Puzzles Category: Sand Score: 165 Juliette Woodrow Puzzles Co

Here is some code that a student might write to solve the given problem. They are coming to you for help similar to how a student would ask for your help at LaIR. Your task is to:

1. Find the bugs in the student's code
2. Identify the misunderstandings causing these bugs
3. Brainstorm a question that might help the student find a bug

**Problem Description:**  
Given grid and x\_from,y\_from and destination x\_to,y\_to. Assume there is sand at x\_from,y\_from. Is it possible to move to x\_to,y\_to? The from location is always in bounds, but the to location may not be. Return True if the move is ok, or False otherwise. Ok move: to location is in bounds, empty, not violating corner rule.

**Student Code:**

```
1- def check_move(grid, x1, y1, x2, y2):  
2-     if not grid.in_bounds(x2, y2):  
3-         return False  
4-     if grid.get(x2, y2) != 's':  
5-         return False  
6-  
7-  
8-     return True  
9-
```

**Step One:** Find the bugs in the student's code. You earn 10 points for every correct bug t

Does not check if destination is empty Only checks if destination is not sand Does not use an if statement for in bounds

Forgets to check if values are in bounds Unnecessary for loop Unnecessarily checks the corner Forgets to check the corner

Forgets to return True

Submit

You did not correctly identify all the bugs. The bugs are:  
*Forgets to check the corner Only checks if destination is not sand*

**Step Two:** Identify the misunderstandings causing these bugs. You earn 15 points for every correct misunderstanding you identify.

A cell is only legal to move to if it is empty meaning the value is None

The need to store and use the return value from grid.inbounds

The need to check for coordinate edge cases (making sure they are in bounds)

The goal of the function is just to check one location rather than the entire grid

The specific time when we need to check if the corner is free vs. when we do not need to check the corner

The need for a "corner case" check What/how to return when the move is valid

Submit

Buggy student code for CS106A

# My First Research Project

Teacher Puzzles Category: Sand Score: 165 Juliette Woodrow Puzzles Co

Here is some code that a student might write to solve the given problem. They are coming to you for help similar to how a student would ask for your help at LaIR. Your task is to:

1. Find the bugs in the student's code
2. Identify the misunderstandings causing these bugs
3. Brainstorm a question that might help the student find a bug

**Problem Description:**  
Given grid and x\_from,y\_from and destination x\_to,y\_to. Assume there is sand at x\_from,y\_from. Is it possible to move to x\_to,y\_to? The from location is always in bounds, but the to location may not be. Return True if the move is ok, or False otherwise. Ok move: to location is in bounds, empty, not violating corner rule.

**Student Code:**

```
1- def check_move(grid, x1, y1, x2, y2):
2-     if not grid.in_bounds(x2, y2):
3-         return False
4-     if grid.get(x2, y2) != 's':
5-         return False
6-
7-
8-     return True
9-
```

**Step One:** Find the bugs in the student's code. You earn 10 points for every correct bug t

Does not check if destination is empty Only checks if destination is not sand Does not use an if statement for in bounds

Forgets to check if values are in bounds Unnecessary for loop Unnecessarily checks the corner Forgets to check the corner

Forgets to return True

Submit

You did not correctly identify all the bugs. The bugs are:  
*Forgets to check the corner Only checks if destination is not sand*

**Step Two:** Identify the misunderstandings causing these bugs. You earn 15 points for  
you identify.

A cell is only legal to move to if it is empty meaning the value is No  
The need to store and use the return value from grid.inbounds  
The need to check for coordinate edge cases (making sure they are in bounds)  
The goal of the function is just to check one location rather than the entire grid  
The specific time when we need to check if the corner is free vs. when we do not need to check the corner  
The need for a "corner case" check What/how to return when the move is valid

Submit

(1) Teacher in training would identify the bugs.

(2) Guess which misconceptions the student has given these bugs.

Buggy student code for CS106A

# My First Research Project

Teacher Puzzles Category: Sand Score: 165 Juliette Woodrow Puzzles Co

Here is some code that a student might write to solve the given problem. They are coming to you for help similar to how a student would ask for your help at LaIR. Your task is to:

1. Find the bugs in the student's code
2. Identify the misunderstandings causing these bugs
3. Brainstorm a question that might help the student find a bug

**Problem Description:**  
Given grid and  $x\_from, y\_from$  and destination  $x\_to, y\_to$ . Assume there is sand at  $x\_from, y\_from$ . Is it possible to move to  $x\_to, y\_to$ ? The from location is always in bounds, but the to location may not be. Return True if the move is ok, or False otherwise. Ok move: to location is in bounds, empty, not violating corner rule.

**Student Code:**

```
1- def check_move(grid, x1, y1, x2, y2):  
2-     if not grid.in_bounds(x2, y2):  
3-         return False  
4-     if grid.get(x2, y2) != 's':  
5-         return False  
6-  
7-  
8-     return True  
9-
```

**Step One:** Find the bugs in the student's code. You earn 10 points for every correct bug t

Does not check if destination is empty Only checks if destination is not sand Does not use an if statement for in bounds

Forgets to check if values are in bounds Unnecessary for loop Unnecessarily checks the corner Forgets to check the corner

Forgets to return True

Submit

You did not correctly identify all the bugs. The bugs are:  
*Forgets to check the corner Only checks if destination is not sand*

**Step Two:** Identify the misunderstandings causing these bugs. You earn 15 points for  
you identify.

A cell is only legal to move to if it is empty meaning the value is No  
The need to store and use the return value from grid.inbounds  
need to check for coordinate edge cases (making sure they are in bounds)  
goal of the function is just to check one location rather than the entire grid  
when we need to check if the corner is free vs. when we do not need to check the corner  
ed for a "corner case" check What/how to return when the move is valid

Submit

(1) Teacher in training would identify the bugs.

(2) Guess which misconceptions the student has given these bugs.

(3) Make a teaching move !!

Buggy student code for CS106A

# My First Research Project

Probabilistically generate the next “step”  
the student would make given your  
“teaching move”.

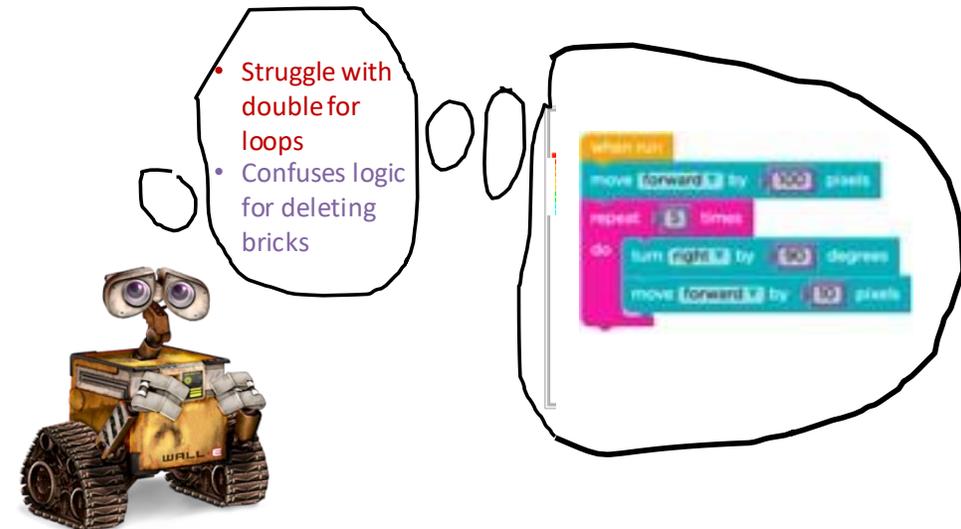
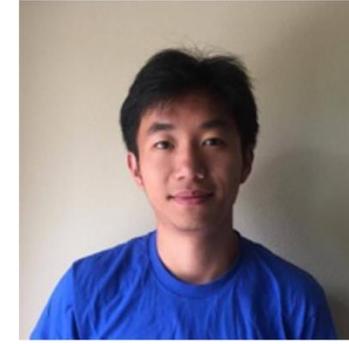
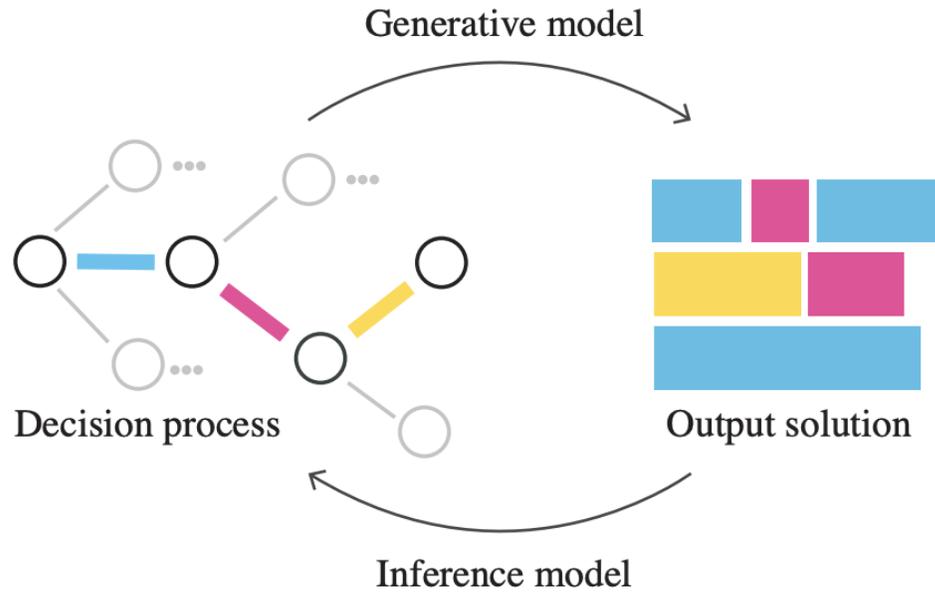
## Student Code:

```
1 def check_move(grid, x1, y1, x2, y2):
2     if not grid.in_bounds(x2, y2):
3         return False
4     if grid.get(x2, y2) != None:
5         return False
6     if grid.get(x1-1, y1) != None or grid.get(x1+1, y1) != None:
7         return False
8     return True
9
```

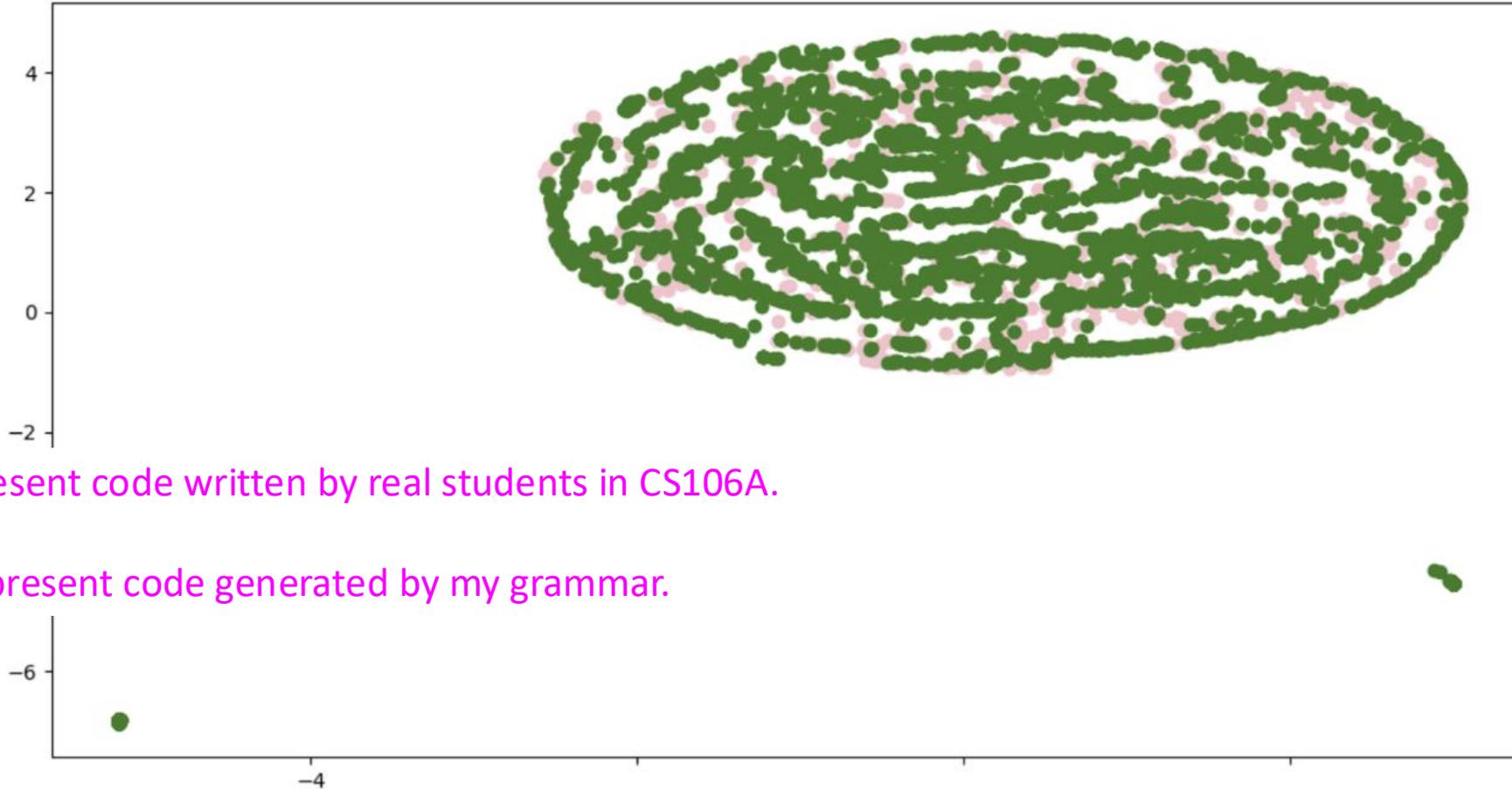
How did we do this?

# Powered by Generative Understanding

Label student code



# My First Research Project



Pink dots represent code written by real students in CS106A.

Green dots represent code generated by my grammar.

With a probabilistic grammar, we can capture the entire space of student errors (on this problem) !!!!  
So, if teachers in training can learn the bugs from our model, they will be able to help real students!

# Stanford Code in Place:



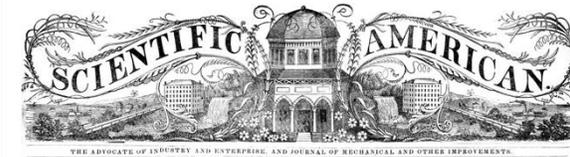
**5000+** section leaders teach

**60,000+** students

**CS106A**

**As Community Service**

Featured in



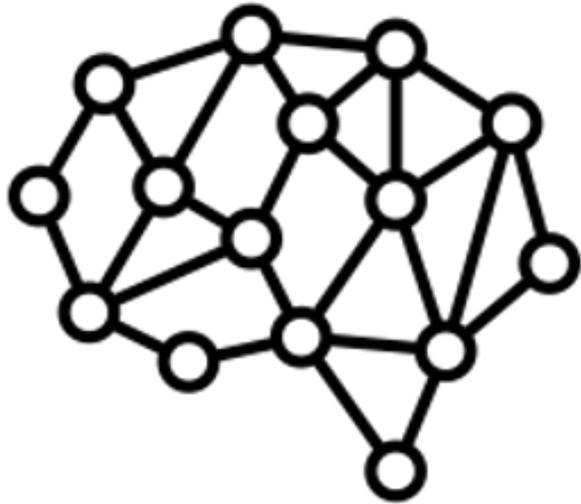
**SCIENTIFIC  
AMERICAN**



We do a lot of teaching in  
Code in Place!!

We also did a lot of cool AI  
things...

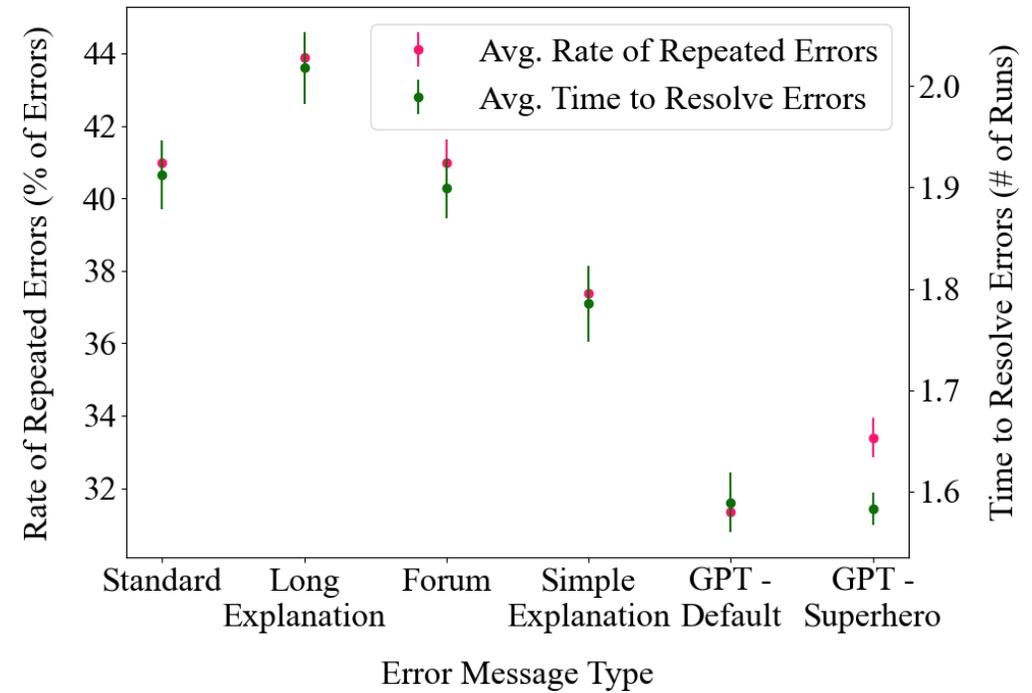
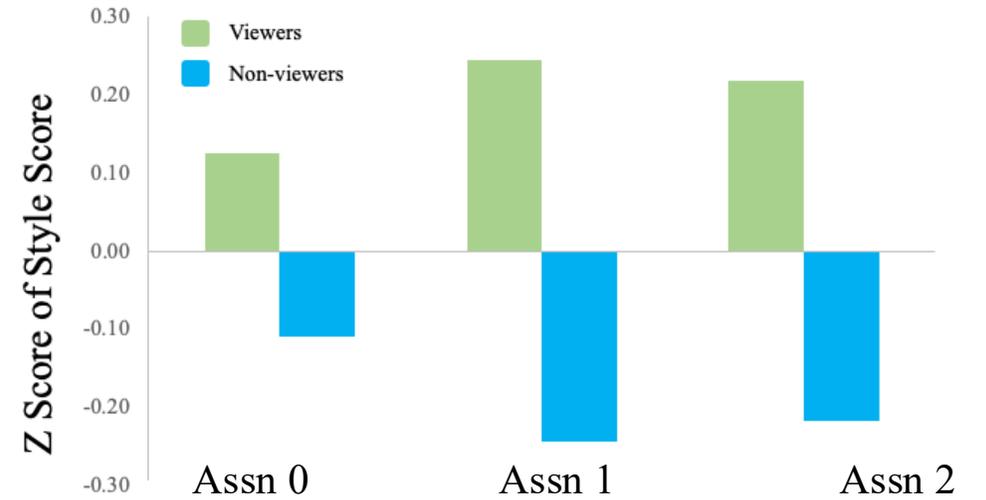
# A Tale of Two Ideas



AI powered  
interventions to  
support education  
at scale !!

# AI Realtime Feedback

The IDE interface displays a Python script for a brick game. The code includes imports for graphics, time, random, and math. It defines canvas dimensions and brick colors. The terminal shows a `SyntaxError: invalid syntax` message due to a missing colon in a function definition.



Both Accepted: SIGCSE 2024

# Unrestricted Access to GPT

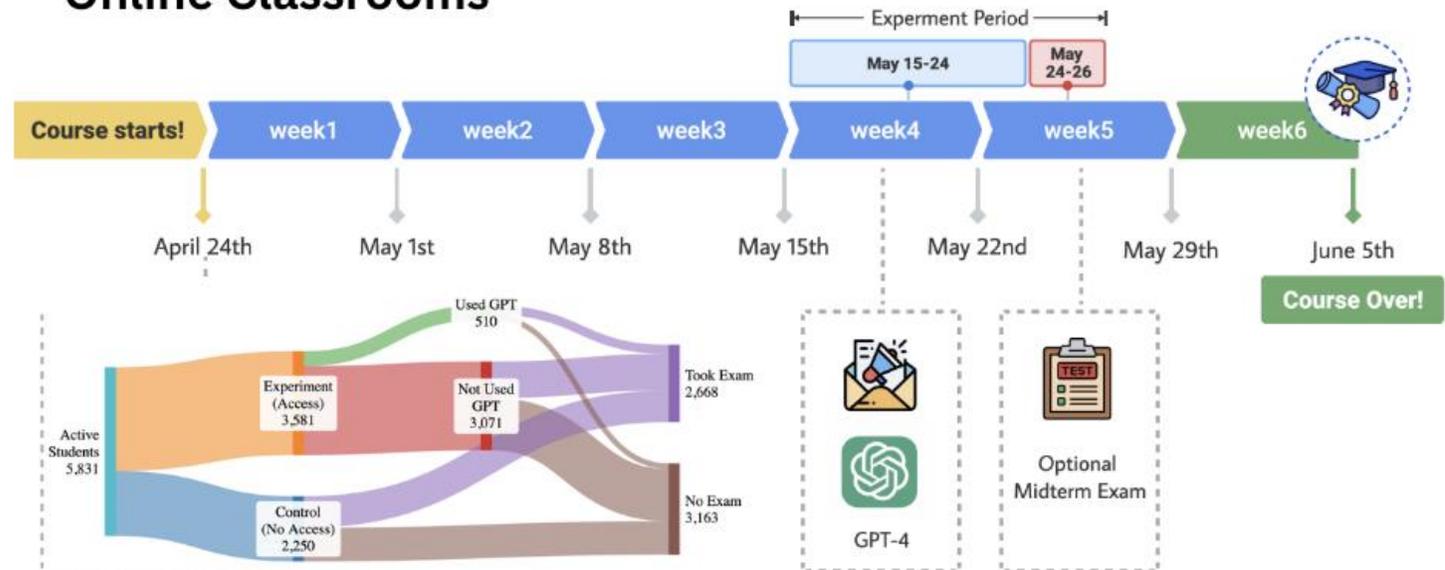
## The GPT Surprise: Offering Large Language Model Chat in a Massive Coding Class Reduced Engagement But May Increase Adopters' Exam Performances



Allen Nie, Yash Chandak, Miroslav Suzara, Ali Malik, **Juliette Woodrow**, Matt Peng, Mehran Sahami, Emma Brunskill, and **Chris Piech**

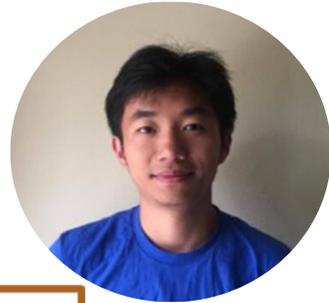


### Randomized Control Trial of General-Purpose AI Use in Online Classrooms



# AI Grading of Exams (before GPT)

SL filled rubrics were used to train an AI based feedback system.



**Rubric Item: Problem Setup**

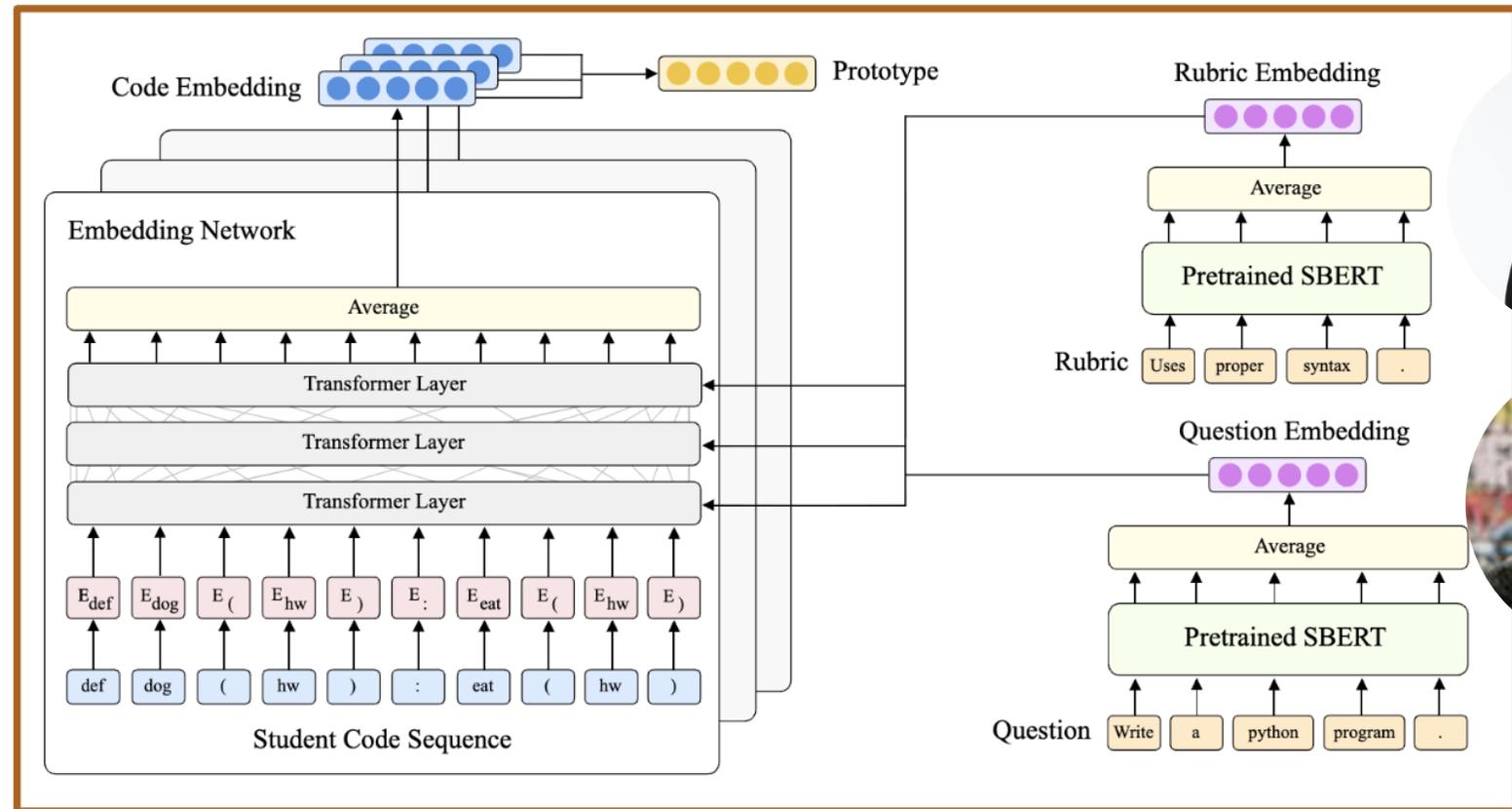
- Perfect
- Minor issue
- Major issue
- No attempt

**Rubric Item: General Deductions**

- Perfect
- >2 syntax errors
- 1 syntax error
- Variable scoping issue
- 2 syntax errors
- Null pointer exception

**Rubric Item: String Insertion**

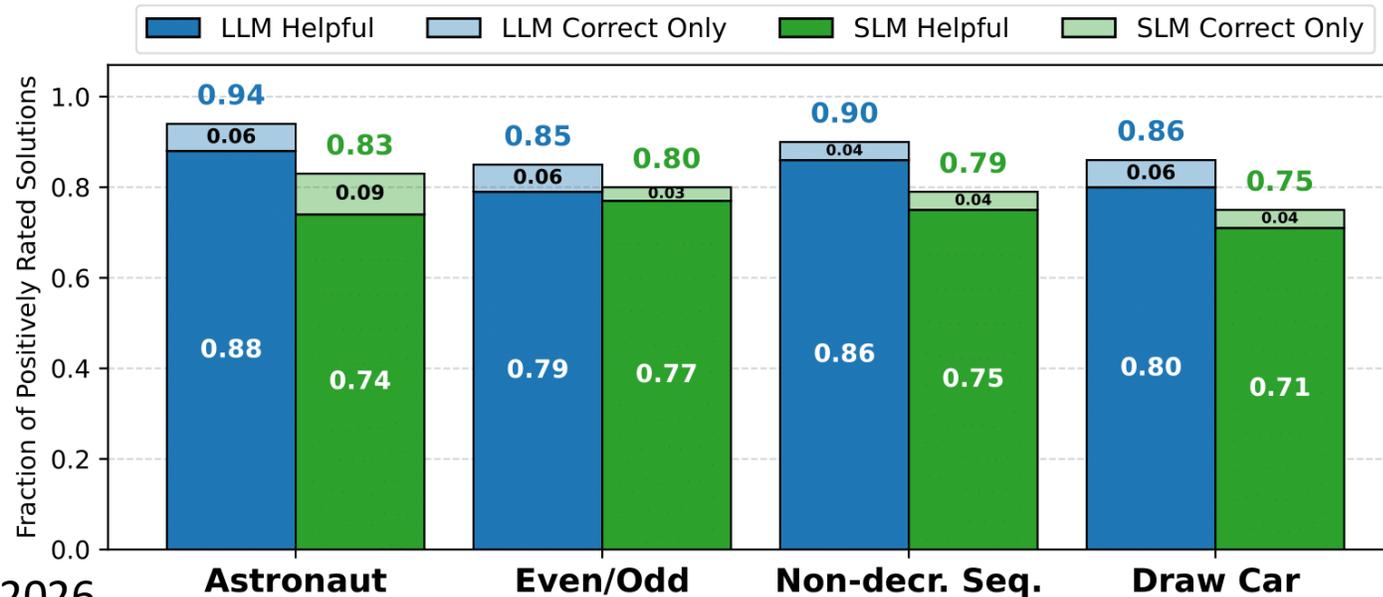
- Perfect
- Incorrectly gets character to insert
- Incorrectly assumes one digit
- Doesn't insert character at correct place



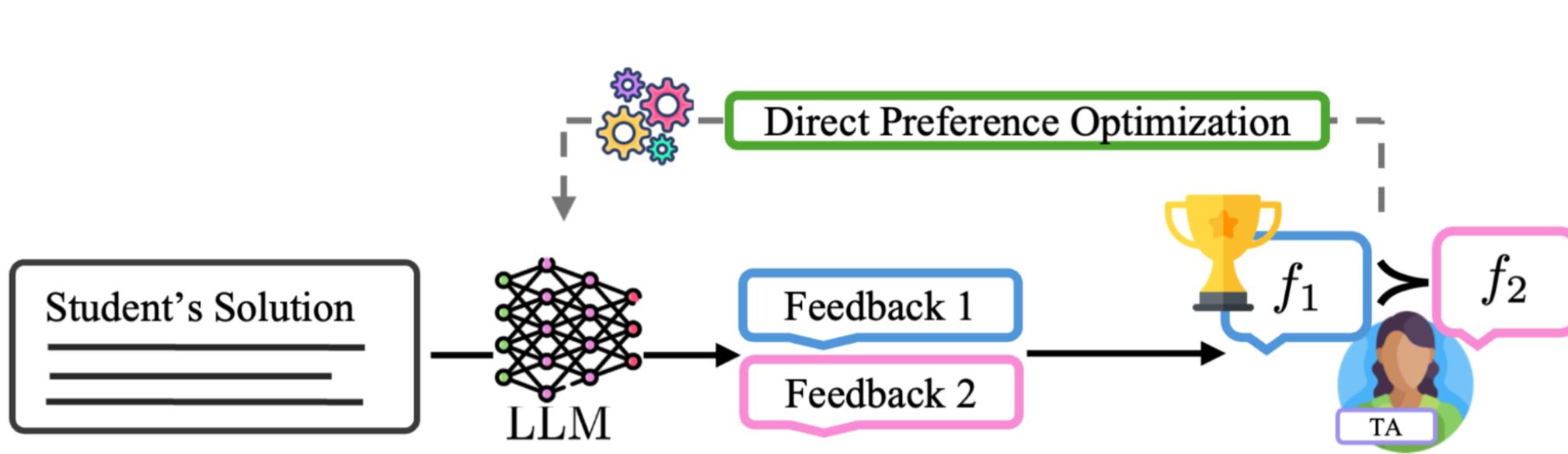
# AI Grading of Exams (after GPT)

We trained a 3B small open model to reproduce the outputs of a larger model.

Smaller models can run on local devices!! Don't need internet. And have privacy !

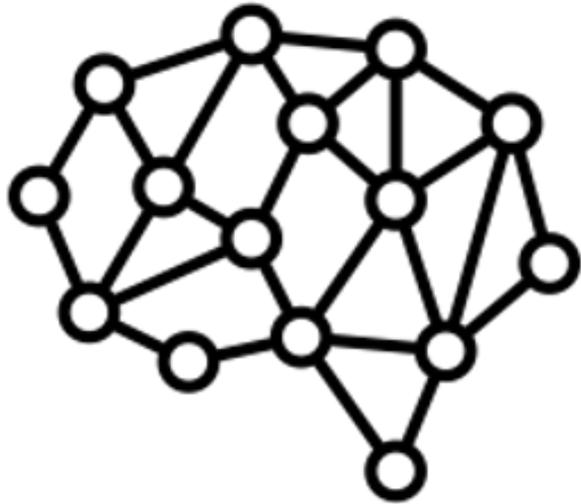


# AI Grading of Assignments (after GPT)



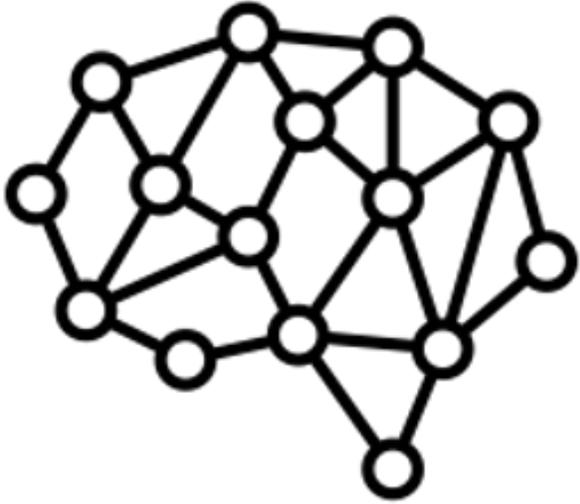
How can we use TA preferences to train an AI model to give high quality feedback that sounds like a TA from our course?

# A Tale of Two Ideas



AI powered  
interventions to  
support education  
at scale !!

# A Tale of Two Ideas

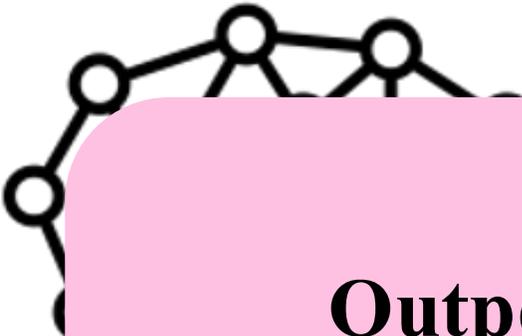


AI powered  
interventions to  
support education  
at scale !!



Solution to support  
education at scale  
***without AI***

# A Tale of Two Ideas



**Outperformed all AI  
approaches we tried by an order  
of magnitude !!**

AI  
interventions to  
support education  
at scale !!



Solution to support  
education at scale  
***without AI***

# A Tale of Two Ideas



**Outperformed all AI  
approaches we tried by an order  
of magnitude !!**

interventions to  
support education  
at scale !!



Solution to support  
education at scale  
***without AI***

Are you ready ??

But first...

# Challenges of 1:1 Help at Scale

## **Student to Teacher Ratio**

Somewhat solved with volunteer teachers.

## **Complex coordination**

Volunteer teachers are from 100+ countries across 24 time zones.

## **Teacher availability is sporadic**

Volunteer free time is often scattered in fragmented chunks of ~20 mins.

## **Avoid the negatives of help-seeking**

Students have to seek help; out of context help; long wait times can be demoralizing; can lead to learned helplessness.

# Key Insight

Give teachers the ability to initiate a 1:1 help session.

# Key Insight

If a teacher says they are free to help,  
we will find them a student!!

**Give teachers the ability to  
initiate a 1:1 help session.**

Don't wait for a student to ask for help.  
That means we would have to have a  
teacher who is free at exactly that time  
!!

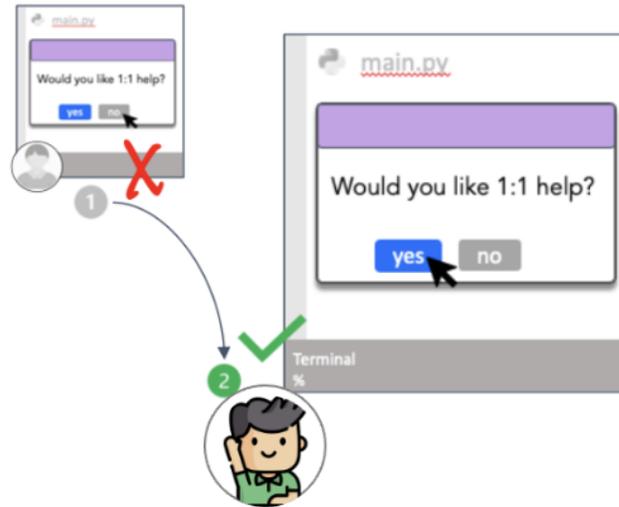
We first tried this in CS109 !



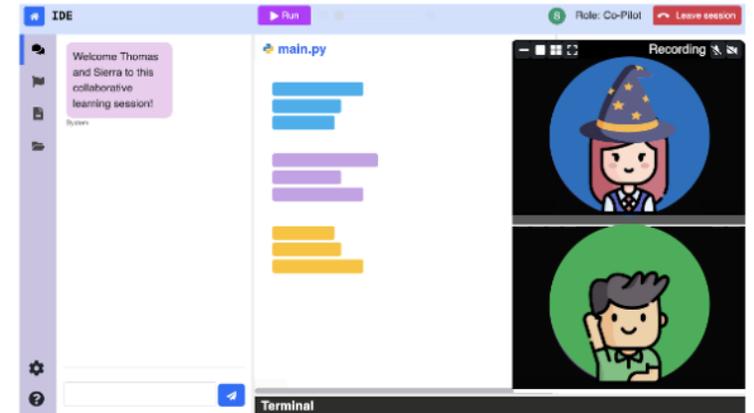
# TeachNow System



A volunteer teacher has 20 mins of spare time and initiates a TeachNow session

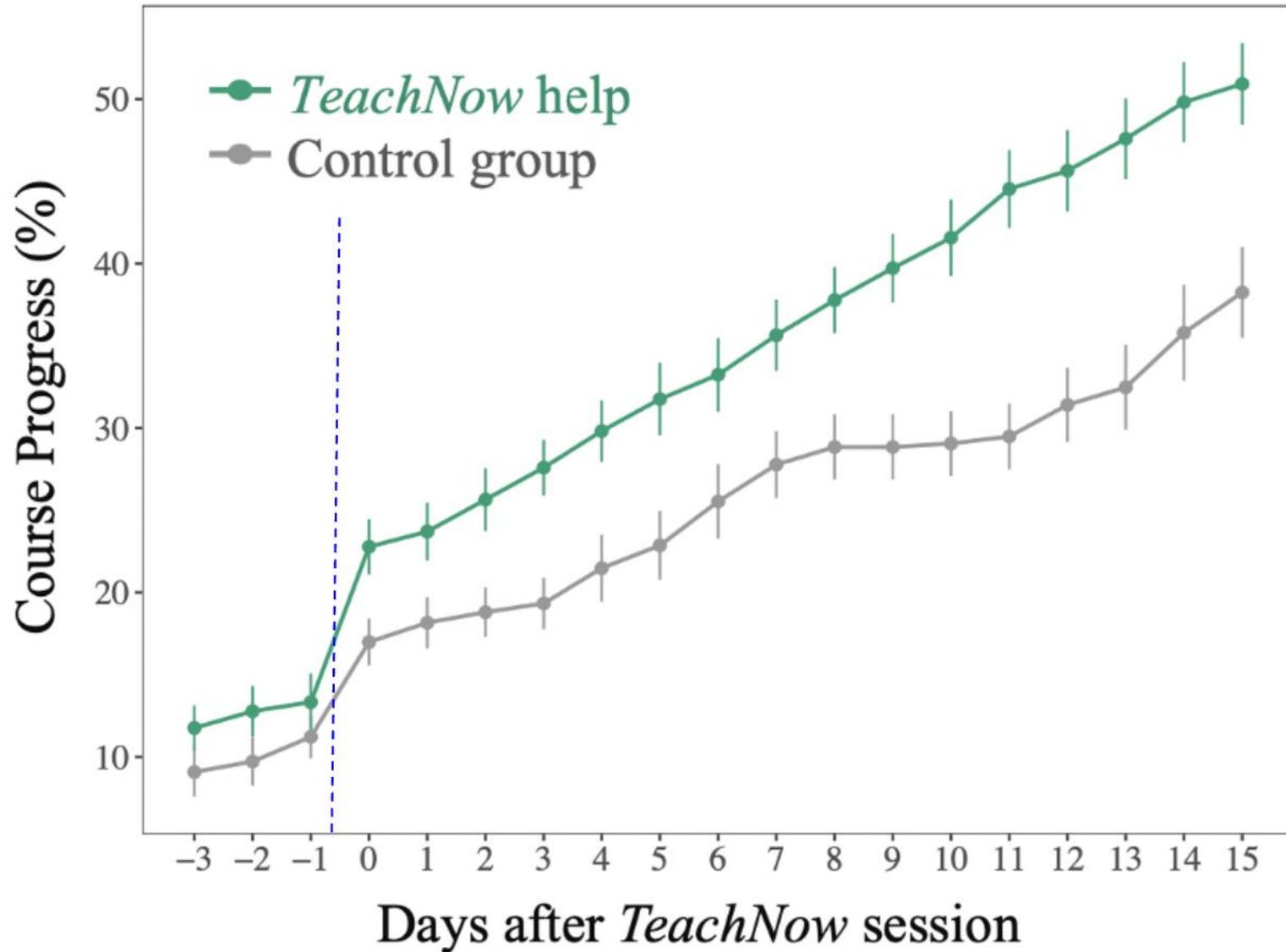


Our system finds a student who is working on an assignment and offers them 1:1 help

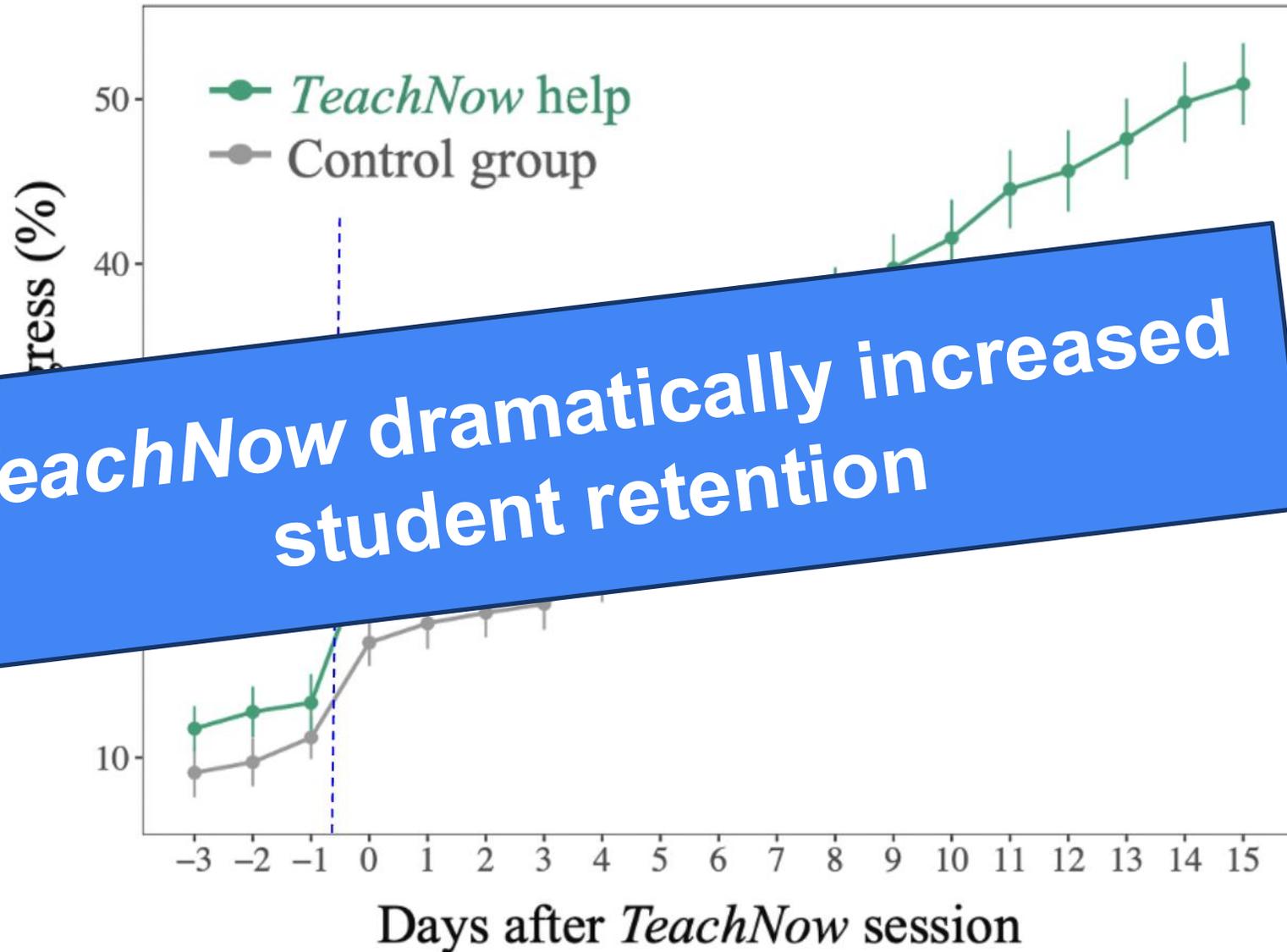


The student and teacher are placed into a virtual room with shared IDE, video, and chat.

# Impact of *TeachNow* on student outcomes



# Impact of *TeachNow* on student outcomes



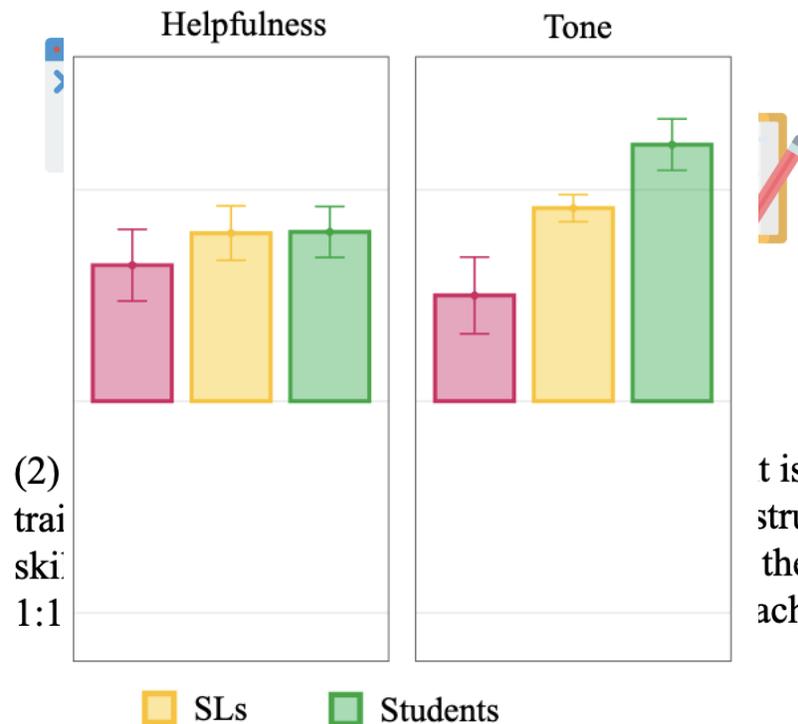
***TeachNow* dramatically increased student retention**

# Who Else Might Want to Teach?

# Who Else Might Want to Teach?

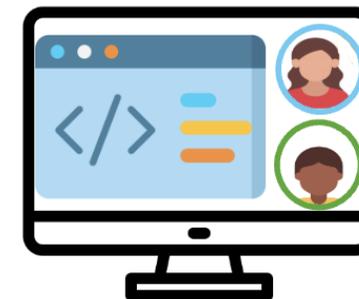


(1) Students in a massive online course have the opportunity to teach other students.



(2) Training 1:1

It is instructors they are teaching.



(4) Once a student is accepted, she can start hosting 1:1 sessions to help other students.

We built a pipeline for students to teach other students in Code in Place !!!

# Who Else Might Want to Teach?



(1) Students in a massive online course have the opportunity to teach other students.



(2) Students go through a training and practice key skills to prepare them for 1:1 teaching.



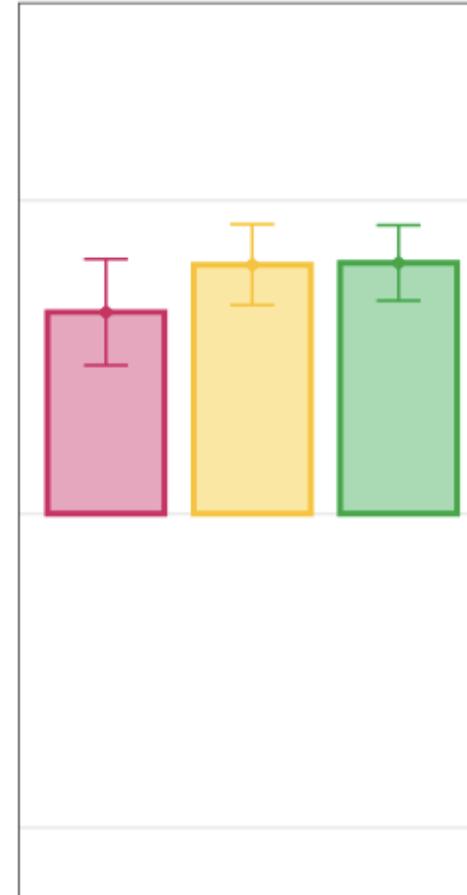
(3) Each student is evaluated by instructors and AI to see if they are ready for 1:1 teaching.



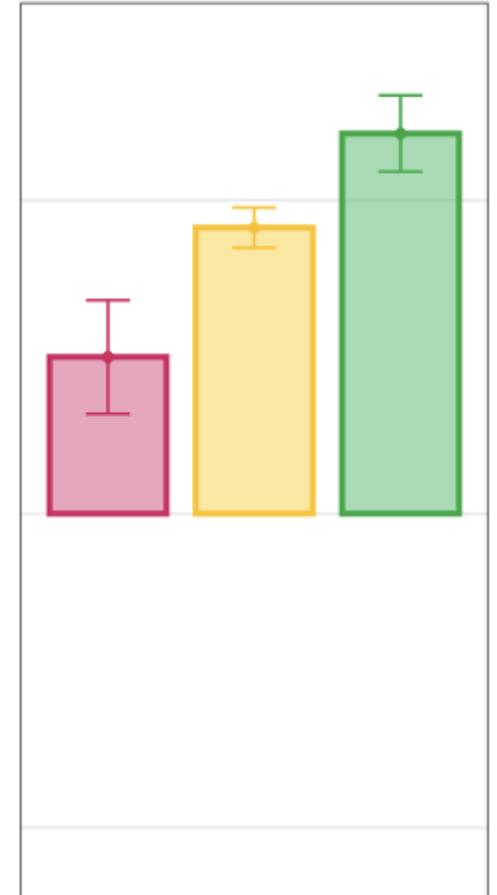
(4) Once a student is evaluated by instructors and AI, she can start hosting 1:1 sessions to help other students.

Experts rated the students to be almost as good as the Section Leaders at helping!

### Helpfulness



### Tone



■ SLs

■ Students

Lots of people want to  
teach. Do you??

Back to Probability

People just miss the random variables

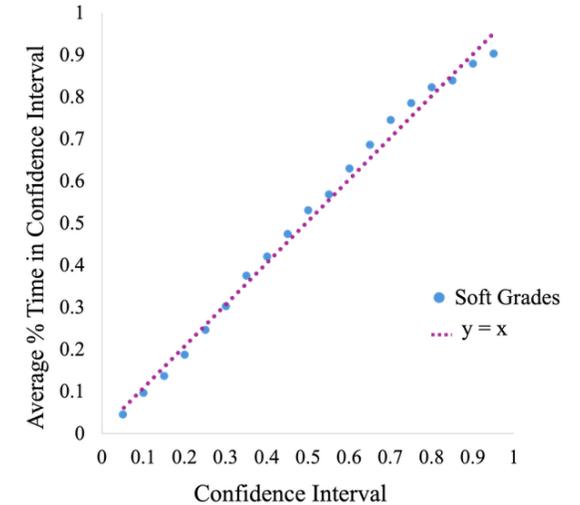
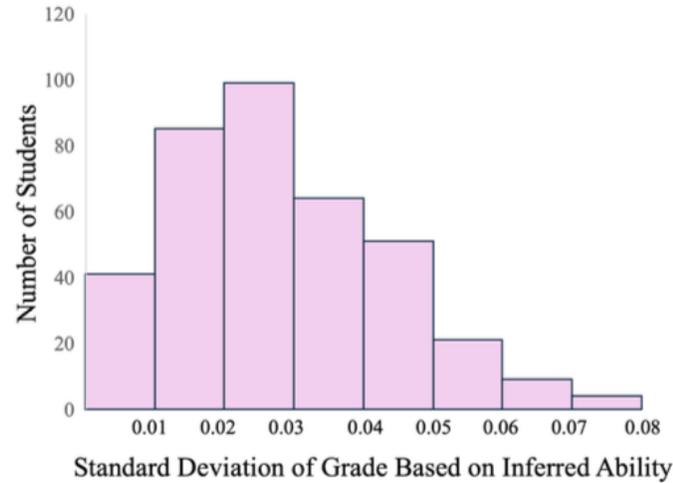
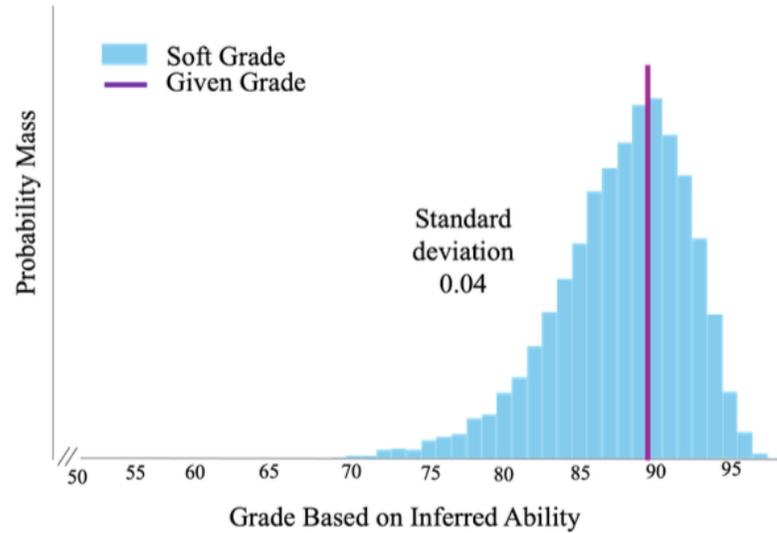
What else should be a **random variable**?

Chris (my PhD advisor) in  
2023: Grades??

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

Juliette 2024: Grades!

# Soft Grades. Accepted for publication in 2025



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

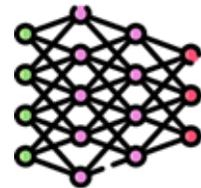
	C1	C2	OULAD-1	OULAD-2	OULAD-3	OULAD-4	OULAD-5	OULAD-6	OULAD-7
<b>Soft Grades (RMSE)</b>	<b>0.042</b>	<b>0.058</b>	<b>0.069</b>	<b>0.067</b>	<b>0.062</b>	<b>0.089</b>	<b>0.071</b>	<b>0.046</b>	<b>0.076</b>
CRM Baseline (RMSE)	0.060	0.264	0.079	0.221	0.071	0.145	0.080	0.401	0.349

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

Can we have a model for this but not  
just for grades? For understanding  
students....

# Learner Insight Tracing

- Generate short, human readable **insights** based on student work.
- Describe what the student appears to understand.
- Everything is represented as text!

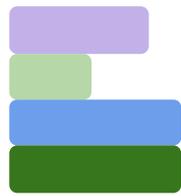


LLM

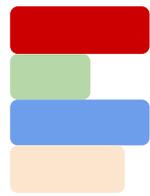


“The student may be unclear on the **difference between printing a string versus the value of an expression....** incorrectly uses quotation marks, **thereby printing the literal string ‘x’ ...**”

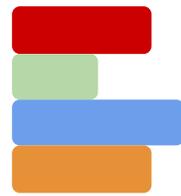
# Learner Insight Tracing: How it Works



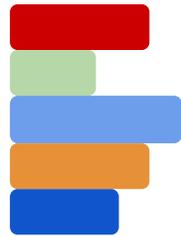
Attempt 1



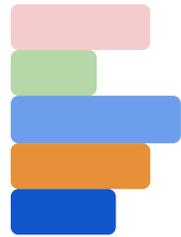
Attempt 2



Attempt 3



Attempt 4



Attempt 5



Attempt 6



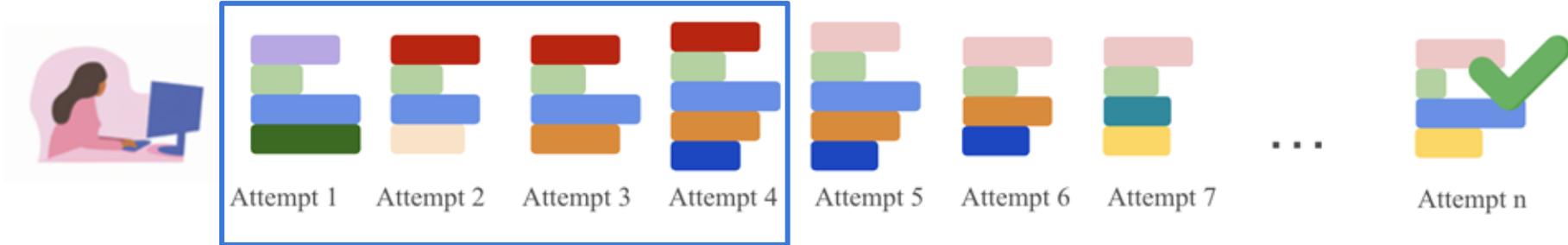
Attempt 7

...



Attempt n

# Learner Insight Tracing: How it Works



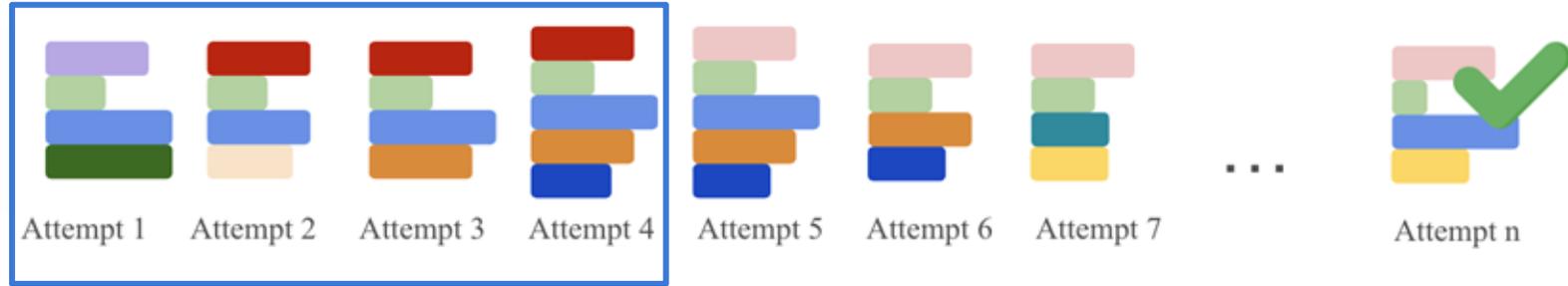
```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    print("Your answer:", z)
```

# Learner Insight Tracing: How it Works



```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
    print("What is", "x",  
          "+", "y", "?")
```

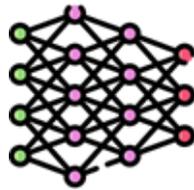
```
    print("What is", "x",  
          "+", "y", "?")
```

```
    print("What is", "x",  
          "+", "y", "?")
```

```
    z = x + y
```

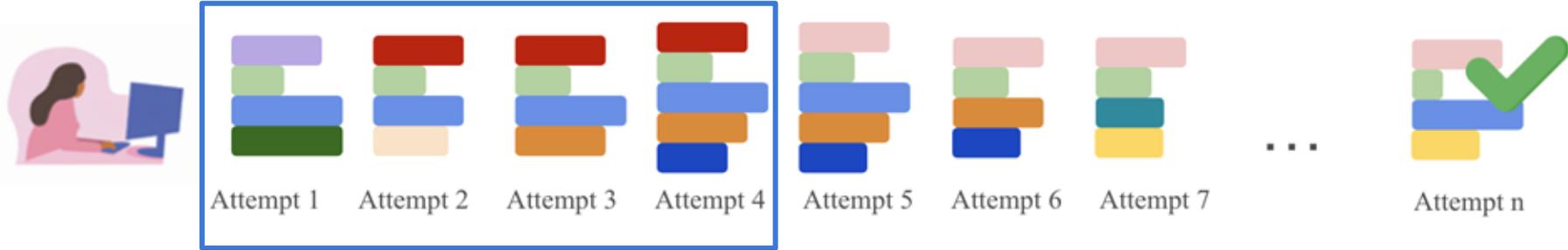
```
    z = x + y
```

```
    print("Your answer:", z)
```



LLM

# Learner Insight Tracing: How it Works



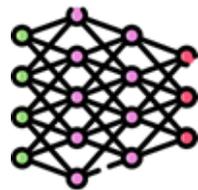
```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    print("Your answer:", z)
```

## Insight 1



LLM



“The student may be unclear on the difference between printing a string versus the value of an expression.... incorrectly uses quotation marks, thereby printing the literal string ‘x’...”

# Learner Insight Tracing: How it Works



Attempt 1



Attempt 2



Attempt 3



Attempt 4



Attempt 5



Attempt 6



Attempt 7



...

Attempt n



# Learner Insight Tracing: How it Works



Attempt 1



Attempt 2



Attempt 3



Attempt 4



Attempt 5



Attempt 6



Attempt 7

...



Attempt n

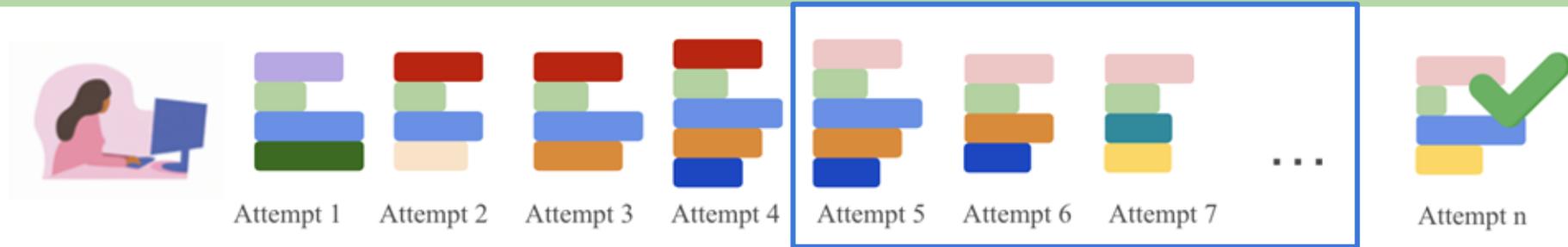
```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    print("Your answer:", z)
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    input("Your answer:", z)
```

# Learner Insight Tracing: How it Works



```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")
```

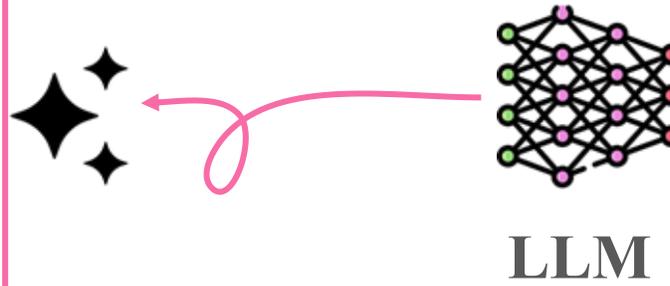
```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y
```

```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    print("Your answer:", z)
```

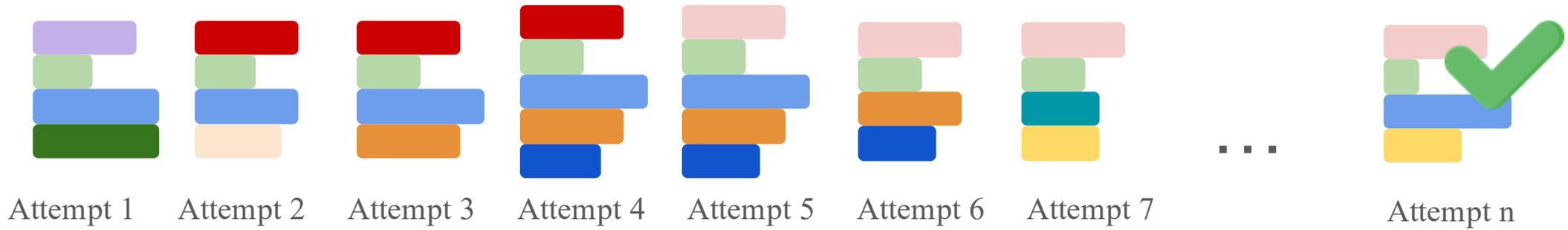
```
def main():  
    print("Khansole Academy")  
    x = int(input("x is"))  
    y = int(input("y is"))  
  
    print("What is", "x",  
          "+", "y", "?")  
  
    z = x + y  
    input("Your answer:", z)
```

## Insight 2

“They appear to **not always understand variables**. They **do not store the user's answer** to the addition problem in a variable or check if it is correct....”



# Learner Insight Tracing: How it Works



**Insight 1**



**Insight 2**



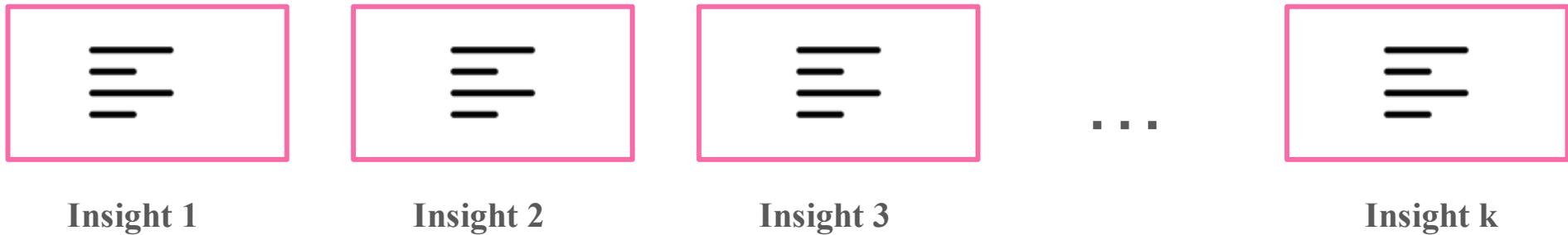
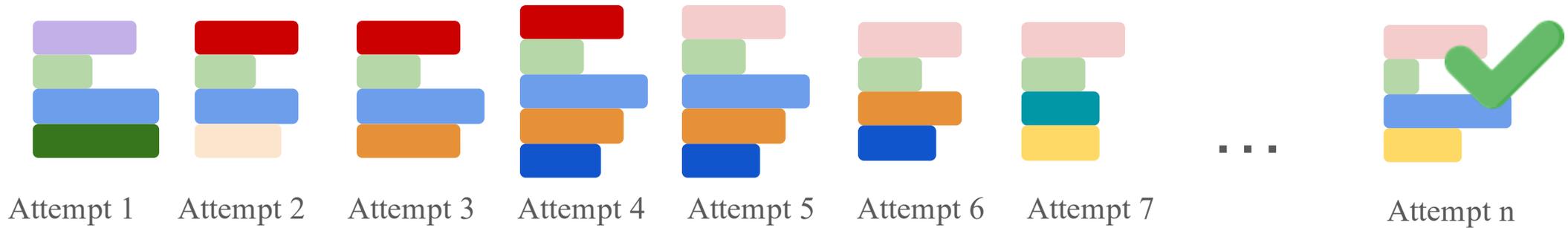
**Insight 3**

...



**Insight k**

# Learner Insight Tracing: How it Works

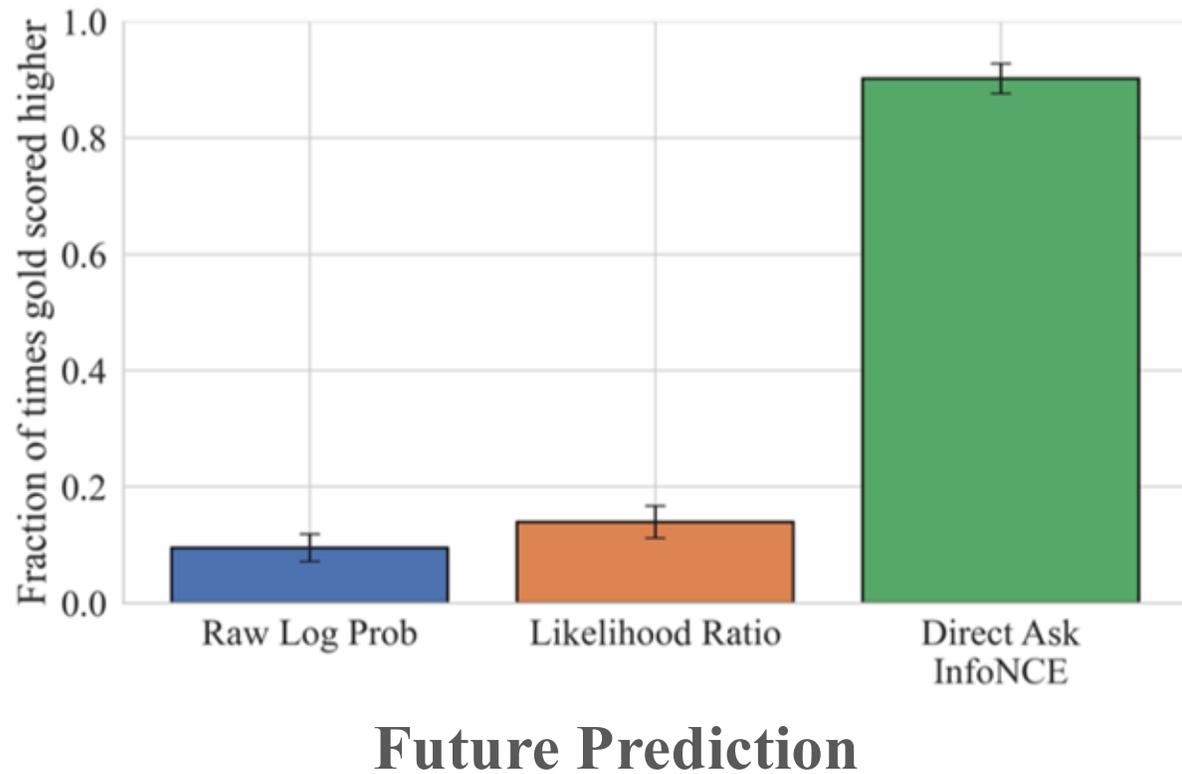
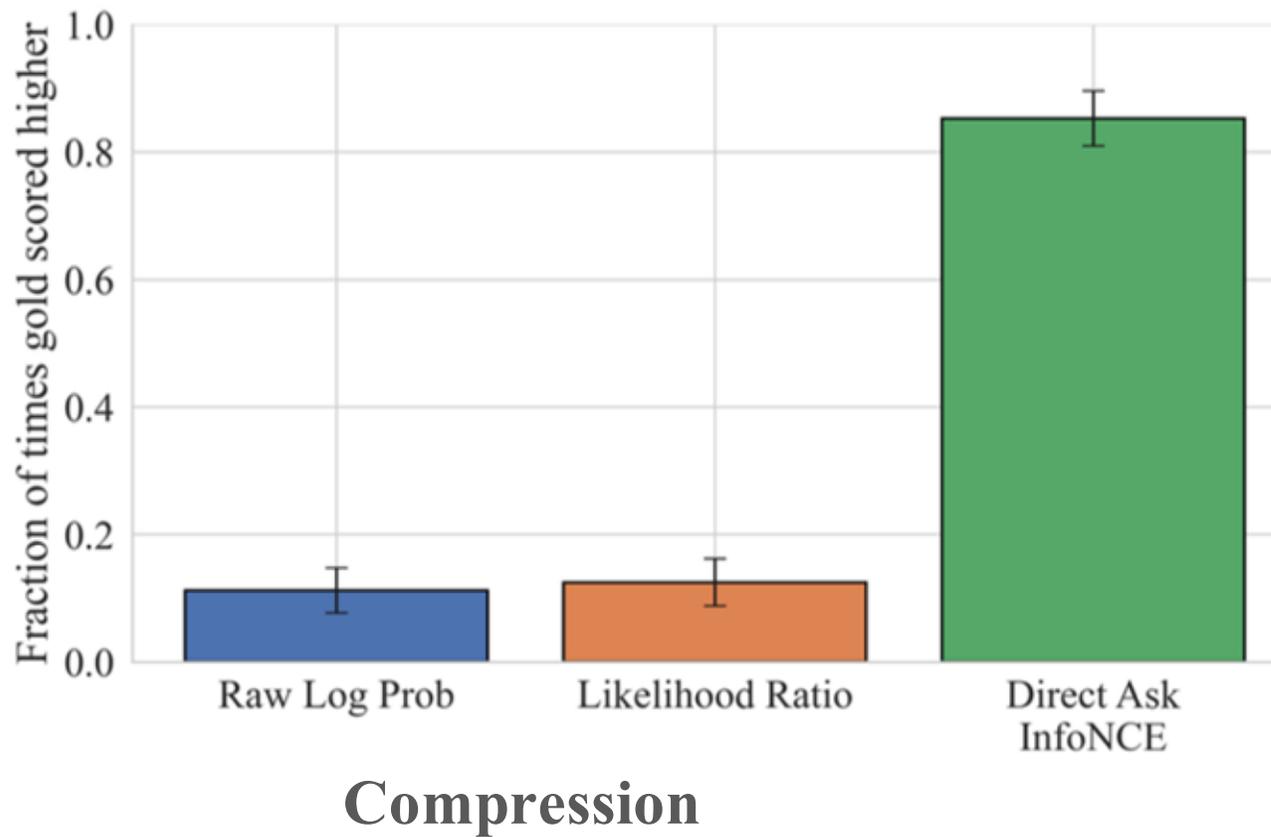


Sequence of interpretable representations of her understanding.

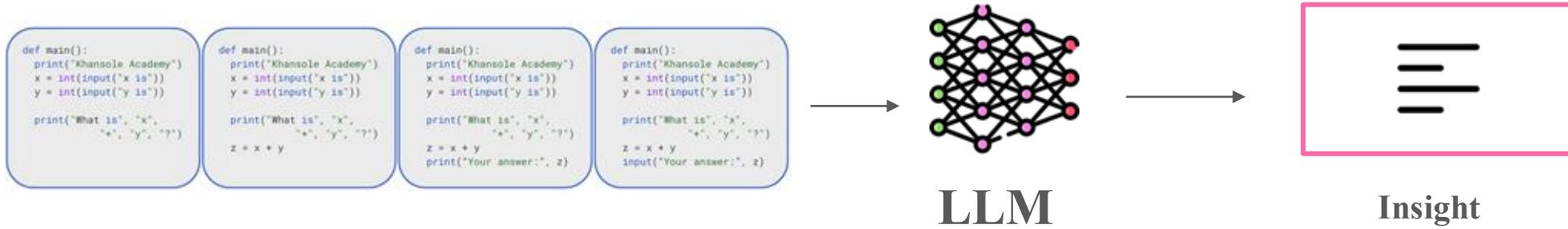
How do you know if you have a good  
insight???

Information Theory + LLM !!

# Direct ask Info NCE



# LIT Big Picture



## Insight generation

Option A



Option B



## Principled scoring function

Use LLMs to generate summaries of student understanding as they are working on assignments.

Show them to teachers?? Use them to provide hints?? So many good options !!

# CS109 Projects

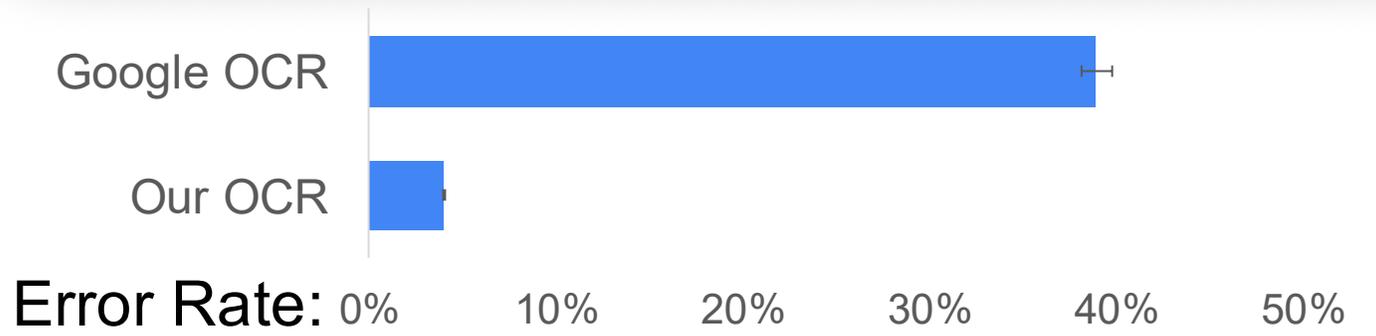
```
def example():  
    → print("two space")  
    → → print('four space')  
    → print(two space, no quotation)
```

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

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

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

```
pythonProject Version control main  
Project  
pythonProject ~/Pycharm  
venv  
main.py  
External Libraries  
Scratches and Consoles  
main.py  
def example():  
    print("two space")  
    print('four space')  
    print(two space, no quotation)  
def main():  
    print(nothing)  
    example_function()  
    print("finishing")  
pythonProject > main.py 9:23 LF UTF-8 4 spaces Python 3.10 (pythonProject)
```



# Information Theoretic Learning!

youtube.com/watch?v=z2n4e\_oS5jU

Acrobat File Edit View E-Sign Window Help

Infolding\_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 (%)								
Random	10.30 ± 0.96	16.03 ± 1.06	15.60 ± 1.15	25.03 ± 1.23	28.40 ± 1.43	41.38 ± 1.40	49.40 ± 1.58	63.95 ± 1.36	69.00 ± 1.46	78.43 ± 1.19
Frequent	11.80 ± 1.02	21.25 ± 1.14	22.60 ± 1.32	34.76 ± 1.35	42.70 ± 1.56	53.30 ± 1.46	60.10 ± 1.55	69.18 ± 1.36		
Divergence	15.60 ± 1.15	25.03 ± 1.23	28.40 ± 1.43	41.38 ± 1.40	49.40 ± 1.58	63.95 ± 1.36	69.00 ± 1.46	78.43 ± 1.19		
Entropy	16.20 ± 1.17	25.85 ± 1.24	28.80 ± 1.43	41.80 ± 1.41	49.70 ± 1.58	63.09 ± 1.38	69.10 ± 1.46	77.72 ± 1.21		

Figure 1. Accuracies Using Four Vocabulary Picking Methods.

still produce different vocabularies, as demonstrated in the cited study (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 learners at different learning stages. One approach is to leverage the A1-C2 language proficiency scale, which 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 uncertainty a learner already knows about that word. Then use inference to update our belief about the word 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 the word.

Infolding - An Information-Theoretic Approach to Language Learning

Unlisted

Alice Heiman  
1,01K subscribers

Subscribe

1 0 Share Download Clip ...

All For you Recently uploaded Watched

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

EULER  
Newthink  
406K views · 2 months ago

LETHAL GENE DISEASES  
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

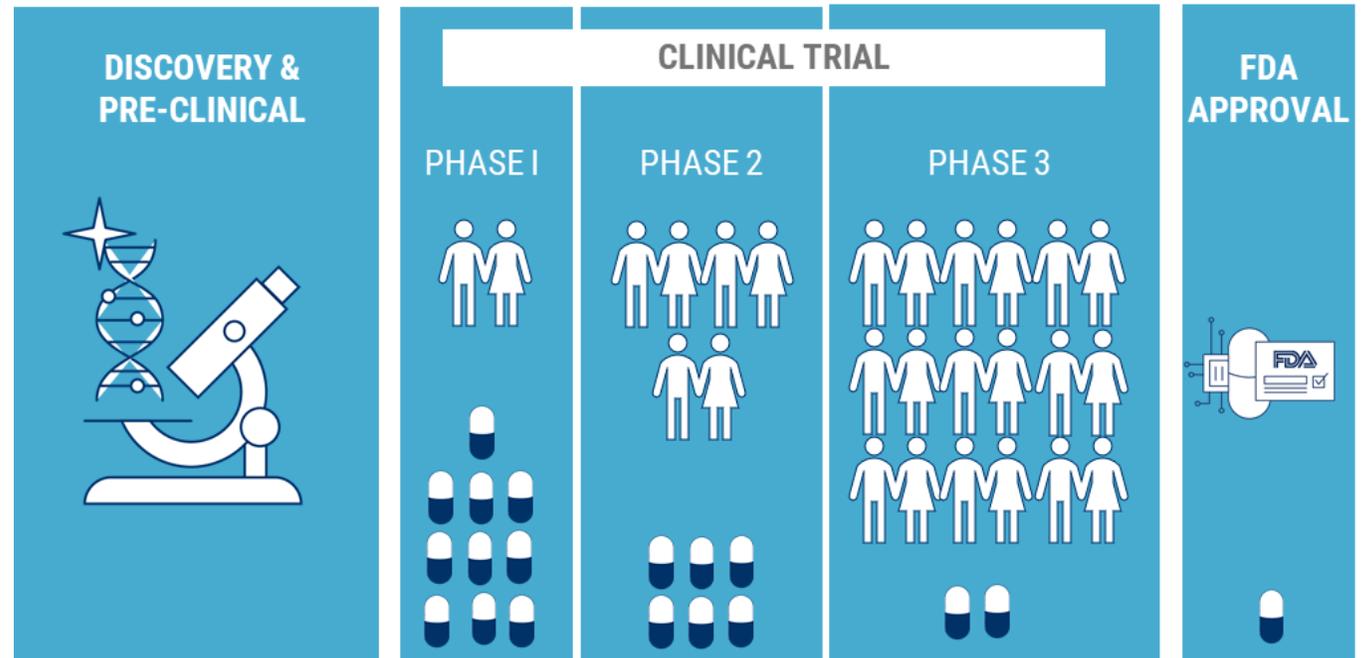
ATTEMPT NO LANDING HERE  
We Might Find Alien Life In 2200 Days  
Veritasium  
9.6M views · 5 months ago

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

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
```

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.

Navigation buttons for "Previous Question" and "Next Question" are visible at the bottom.

On the right side, there is an "Answer Editor" and "Solution" section. The "Numeric Answer" is 4. The "Explanation" contains the text "asdfasd!!!!?? Y = 5". Below this, there are two code editors. The first contains `1 print('test')` with "Run" and "Show" buttons. The second contains `1 for i in range(100):` and `2 print(i)` with "Run" and "Show" buttons. The output of the second code editor is shown as "abcdef" and "Y = 25".

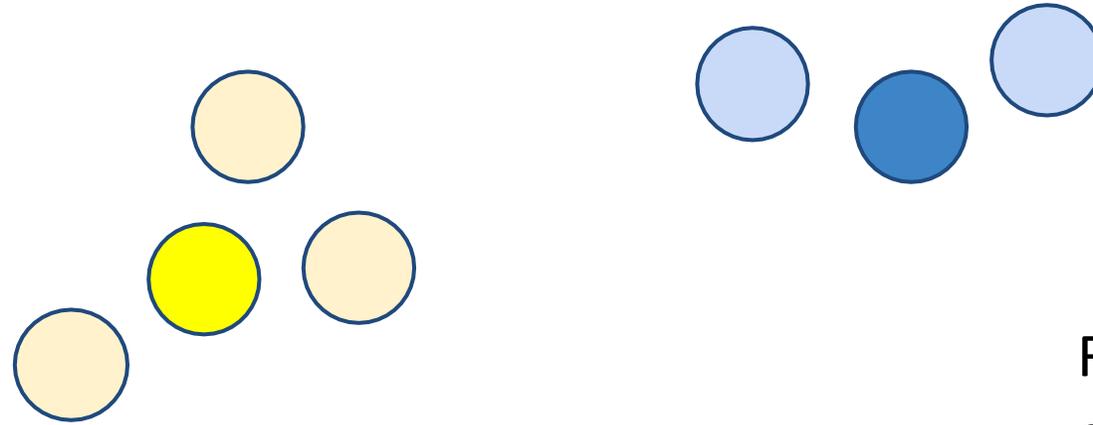
300 solutions

Find the 10 solutions which are most representative



# K Medoids: A Classic Algorithm

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



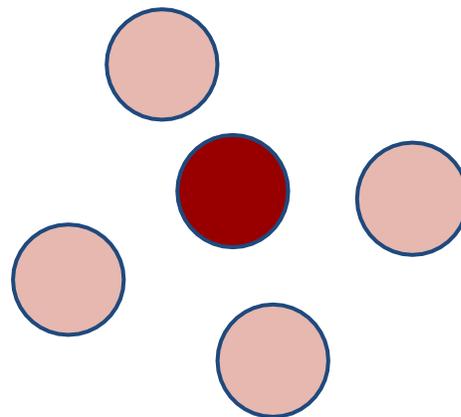
For any two points you can query the distance between them. May be non-Euclidean

Before

$$O(n^2)$$

After

$$O(n \log n)$$

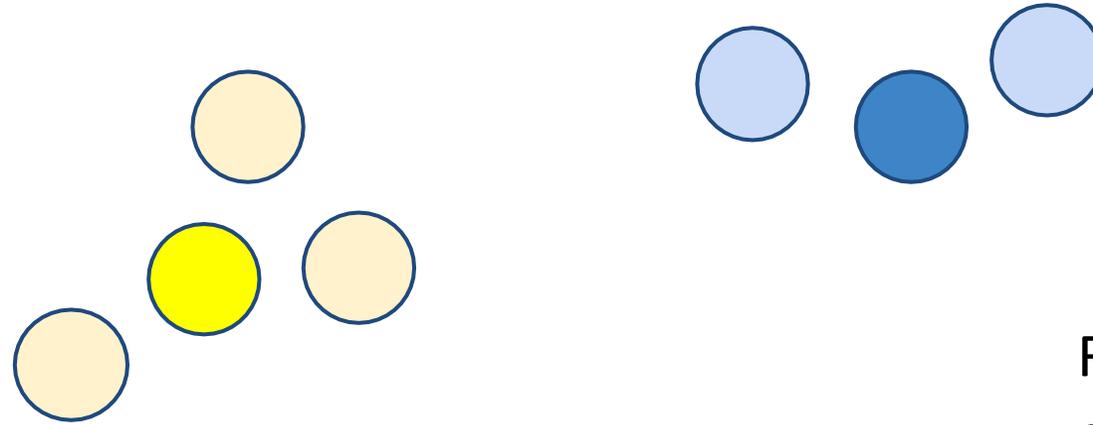


CS109 TA

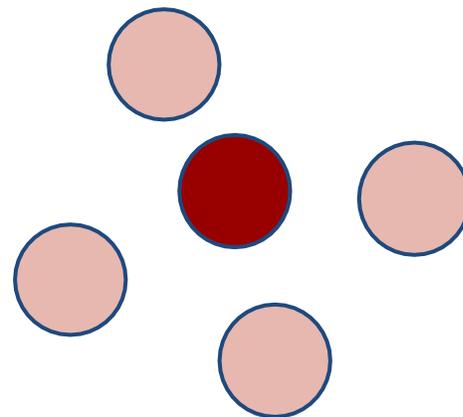


# Step 1: Chose the node closest to the rest

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

$$O(n^2)$$

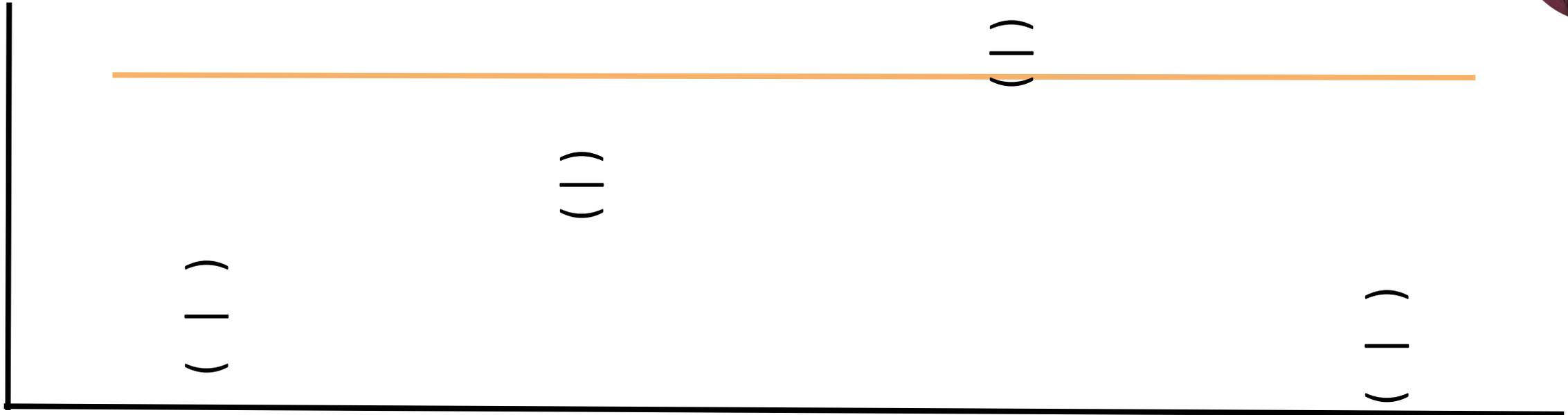
After

$$O(n \log n)$$

CS109 TA



# 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



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

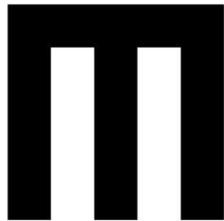
Chris in 2017: Ability to See??

# More than education

Vision Test

myeyes.ai/measure

## Left Eye



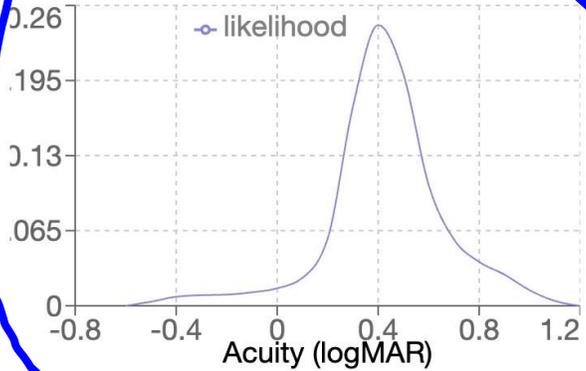
Featured in  
**THE LANCET**

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

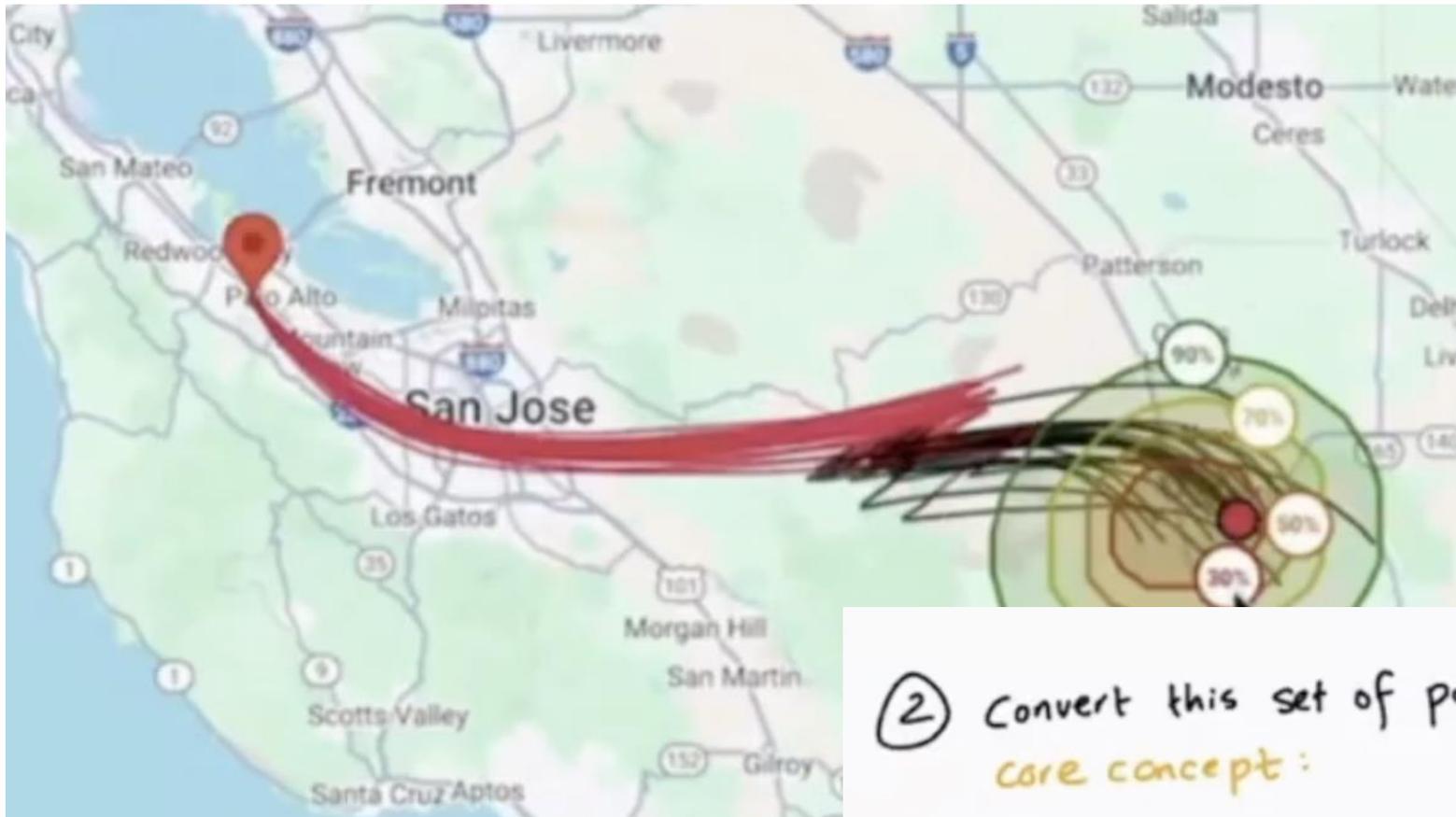
Likelihood of Acuity Scores:



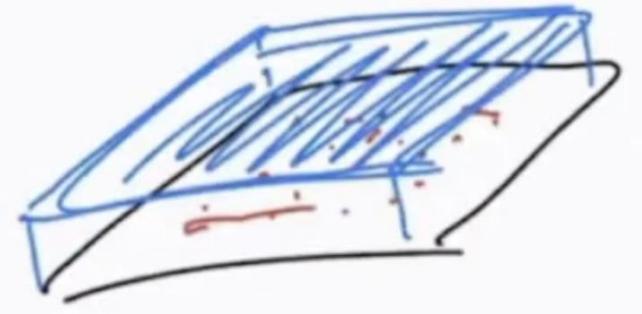
Acuity (logMAR)	Likelihood
-0.8	0.00
-0.4	0.00
0.0	0.01
0.4	0.26
0.8	0.05
1.2	0.00



# Balloon Landing as Random Variable



② Convert this set of points into a probability distribution  
core concept:



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](#)



ed CS 109 Winter 2026 – Ed Discussion

New Thread

COURSES +  
CS 109 Winter 2026 7

CATEGORIES  
General  
Lectures  
Sections  
Problem Sets  
Exams  
PEP  
Social

Search

Filter

Beyond CS109 #375

Juliette Woodrow STAFF  
4 hours ago in General  
UNPIN STAR WATCHING VIEWS 16

I'll talk more about this in lecture today, but in the meantime I thought I would also share a discussion among CS109 TAs about what classes to consider after takings CS109.

### Answer 1

Ooh I love this question! We (the teaching team) were just talking about this!

A couple from me:

**CS238/AA228: Decision-Making Under Uncertainty.**

Bayes Nets are a big player here, and you'll get an intro to things like reinforcement learning (with applications into things like autonomous aviation/navigation, for example, which is why it's cross-listed with AA). Mykel's a great lecturer, the content is interesting, and the pace is approachable and chill (quizzes and a final project as opposed to exams).

**CS265: Randomized Algorithms with Probabilistic Analysis.**

A *HUGE* warning label on this rec: I would probably not recommend this to people who don't have a very strong math background (well beyond the normal SoE reqs) unless you're willing to push yourself and take an academic risk. The material is really, interesting, but the prereqs are misleading. (I thought that — because I'm into Theory and crushed 109, 161, and the SoE math reqs — I'd be prepared. I wasn't.)

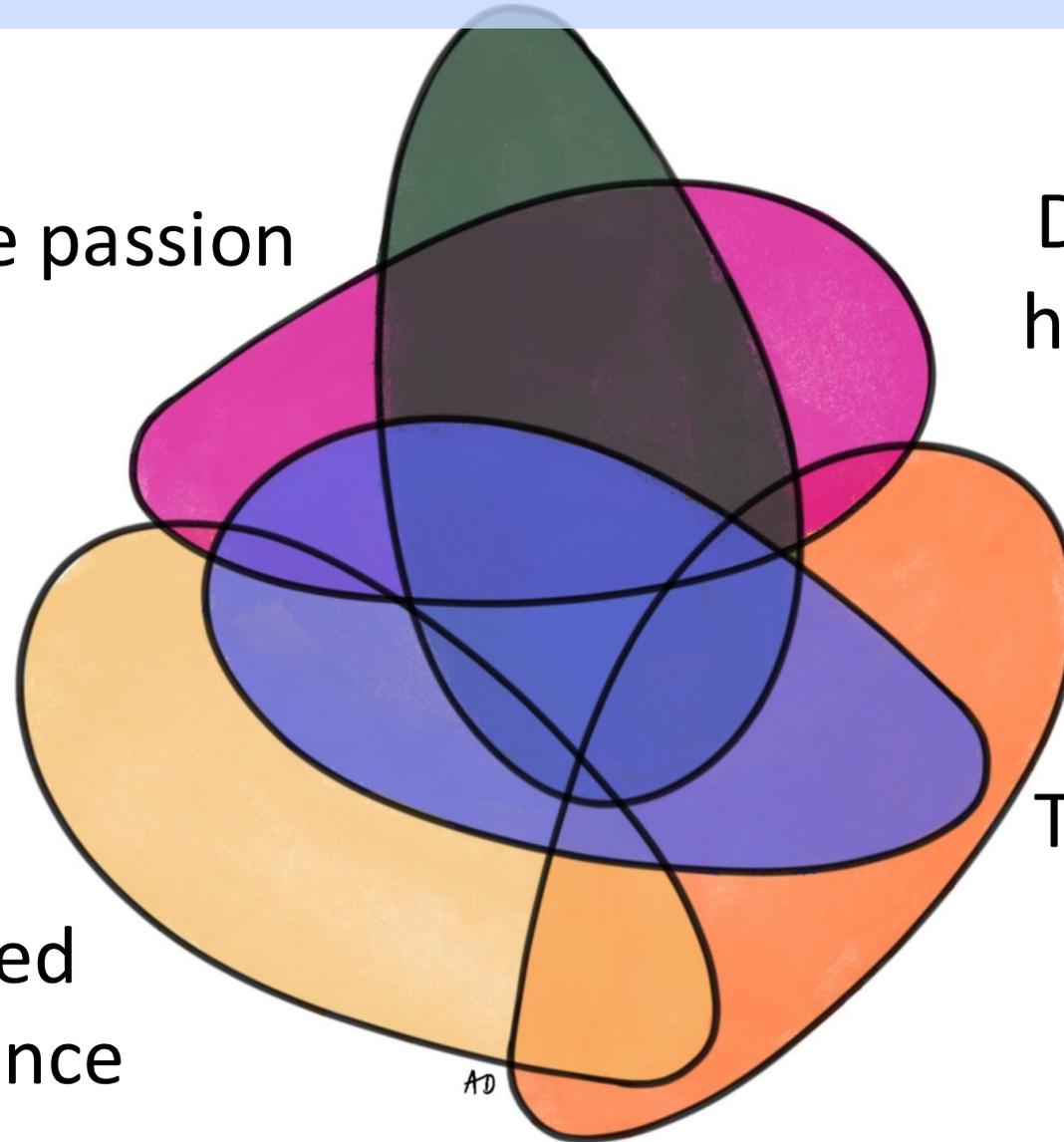
That said, I survived — and if I'd waited another year or two to take this class, I think I would've gotten a lot more out of it!

Open <https://edstem.org/us/courses/90458/discussion/7799469> in a new tab and focus it

# Think about intersectionality

Your side passion

Data that you  
have access to



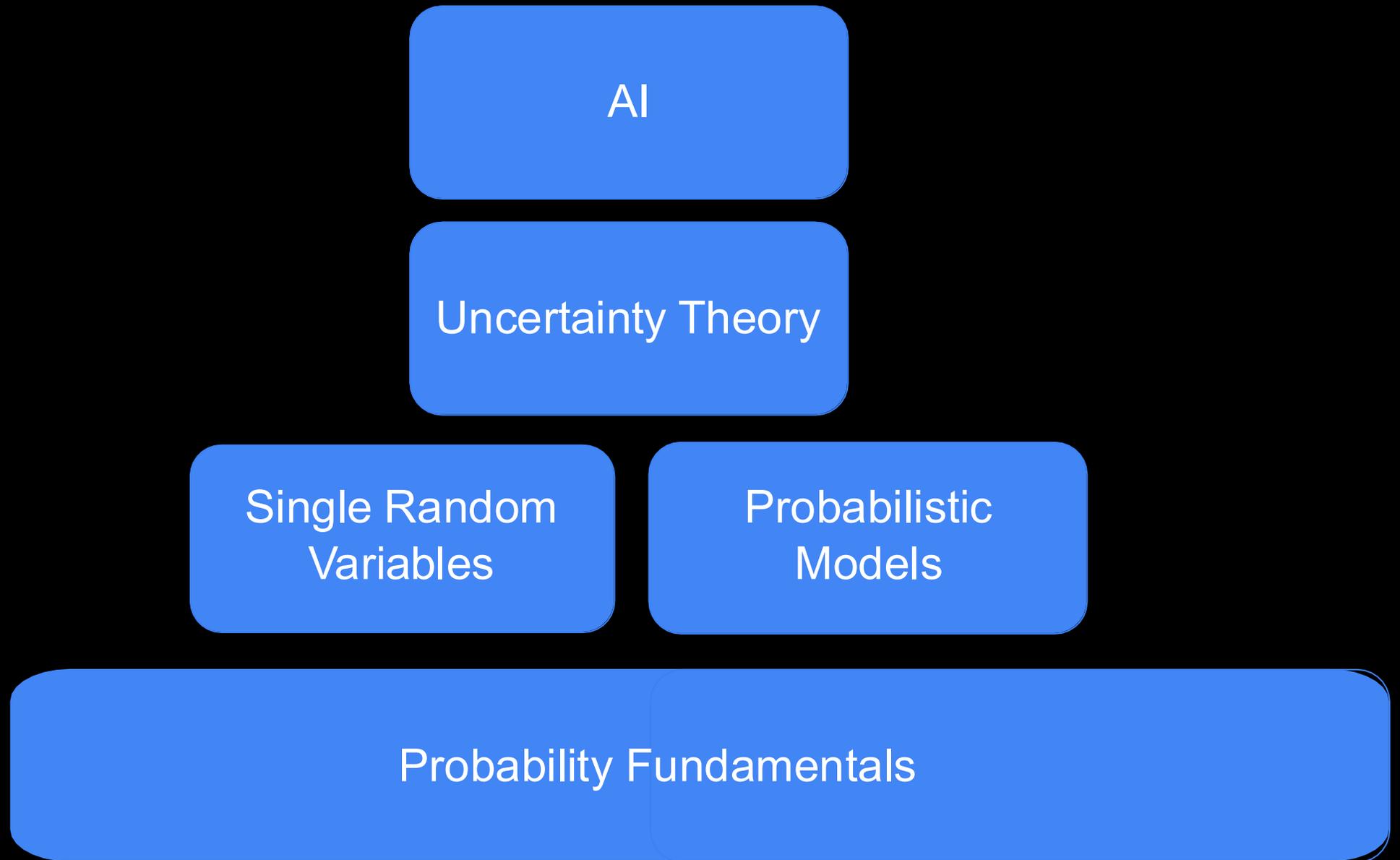
Your lived  
experience

Thompson  
sampling

AD



Last Class...

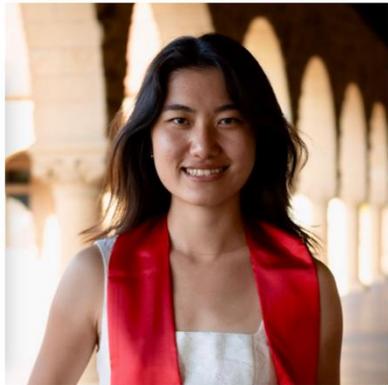




## COURSE VALUES

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 "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 are "Previous Question" and "Next Question" buttons.

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

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

```
Running 100000 trials
0.38563
```

# Netflix and Learn

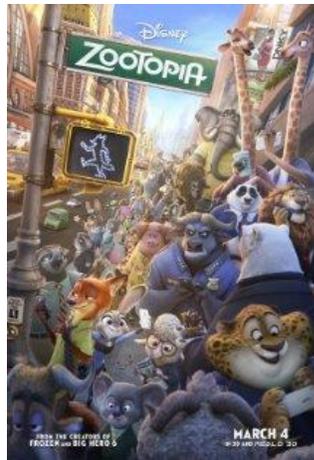
$$P(E|F) = \frac{P(EF)}{P(F)}$$
 Definition of Cond. Probability

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



$$P(E) = 0.19$$

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



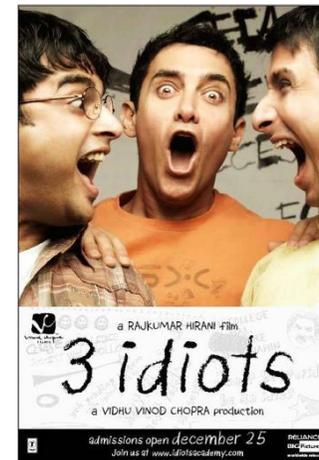
$$P(E) = 0.32$$

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



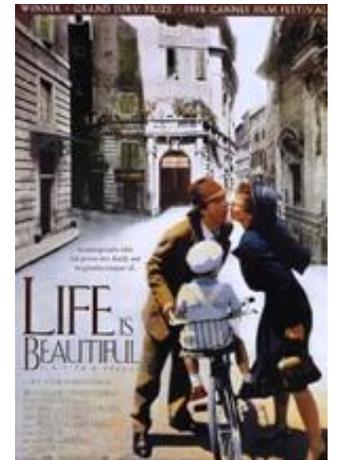
$$P(E) = 0.20$$

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



$$P(E) = 0.09$$

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



$$P(E) = 0.20$$

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

# Montey Hall Problem



Marilyn discovers the  
Probability Bug



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



**WHEN YOU MEET YOUR BEST FRIEND**

Somewhere you didn't expect to.

# Zika Test



Positive Zika.

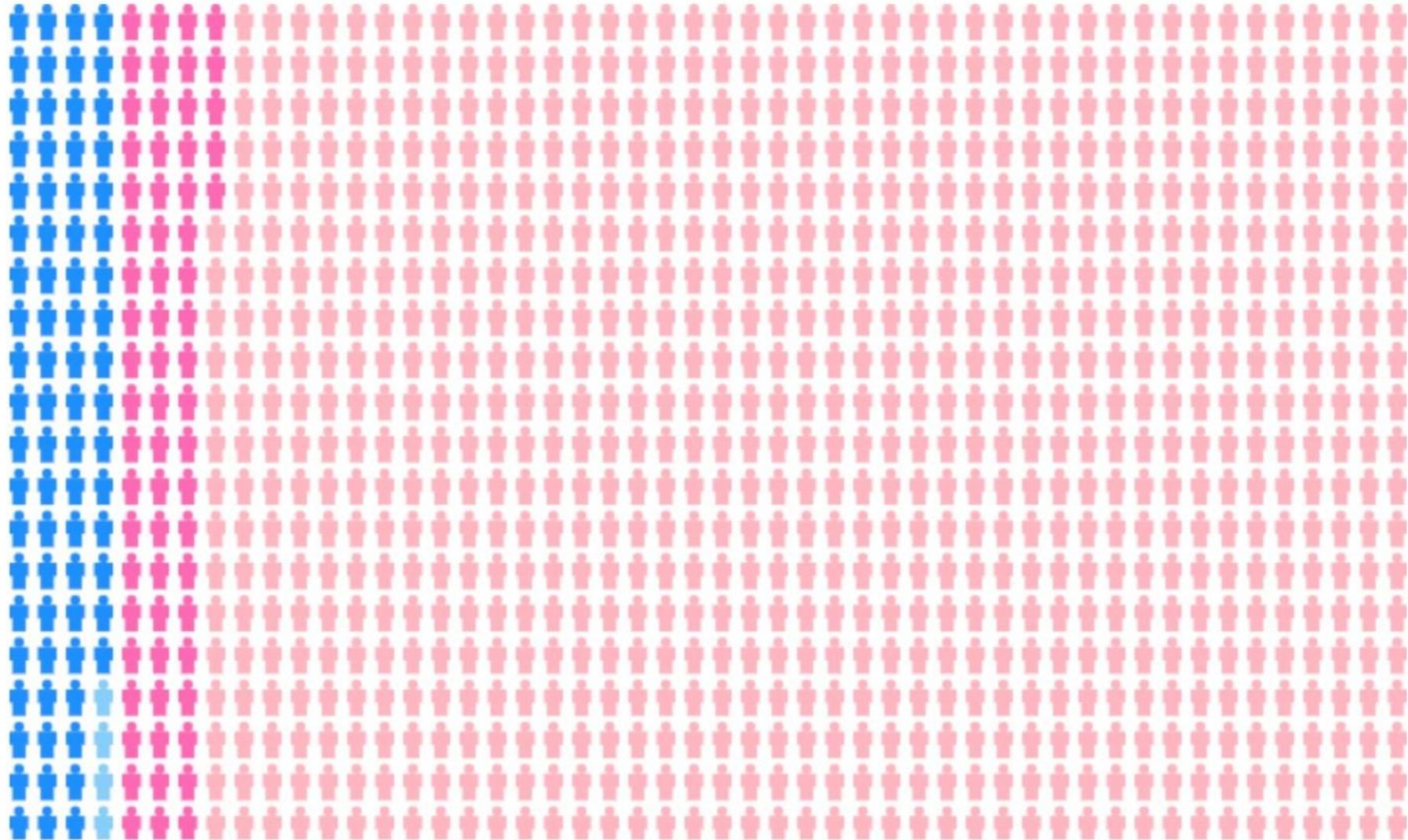
*What is the probability of zika?*

- 
- *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

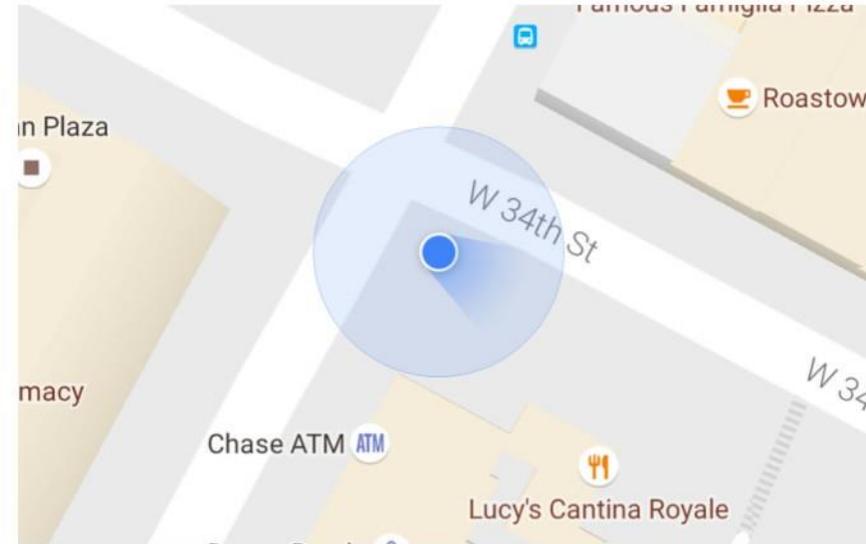
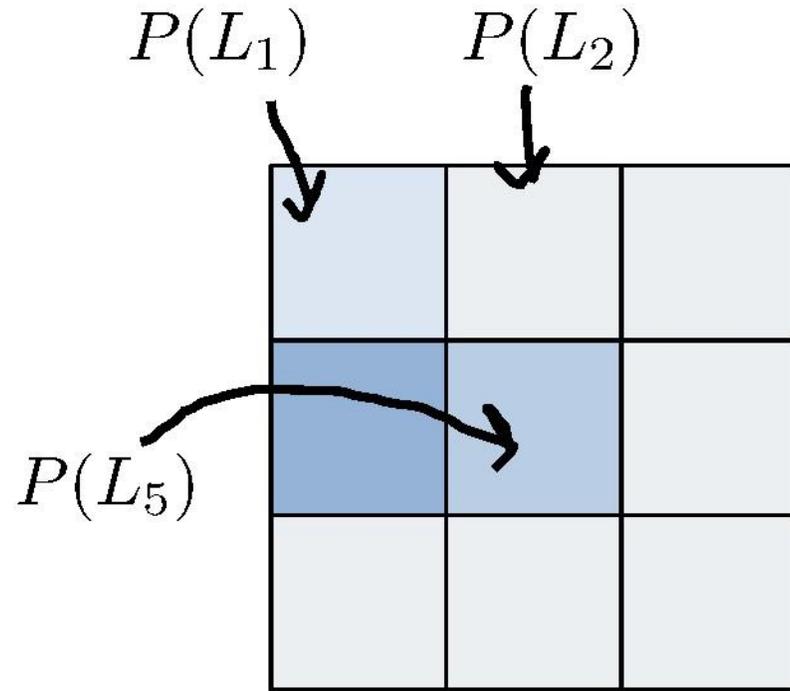
Answer Editor      Solution

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



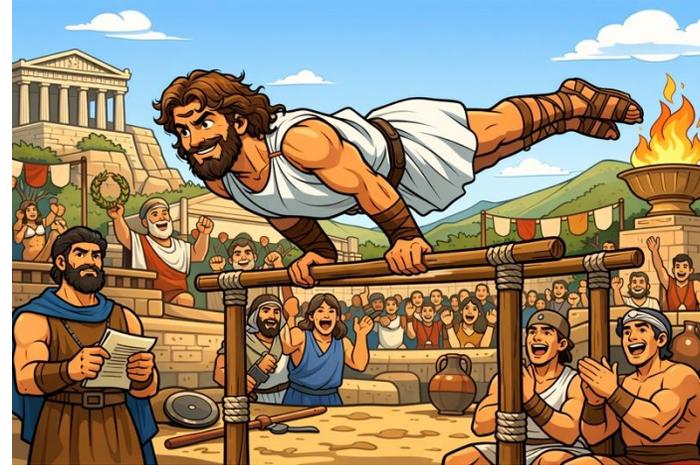
Before Observation



# Recall our Ebola Bats



# Sixth Year of Sections



#4 #3 #3



# 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.

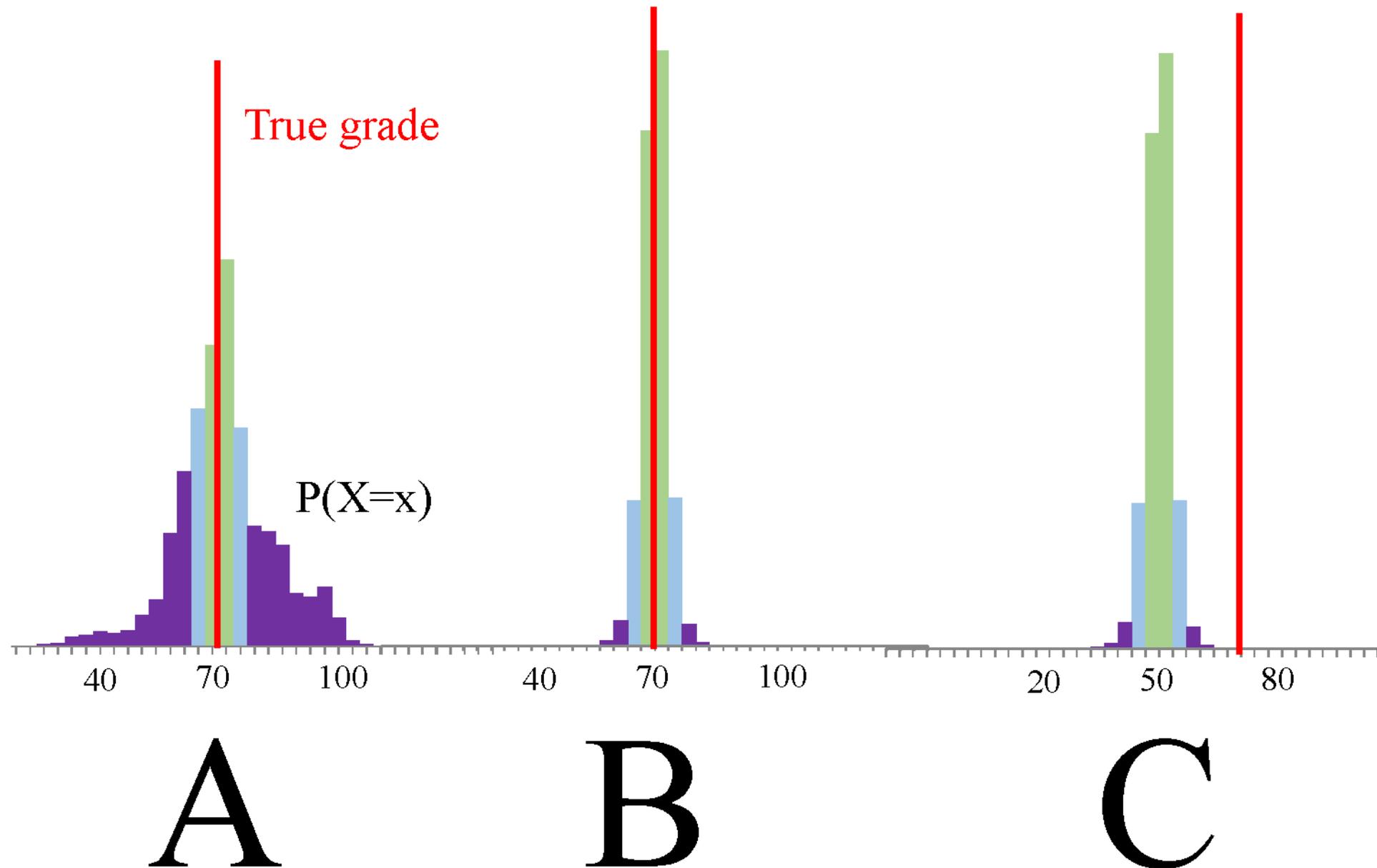
There are many others! Let's ask the computer to list the ways we could generate exactly  $k$  heads within  $n$  coin flips. The output region is scrollable:

(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)

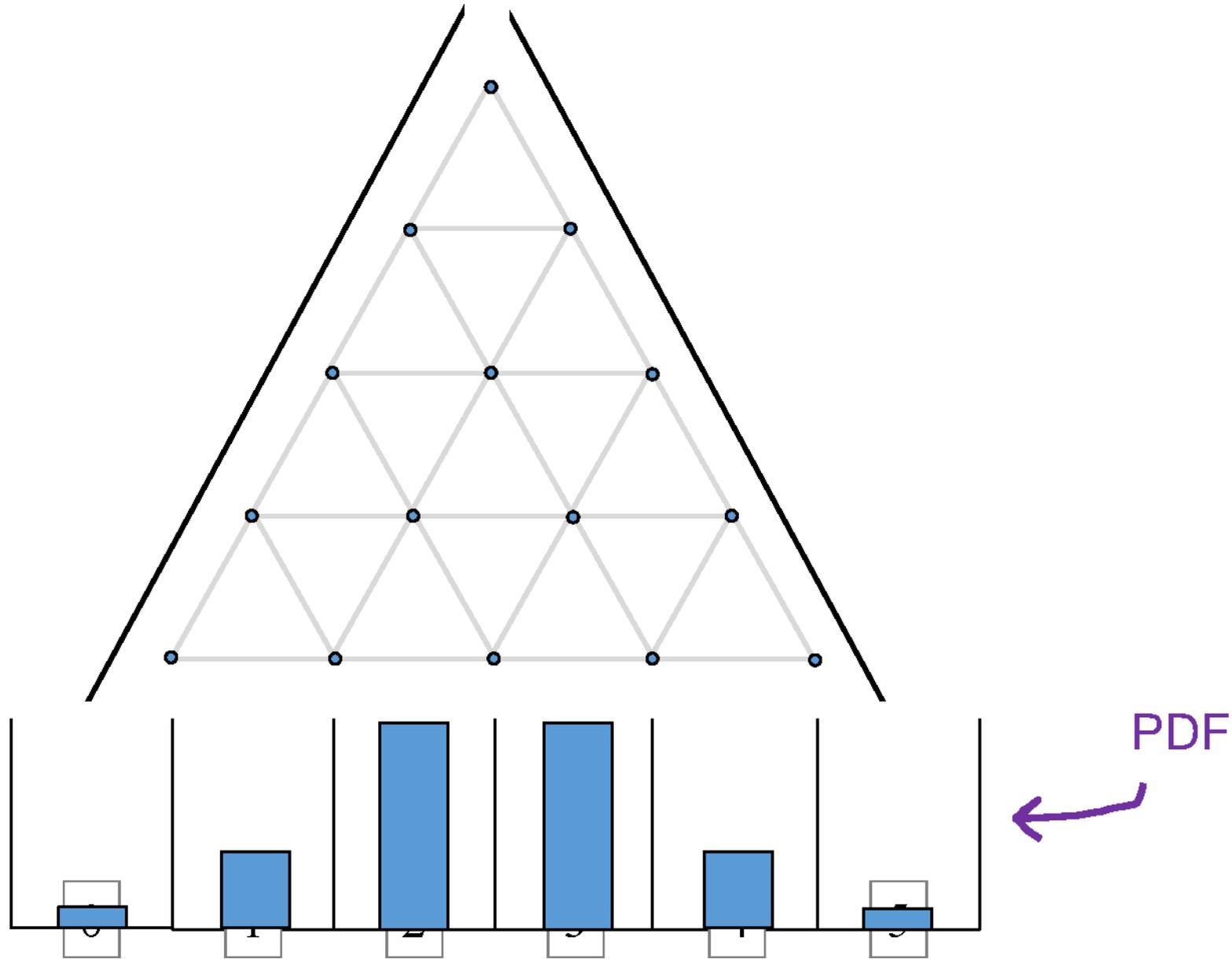


Random Variables

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



# Binomial

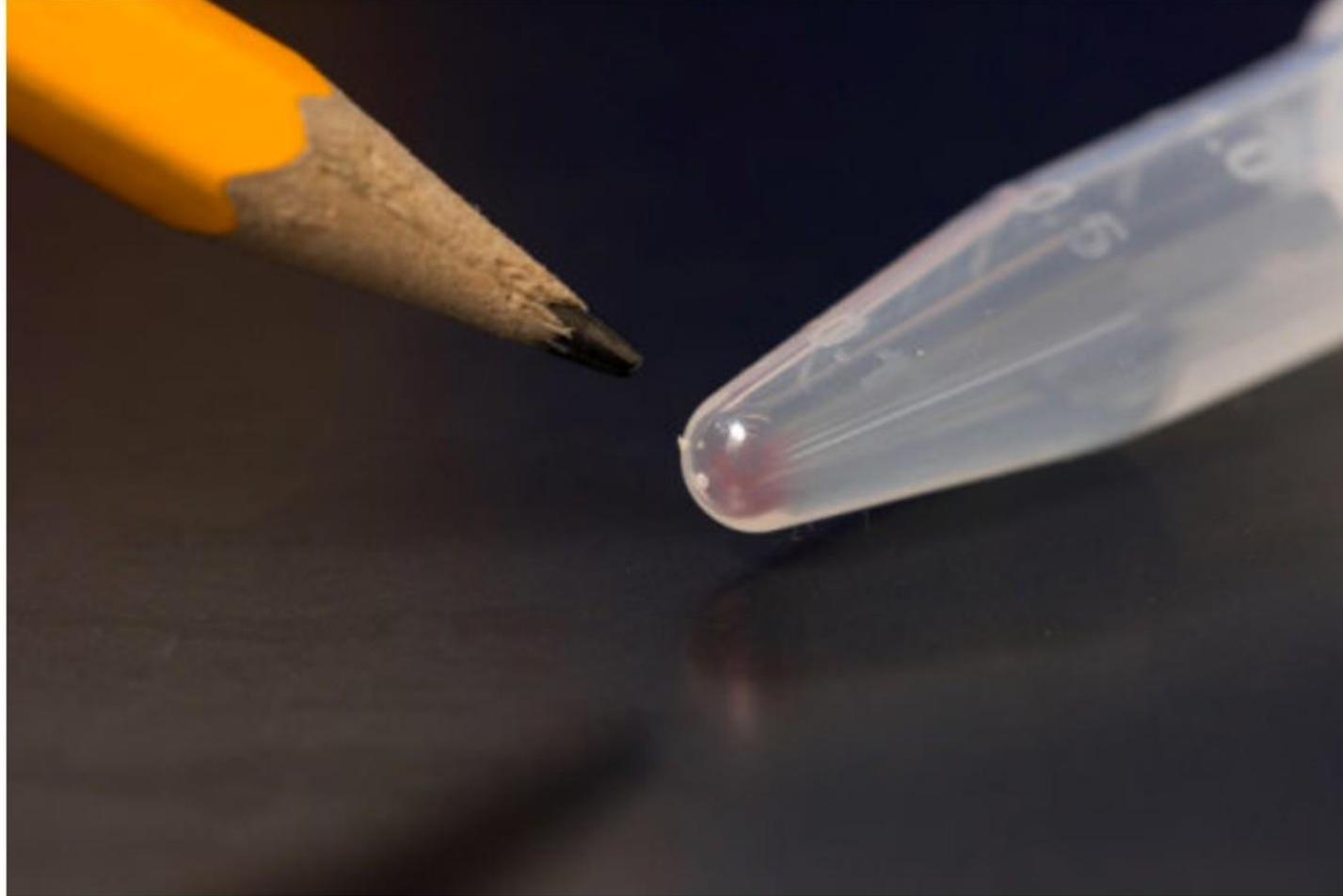


# Geometric

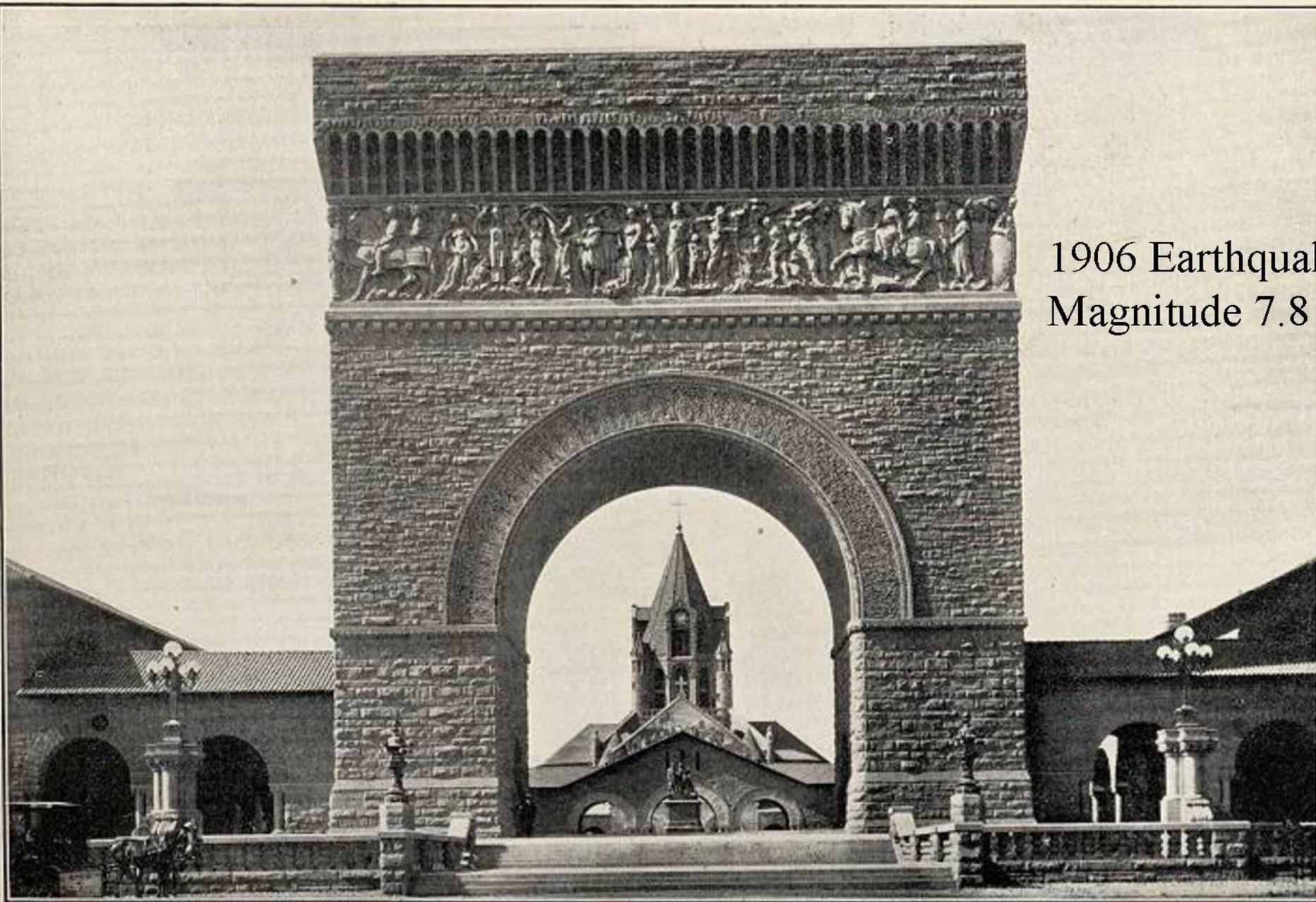
Sequence 1:

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?**

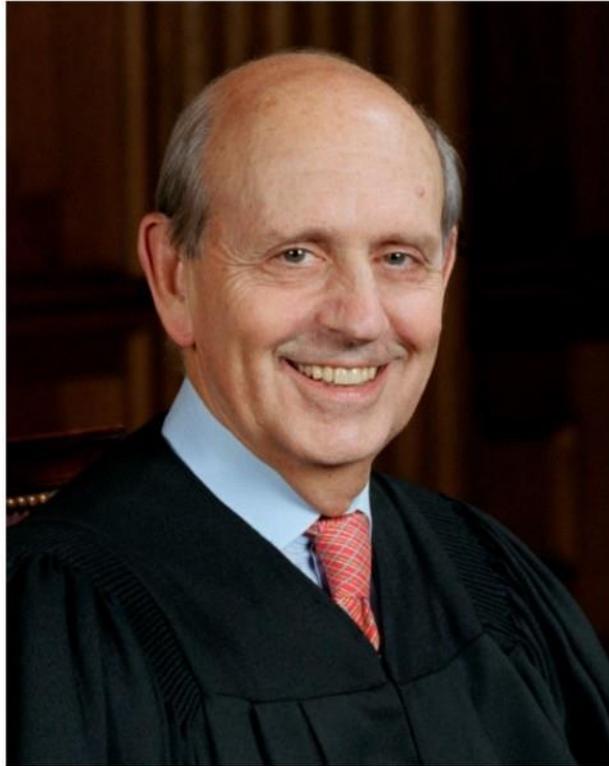


# Bit Coin Mining

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.

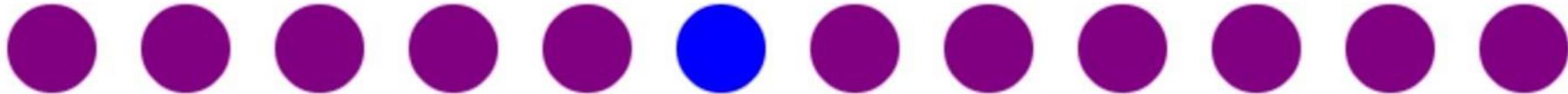


# Representative Juries



Simulate

Simulation:

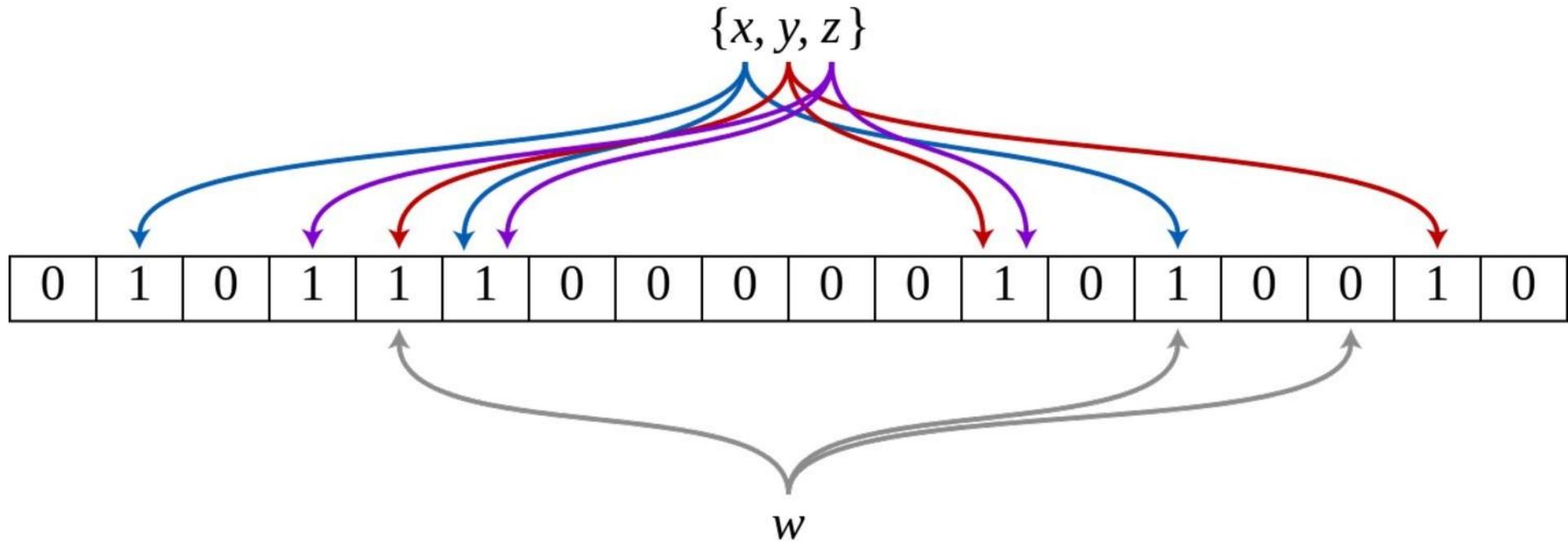


# Dating at Stanford

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( ) ?

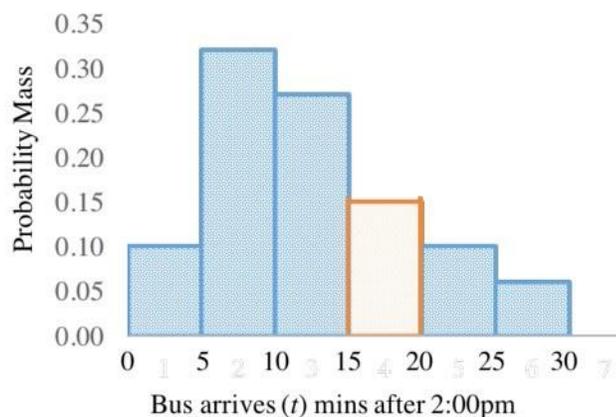
# Riding the Marguerite



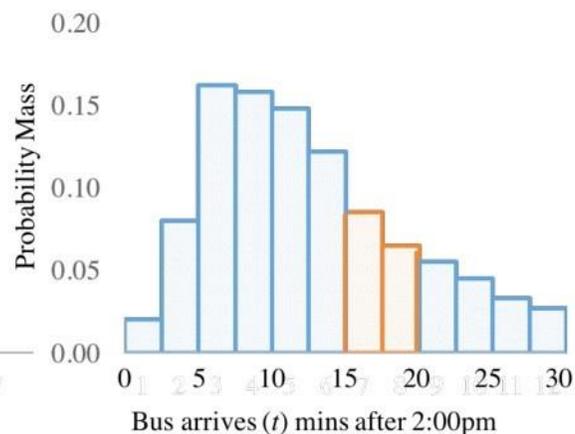
You are running to the bus stop.  
You don't know exactly when  
the bus arrives. You arrive at  
2:20pm.

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

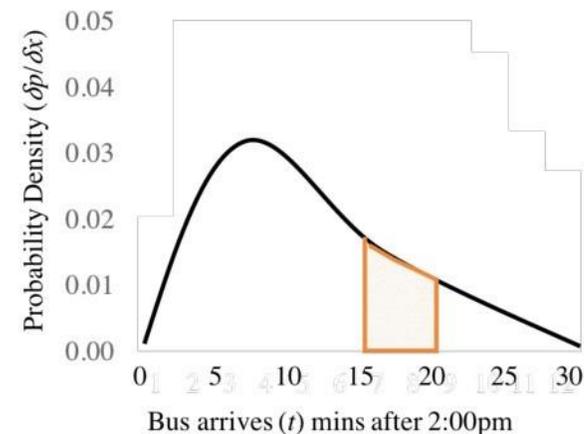
Discretize into 5 min chunks



Discretize into 2.5 min chunks



The limit at discretization size  $\rightarrow 0$



# Integrals

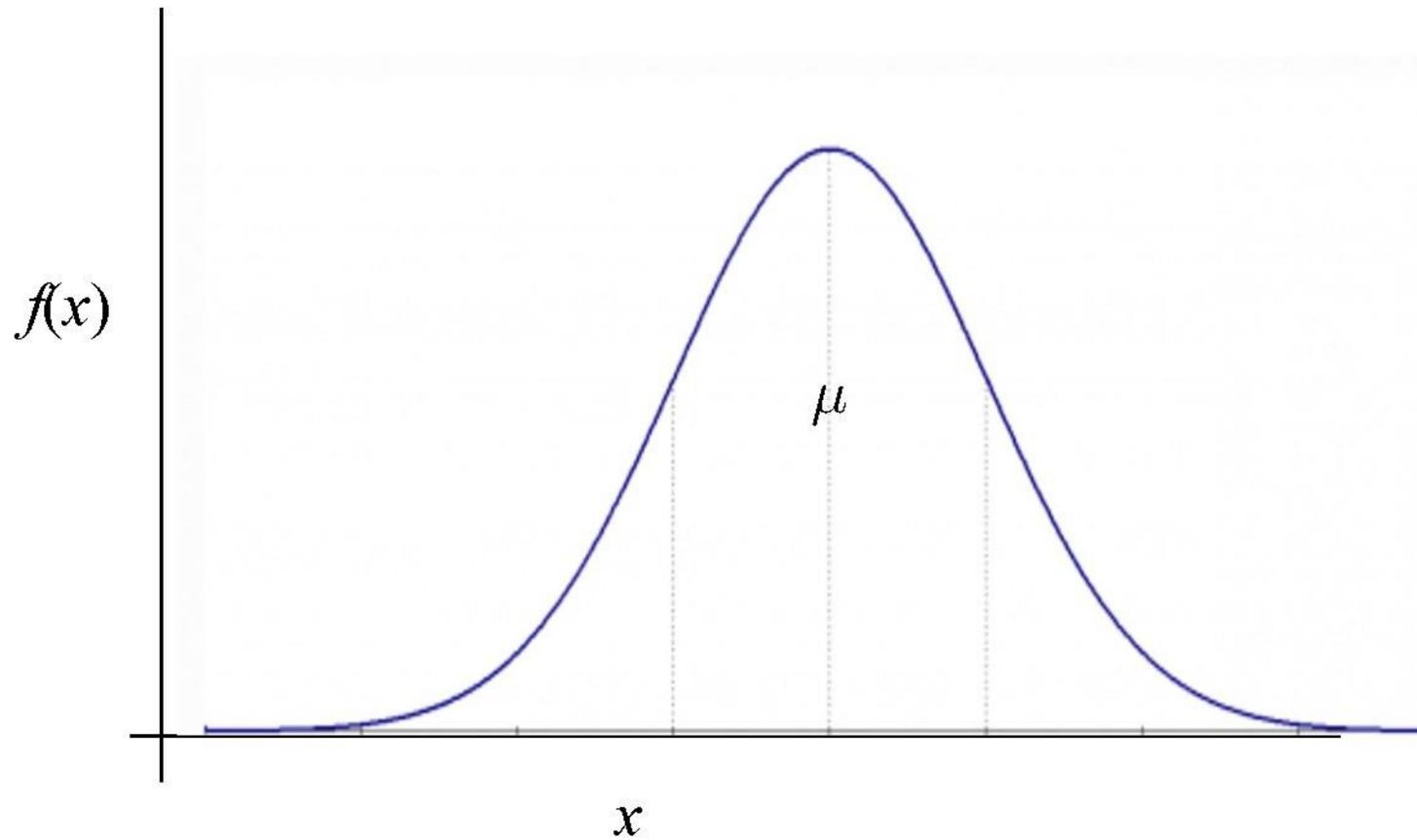


\*loving, not scary

# Probability Density Function

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

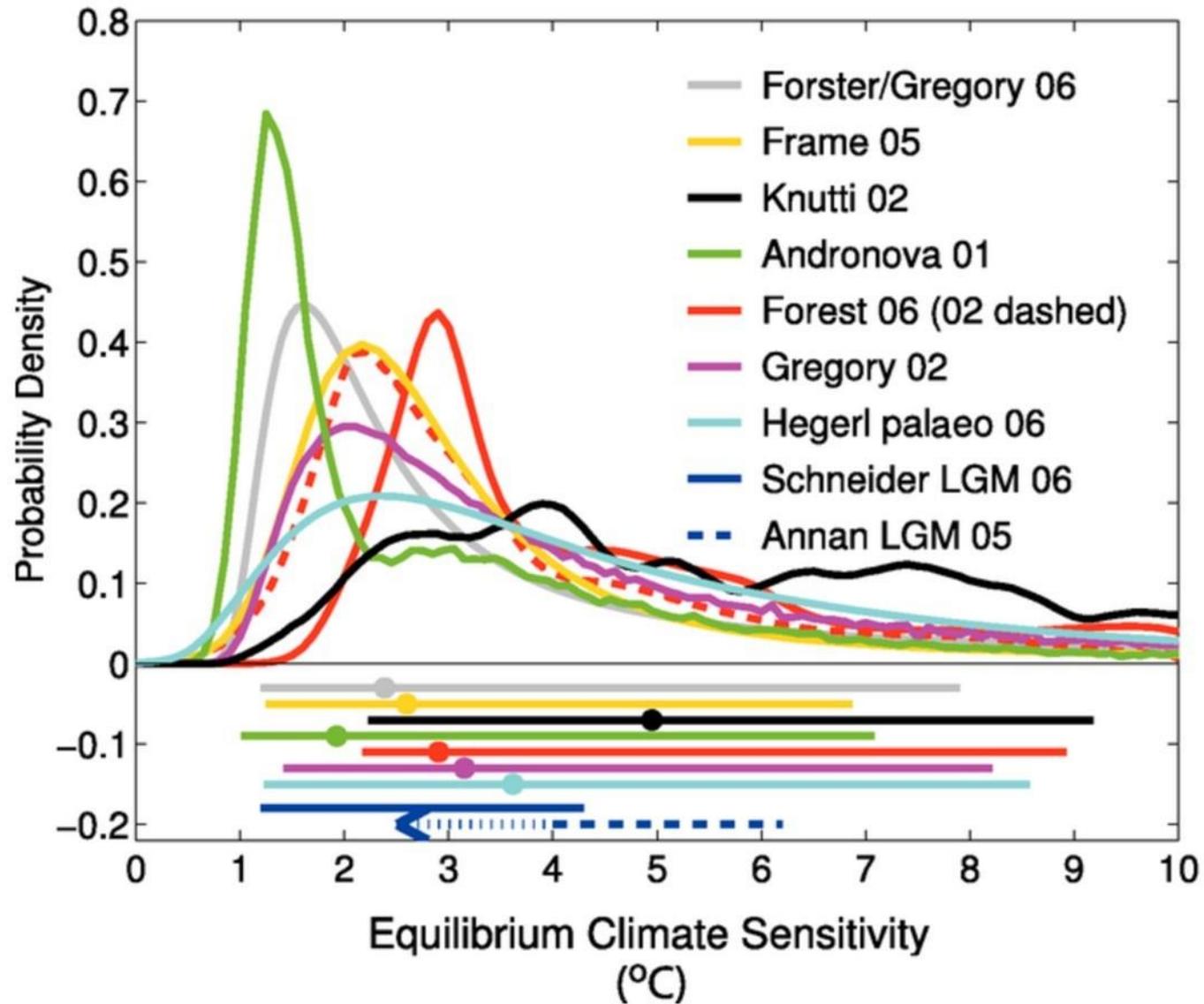
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



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

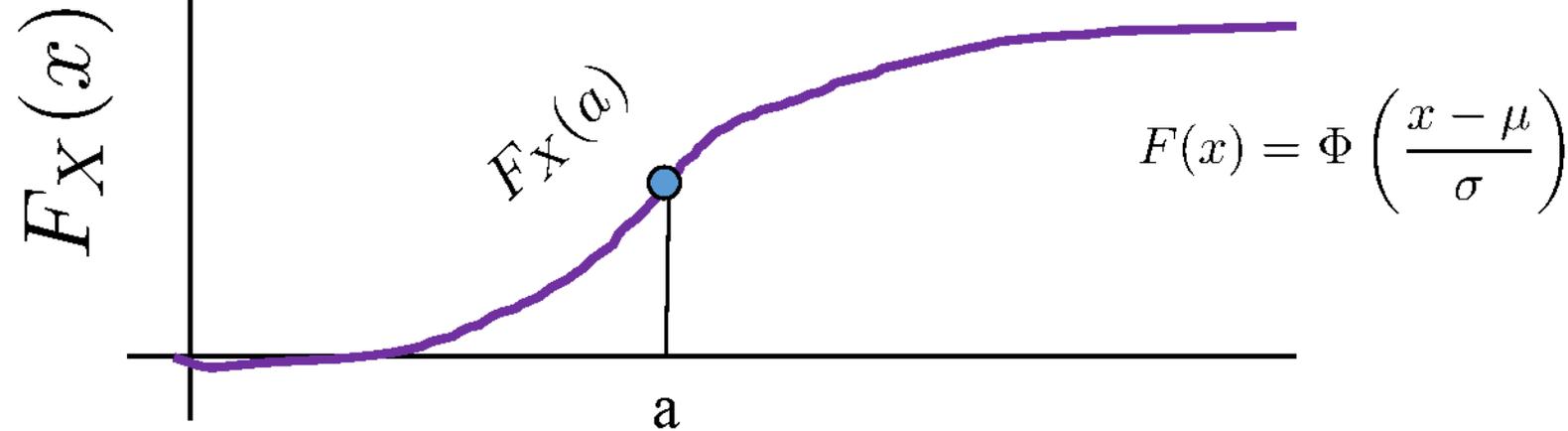
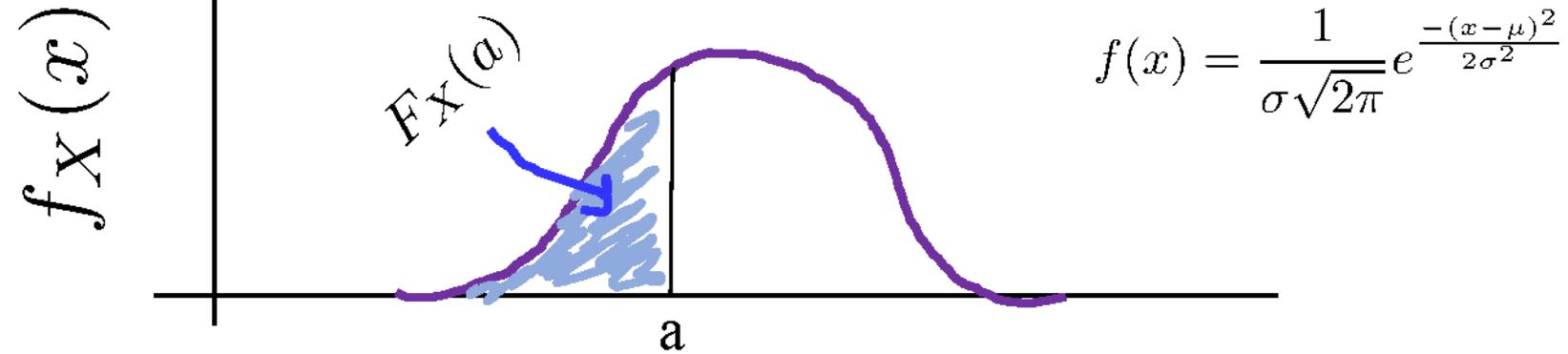
**A probability!**

# Climate Sensitivity



# PDF and CDF of a Normal

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



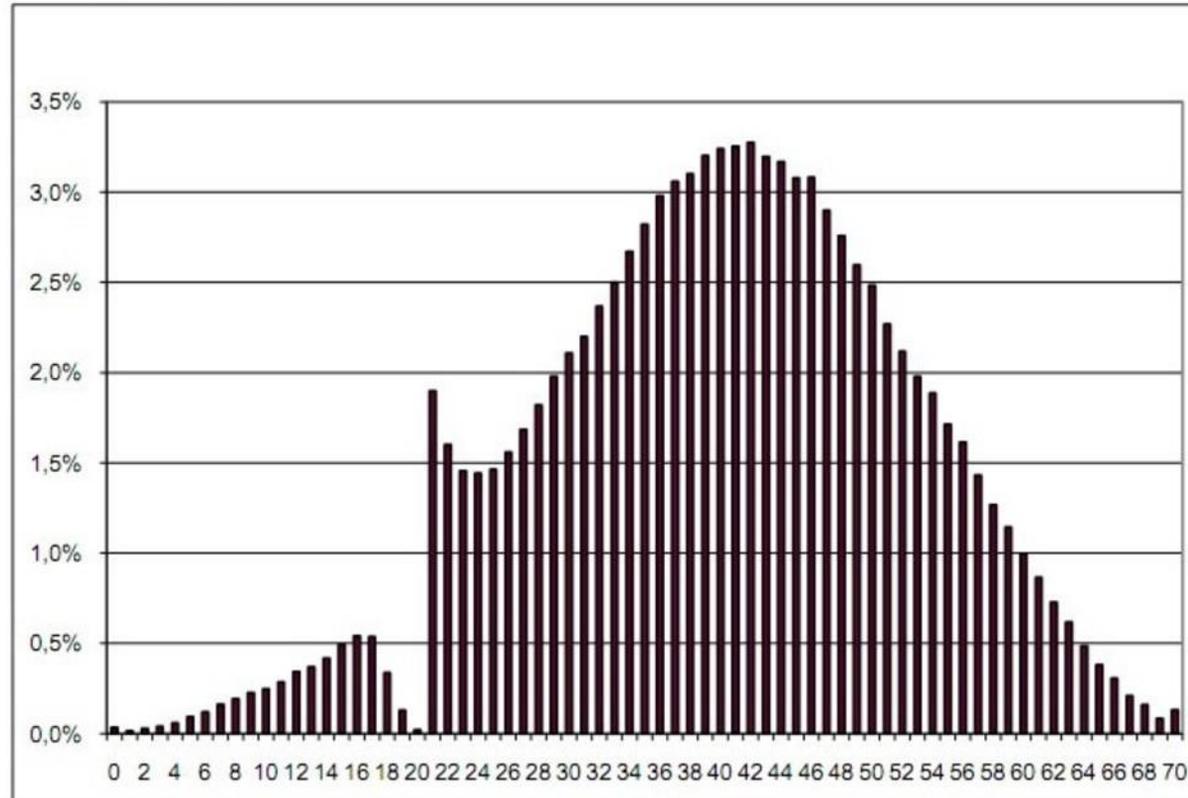
A CDF is the integral from  $-\infty$  to  $x$  of the PDF

# 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

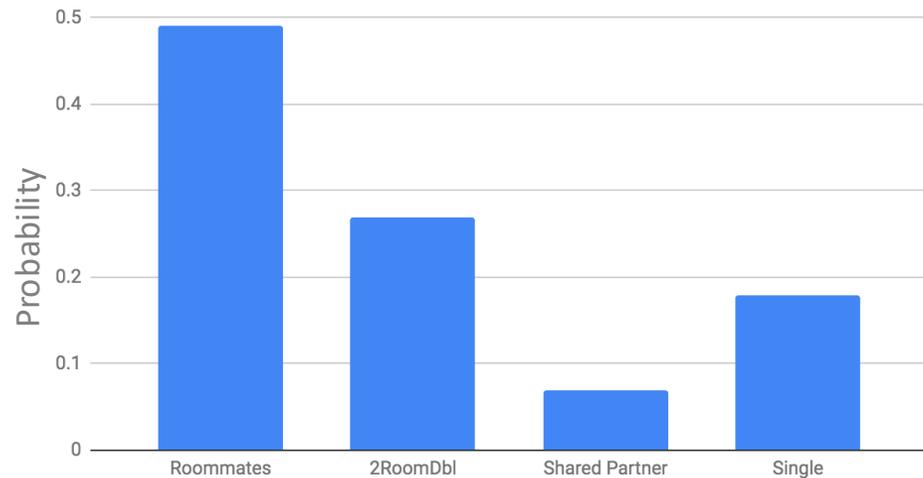
# Probabilistic Models



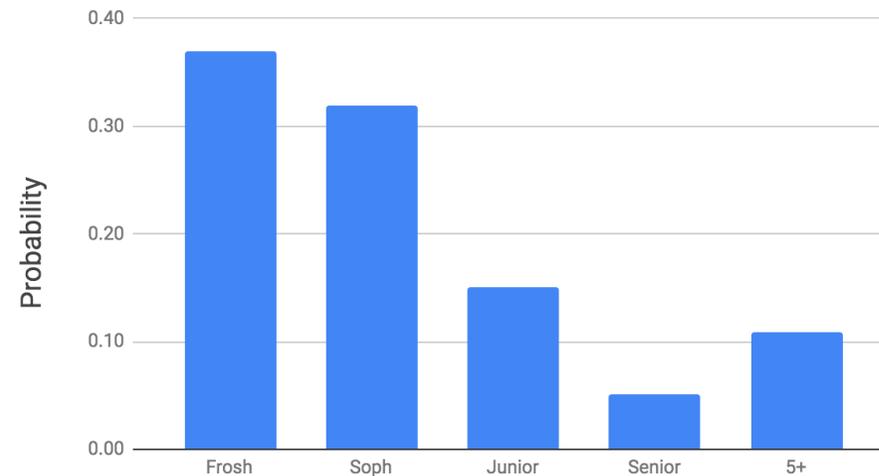
# Joint Probability Table

	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

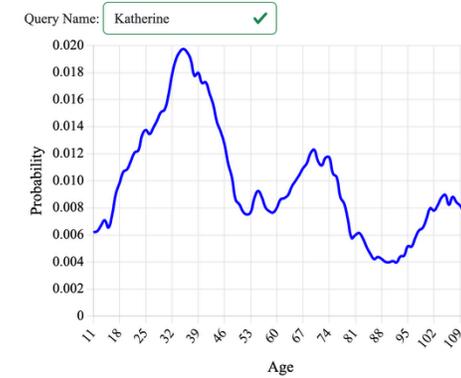
## Age from C14



## Updated Delivery Prob



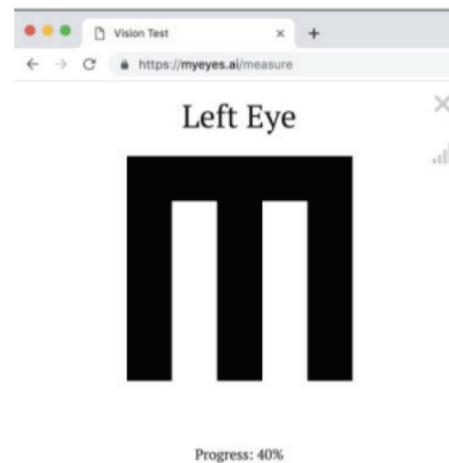
## Age from Name



## 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

or Choose common symptoms

- bloating
- cough**
- diarrhea
- dizziness
- fatigue
- fever
- headache
- muscle cramp
- nausea
- throat irritation

AGE 30

GENDER Male

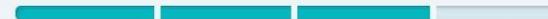
MY SYMPTOMS

cough ×

throat irritation ×

sneezing ×

Results Strength: **MODERATE**

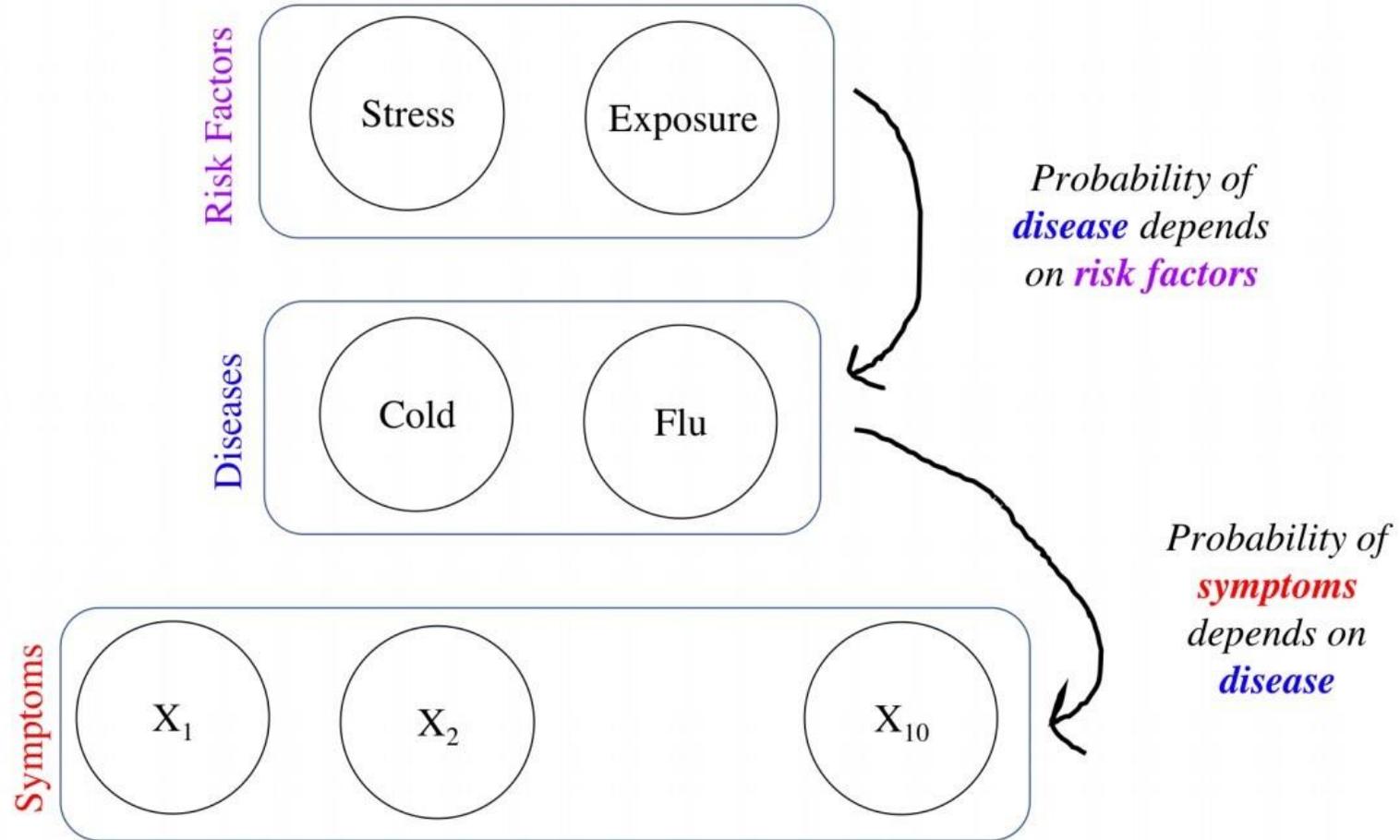


< Previous

Continue >

Info

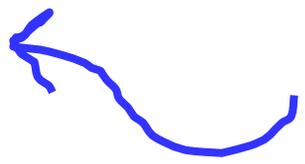
# Huge Joint Models



```
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.”

$n = 18$

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

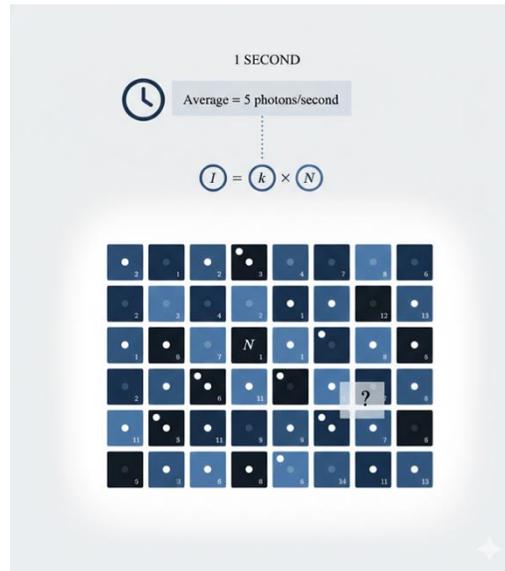
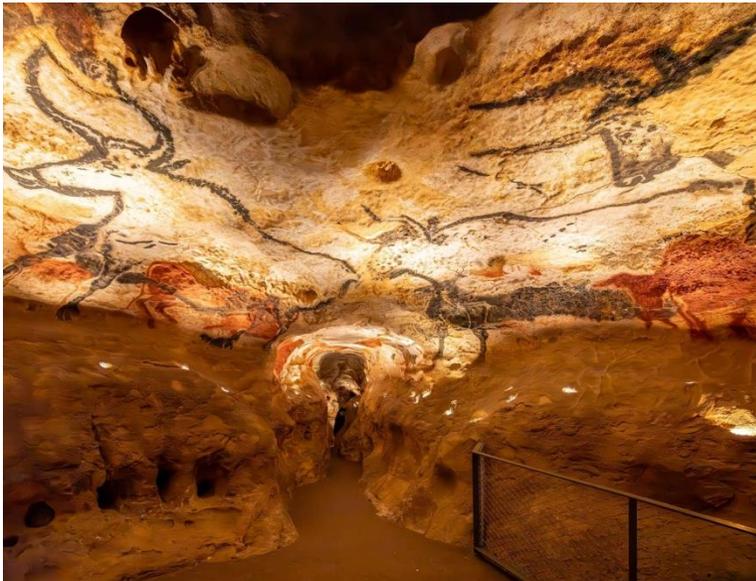
It's a Multinomial!

Probability of seeing  
this document | spam

The probability of a word in  
spam email being viagra



# Hard Midterm, Great Job



# PEP

## 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

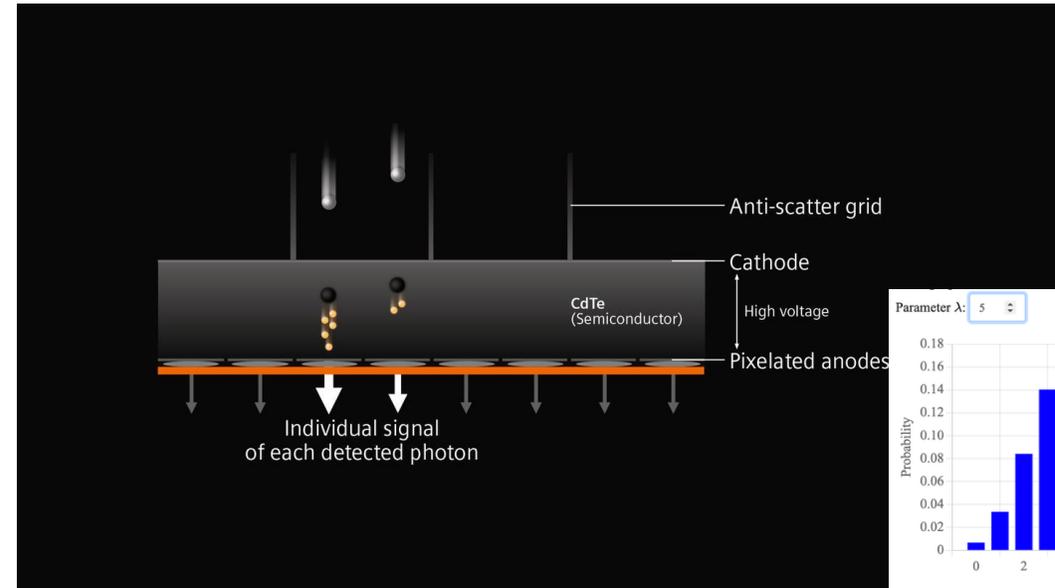
Tuesday, Dec 3

Select a time slot

Wednesday, Dec 4

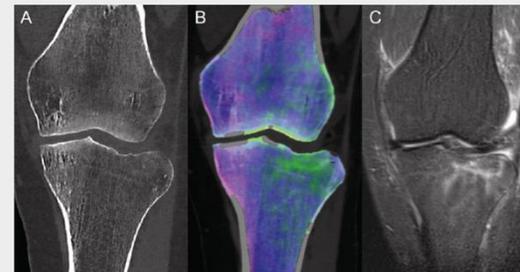
✓ Select a time slot

- 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

270 Midterm, 270 Final 134 Hours of Extra TA time

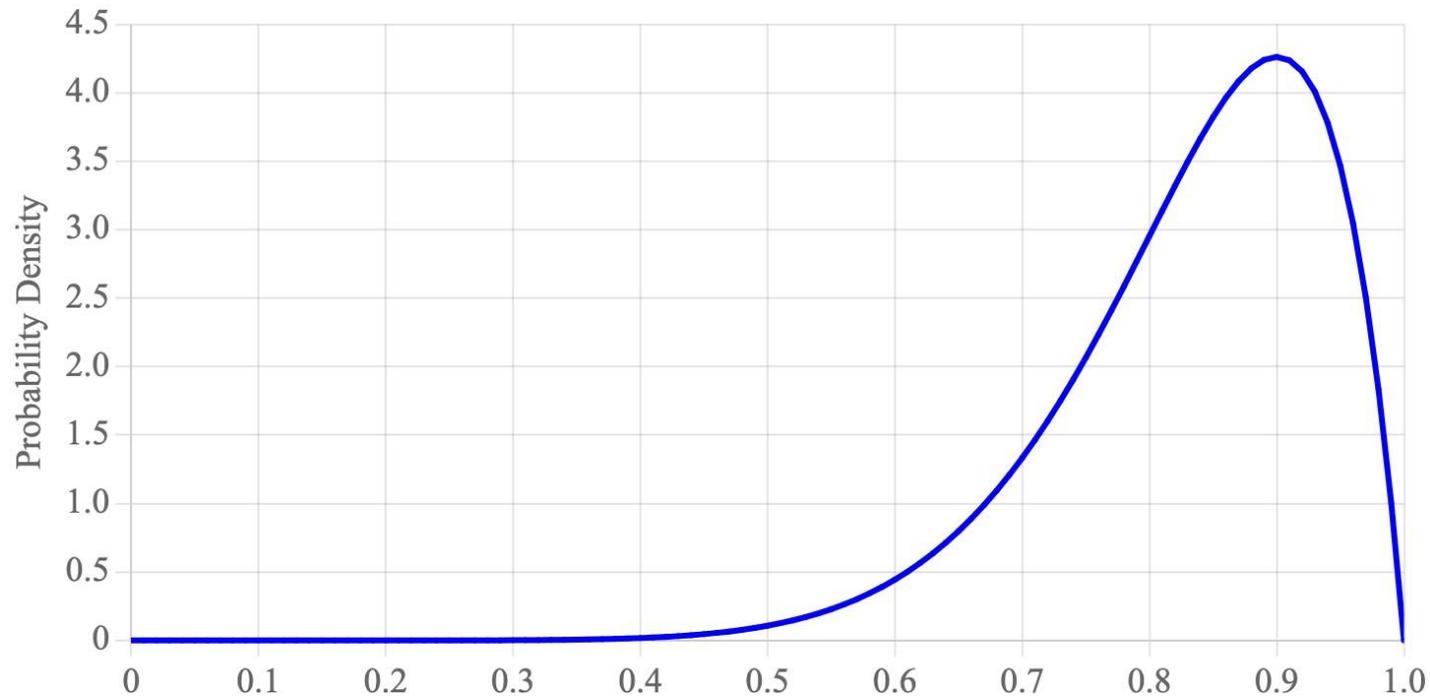
Learning Goal: Be fluent in the language of probability

# 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

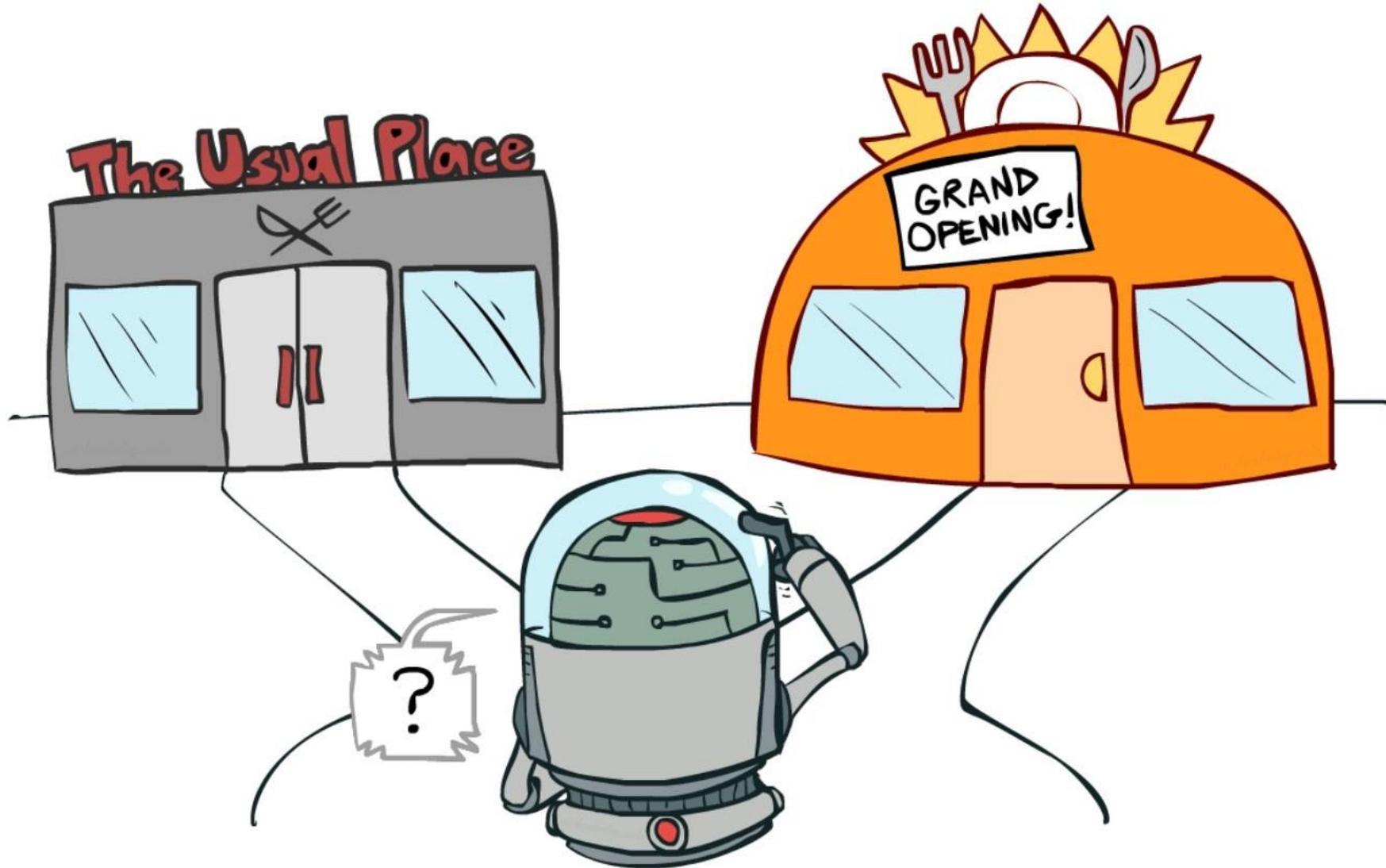


Drug B

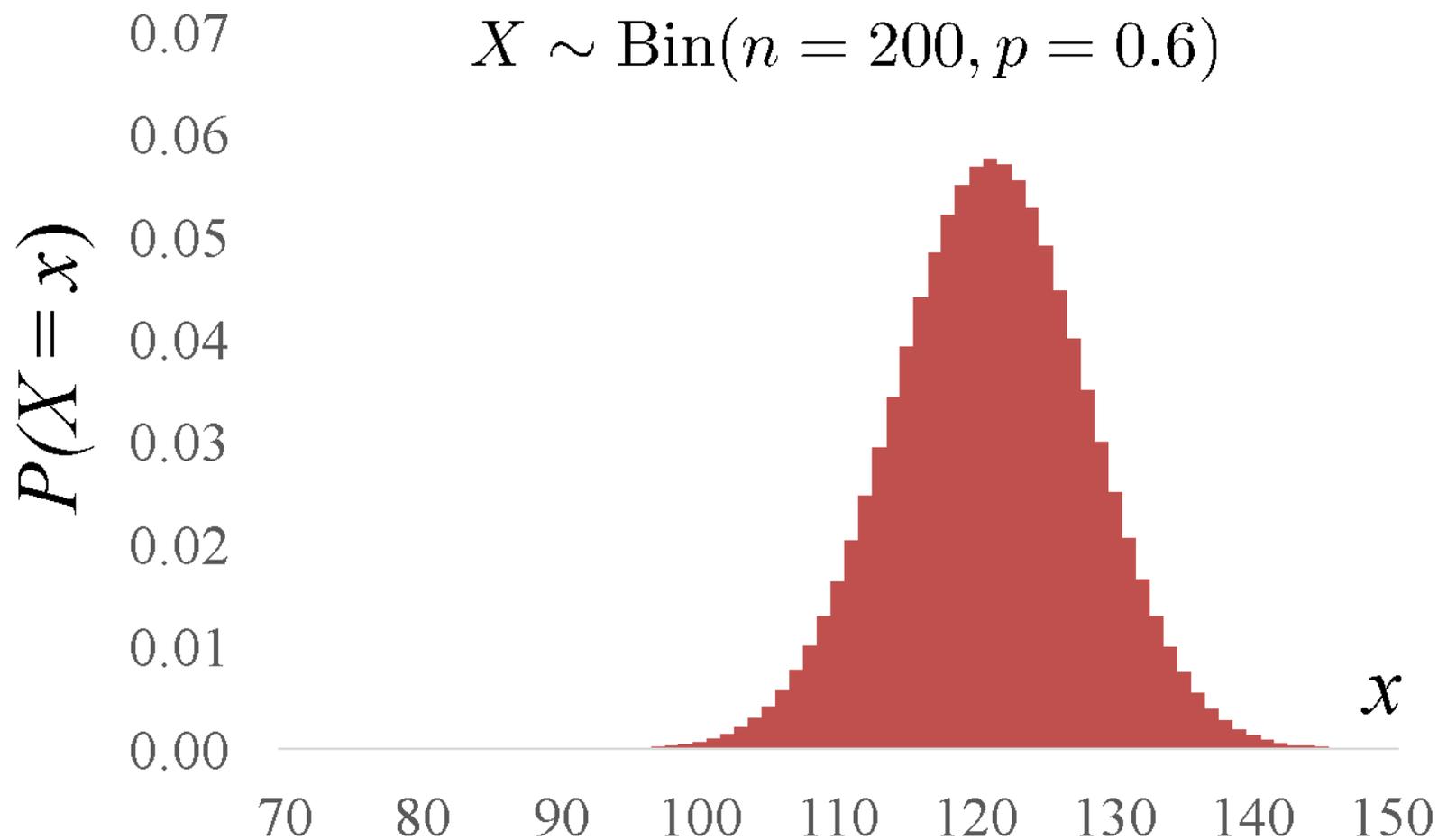


Which one do you give to a patient?

# Thompson Sampling



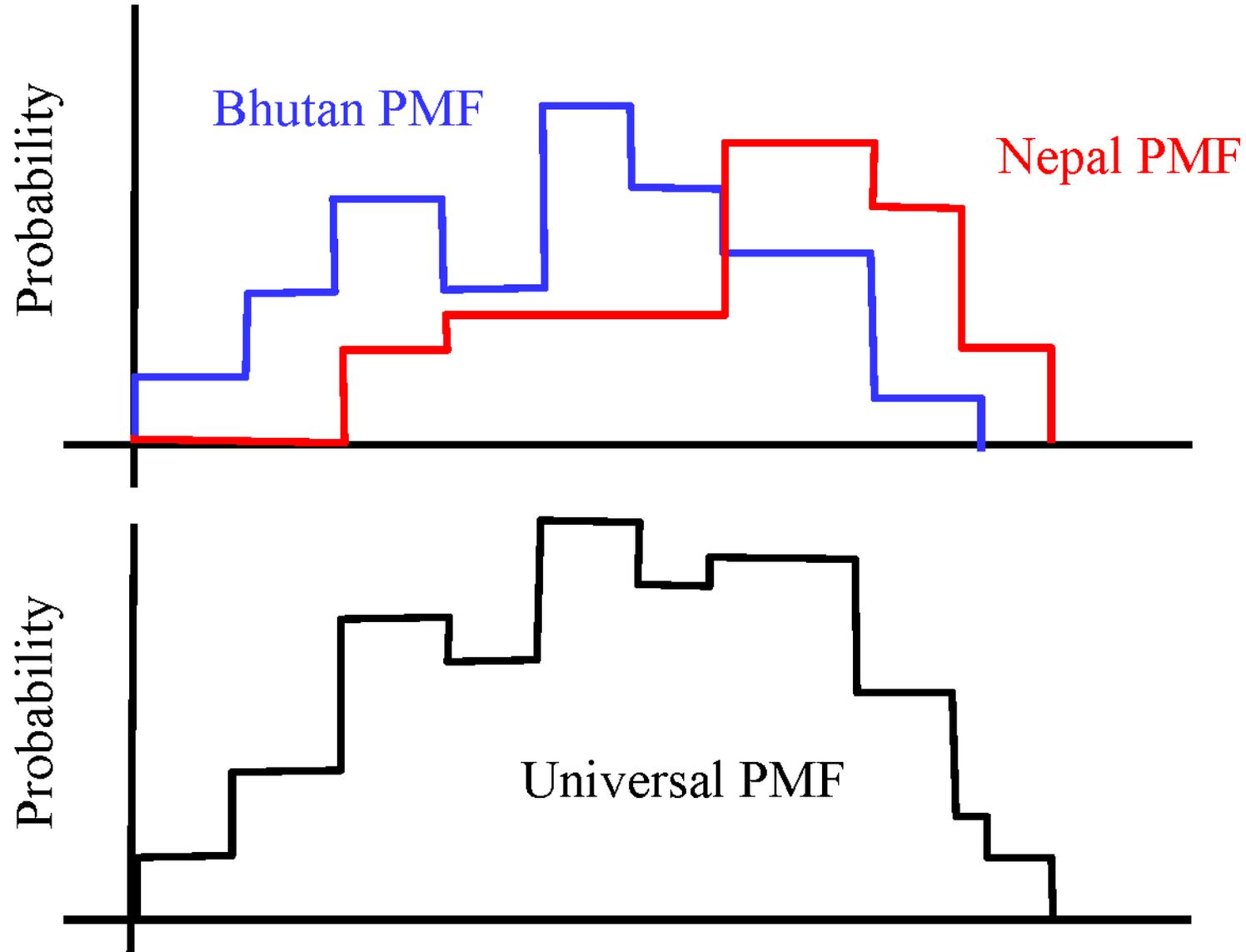
# C.L.T. Explains This



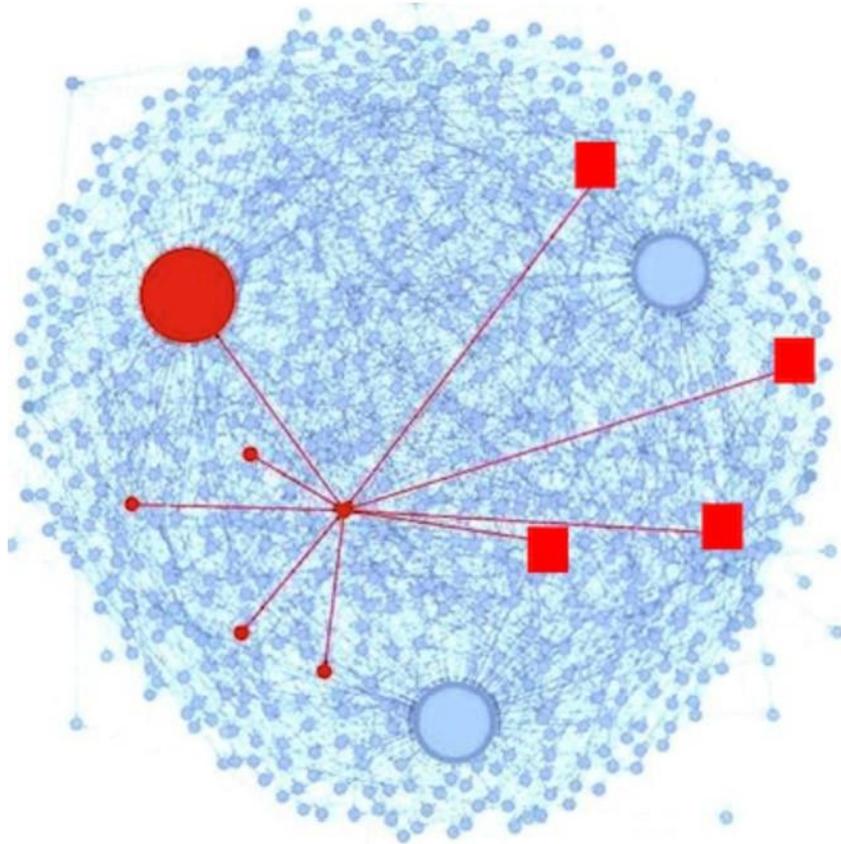
Bootstrap



# Universal Sample



# Peer Grading

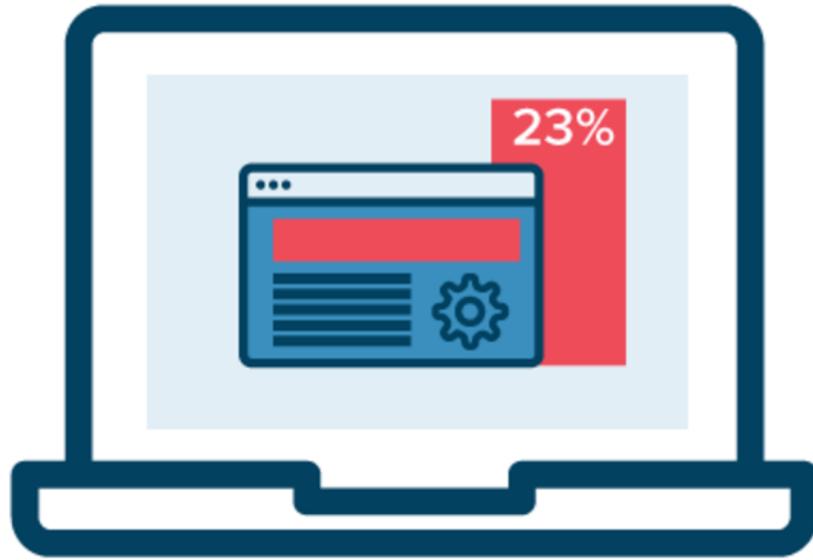


Peer Grading on Coursera  
HCI.

31,067 peer grades for  
3,607 students.

# A/B Testing

## A



CONTROL

## B



VARIATION



# Information Theory

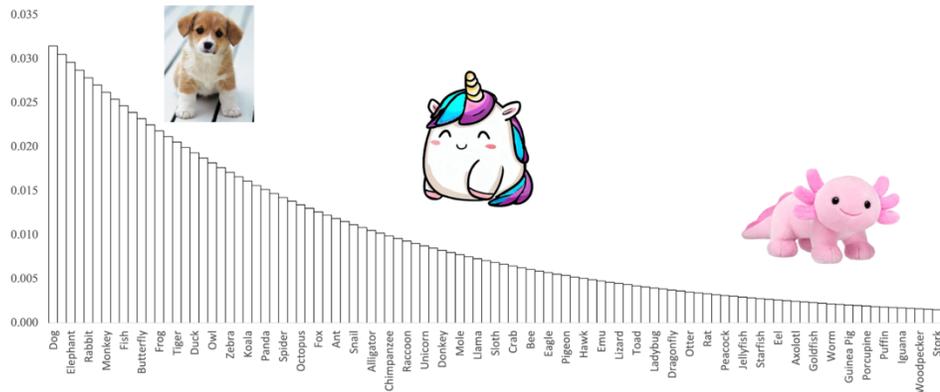


## Information Theory

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?

The screenshot shows a browser window with a Wordle game page. The game title is "Wordle" and the URL is "cs109psets.netlify.app/fall24/pset5/wordle". The game text says: "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?". Below the text is a 5x3 grid representing the Wordle board. The grid shows the following colors: Row 1: Grey, Green, Grey, Grey, Yellow; Row 2: Yellow, Green, Green, Grey, Grey; Row 3: Green, Green, Green, Green, Green. A cartoon robot is positioned in the center of the grid. To the right of the grid is a Python code editor with the following code:

```
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"],
        "xYx": ["storm"]
    }
```

Below the code editor is a "Run" button. At the bottom of the page, there are "Previous Question" and "Next Question" buttons. The page number "85" is visible in the bottom right corner.

# 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

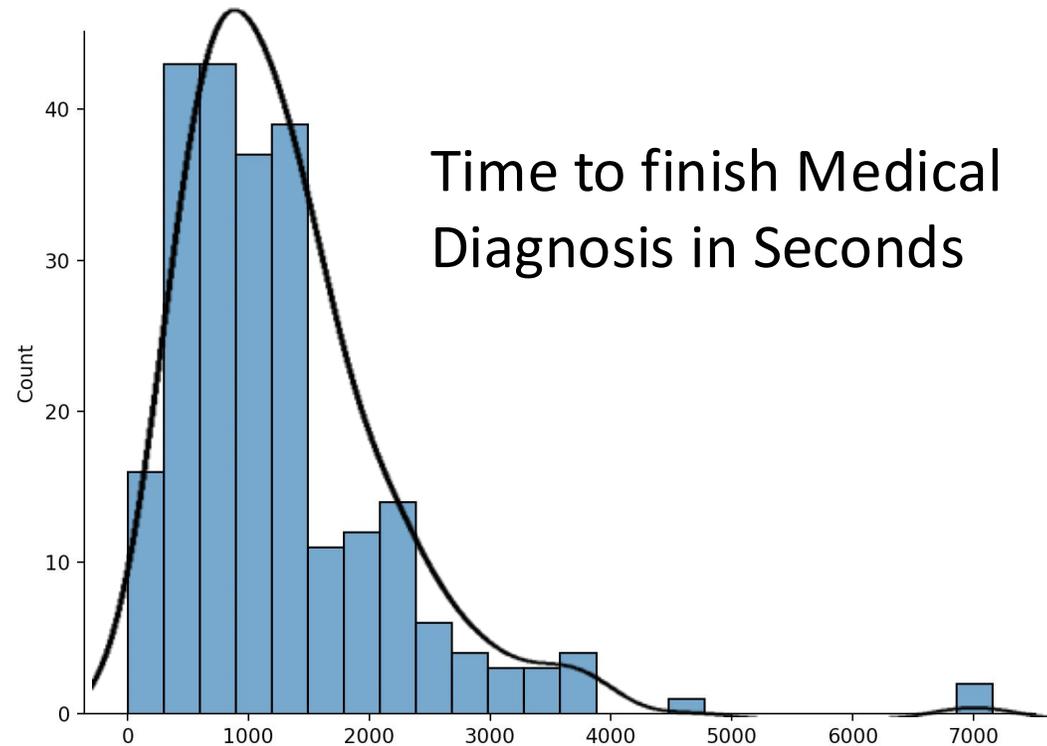
True Probability (p)	Guess Probability	Your score_guess function	exp(-10 * KL(X, Y))
0.0	0.0	0.0	0.0
0.1	0.1	0.0	0.0
0.2	0.2	0.0	0.0
0.3	0.3	0.0	0.0
0.4	0.4	0.0	0.05
0.5	0.5	0.0	0.2
0.6	0.6	0.0	0.6
0.7	0.7	0.0	0.95
0.777	0.777	0.0	1.0
0.8	0.8	0.0	0.95
0.9	0.9	0.0	0.6
1.0	1.0	0.0	0.0

# Machine Learning



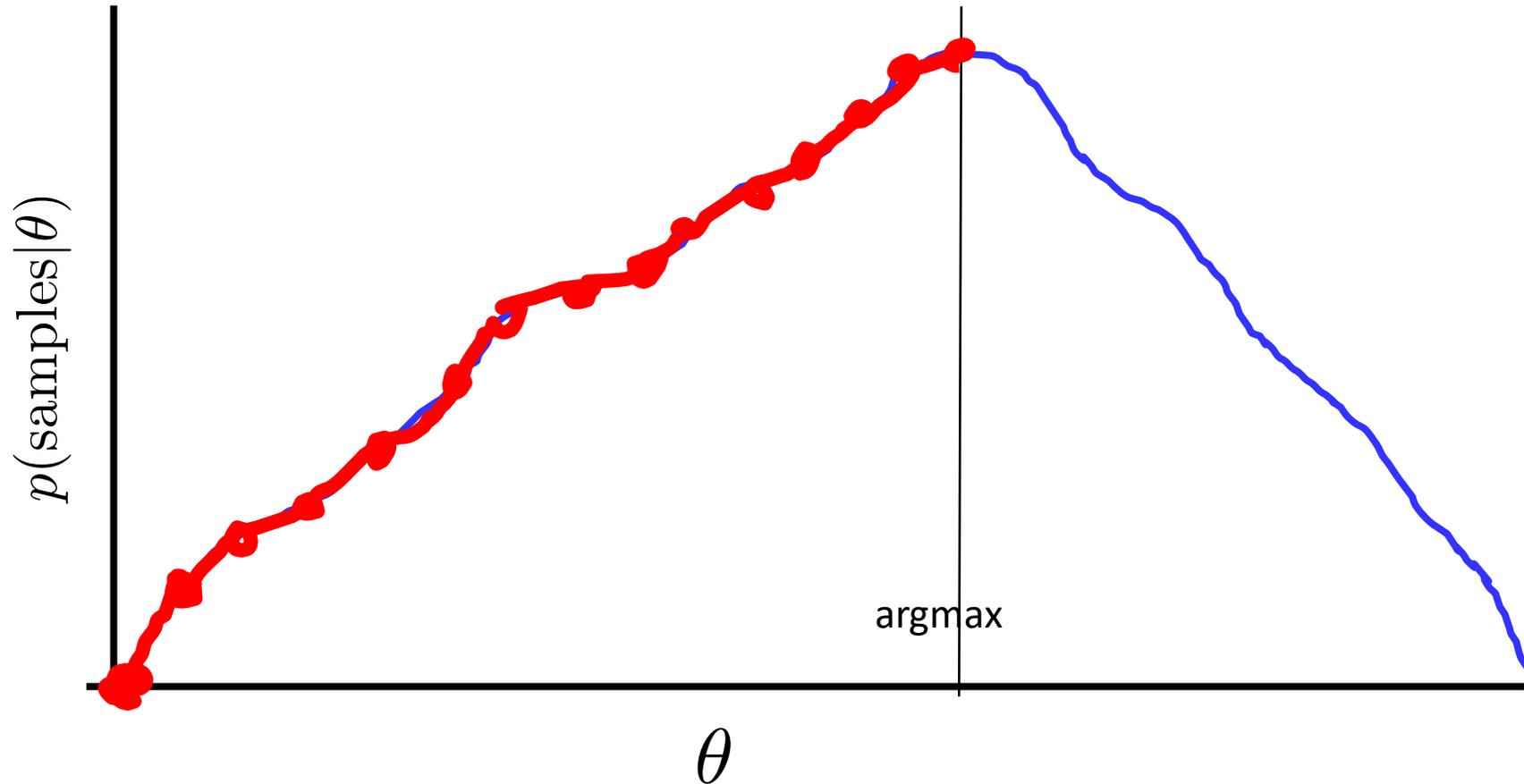
# MLE for Choosing Params

```
[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]
```



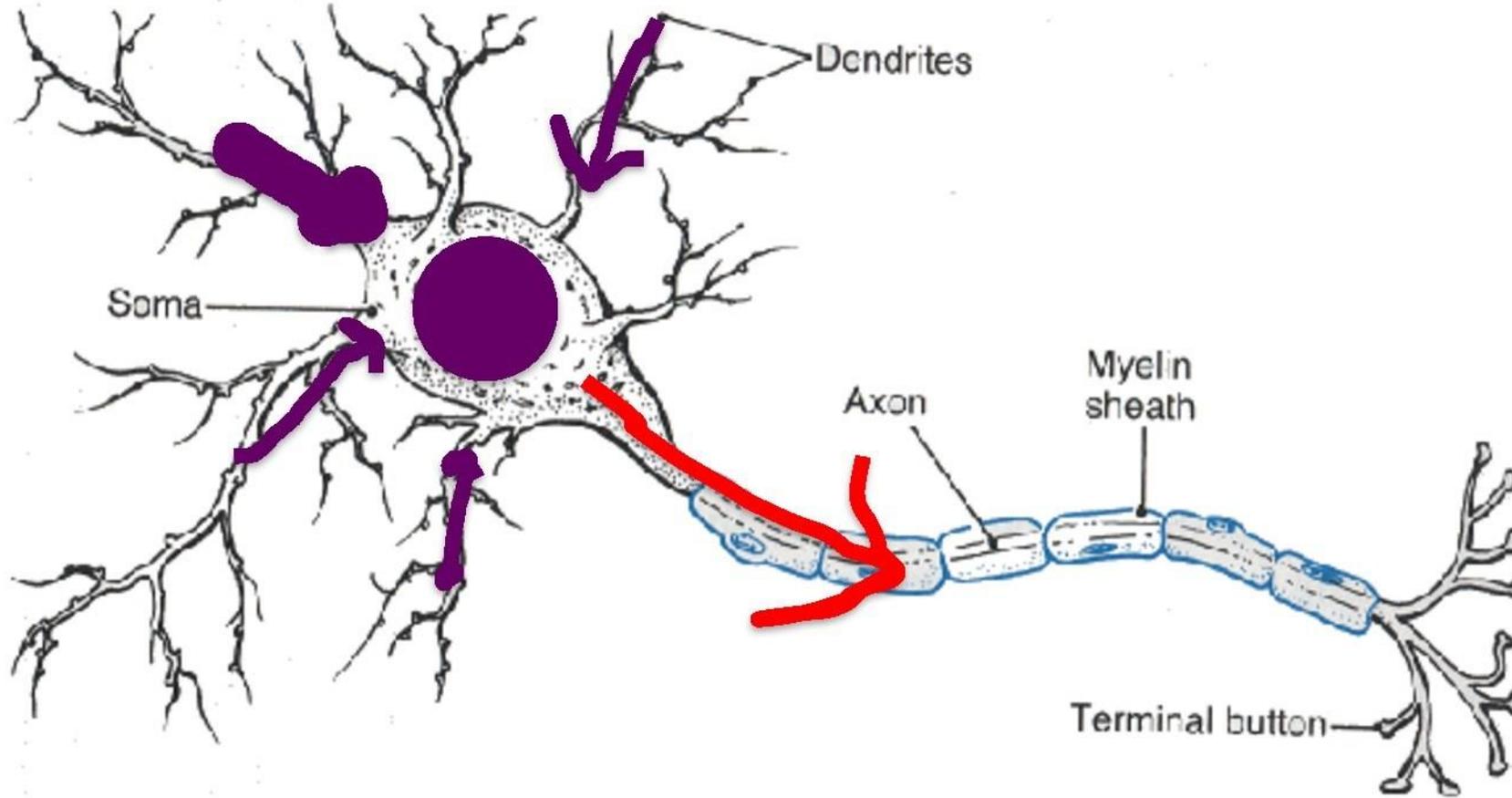
$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}$$

# Gradient Ascent



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

# Logistic Regression



# Machine Learning

Heart



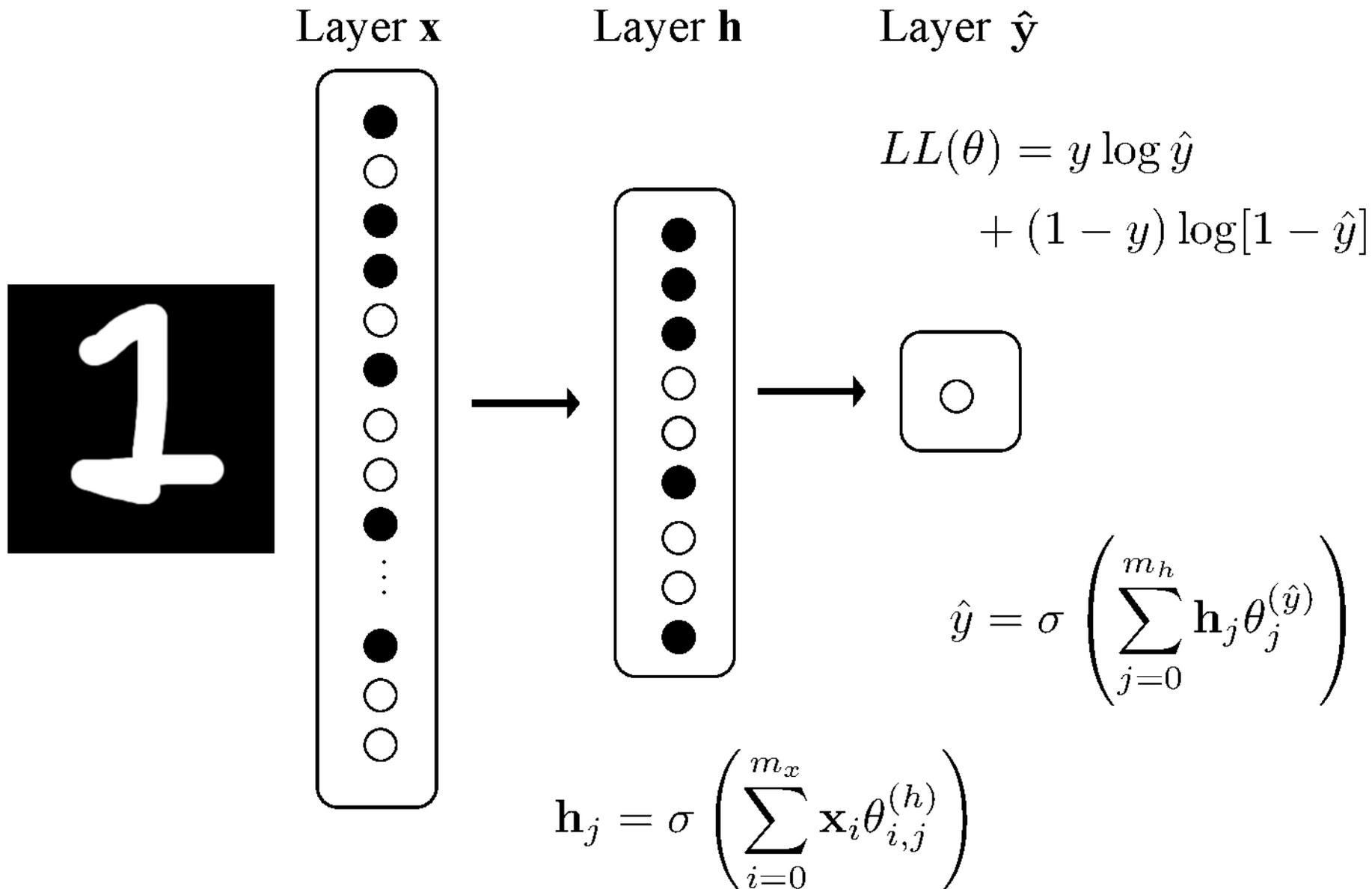
Ancestry



Netflix

**NETFLIX**

# Deep Learning



# Calibration

New!

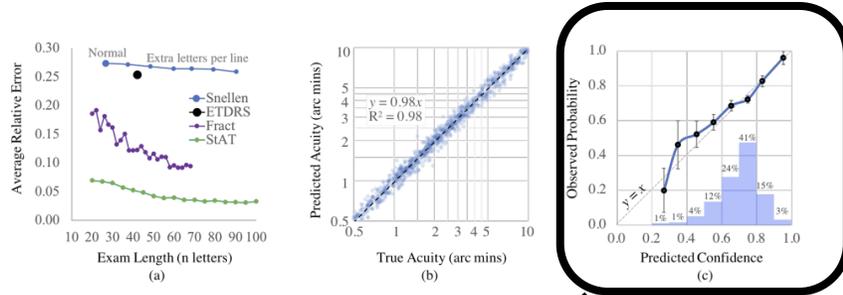


Figure 4: (a) The tradeoff between length of exam and error for the different algorithms. (b) Visualization of the predictions made by StACT. (c) Calibration test: StACT confidences correspond to how often it is correct.

## 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

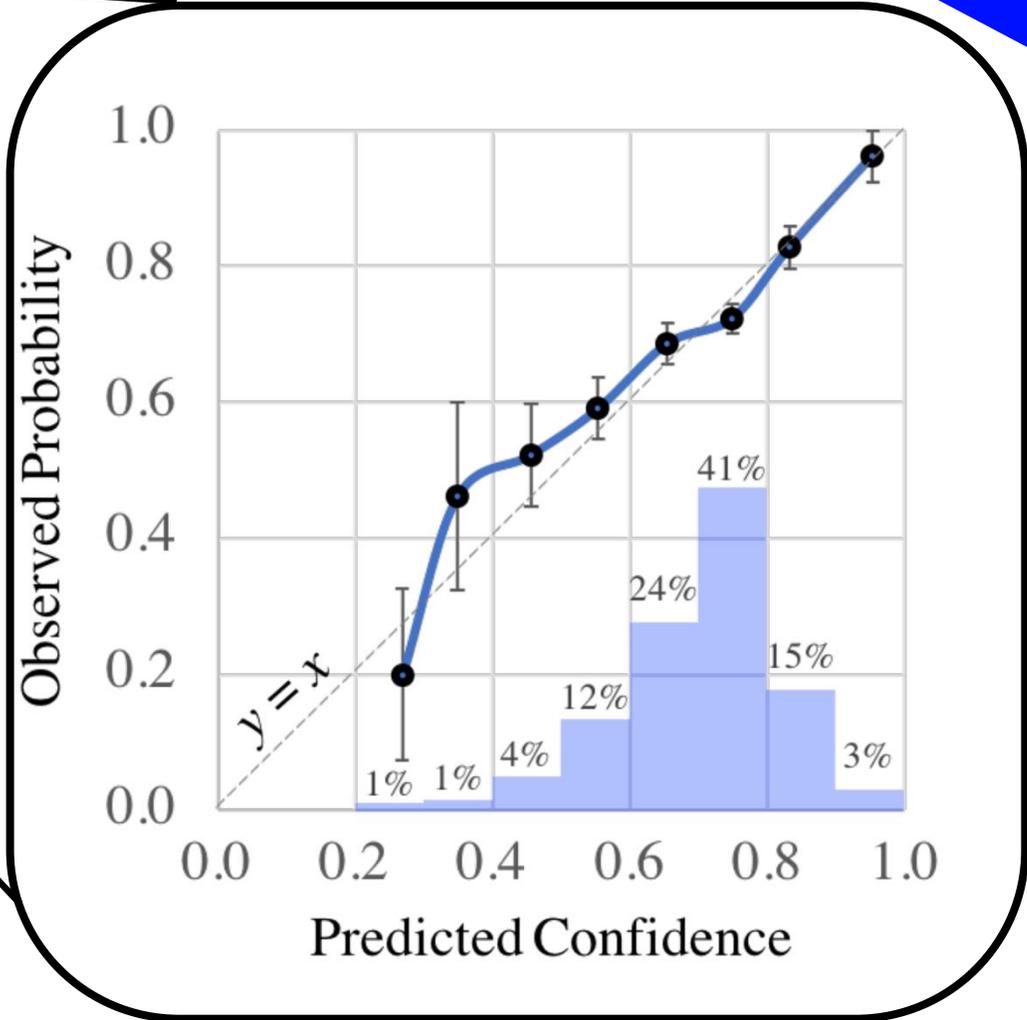
Table 1: Average relative error for each algorithm. Except for Snellen each test was allowed 20 letters. Results are average relative error after 1000 tests. <sup>†</sup> Snellen and ETDRS used 19 unique optotypes.

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.

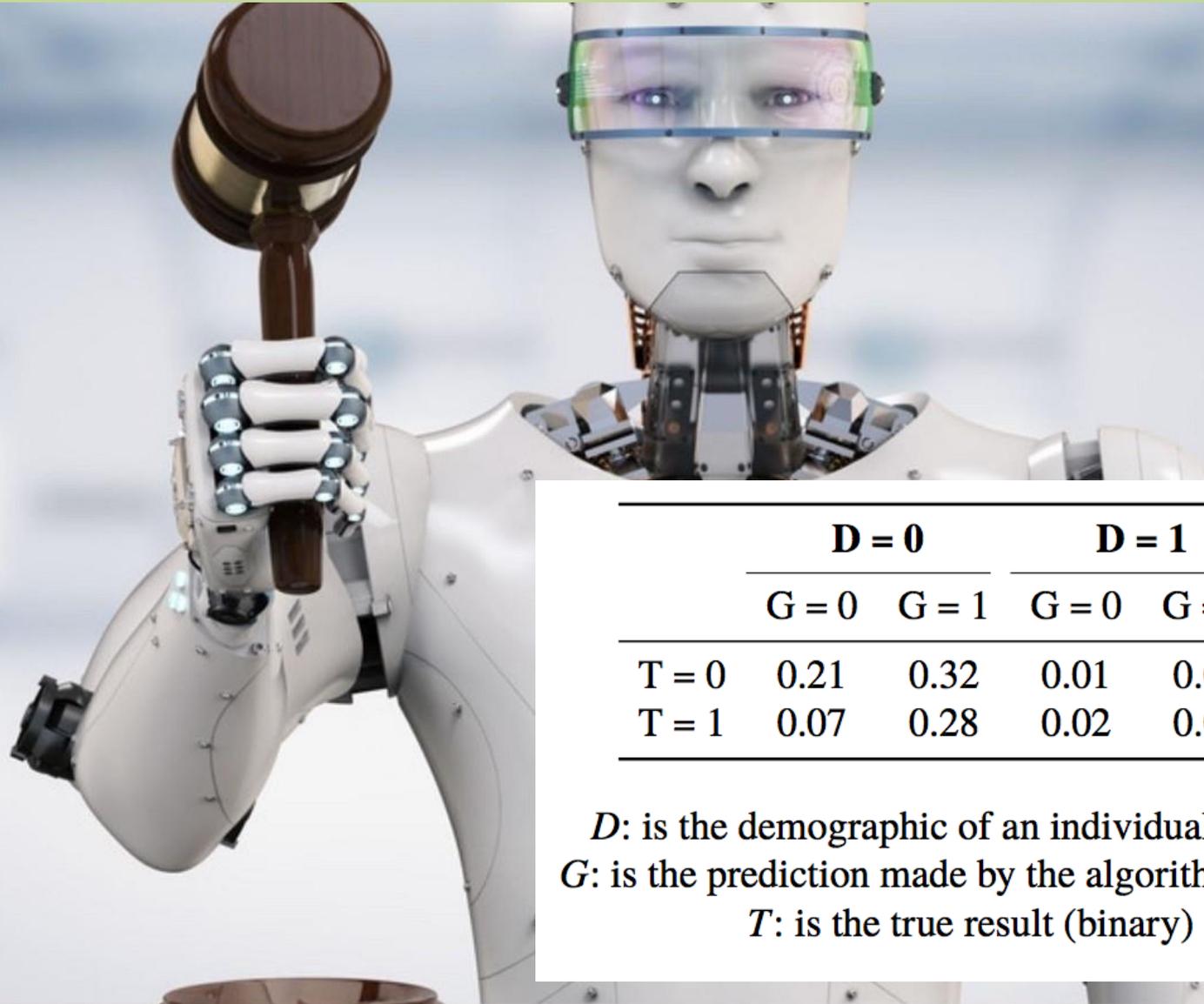
Figure 4 (b) visualizes what StACT's small relative error means in terms of predictions. Each point in the plot is a single patient. The x-axis is the true acuity of the patient and the y-axis is the predicted accuracy. We can qualitatively observe that the predictions are often accurate, there are no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



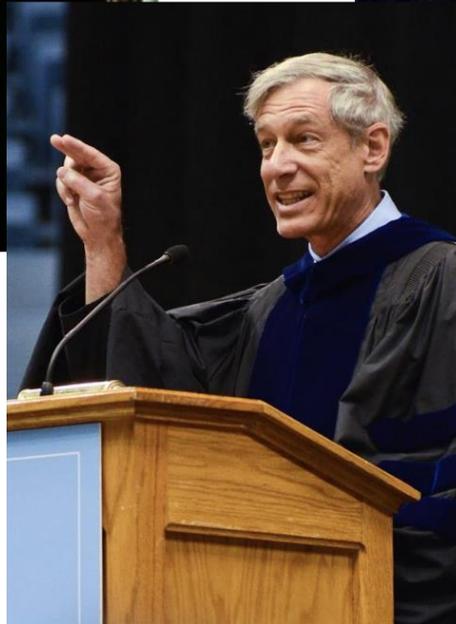
Reinforcement!

# Night Sight



Mark Levoy, Stanford Emeritus Professor

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



# Serendipity



**WHEN YOU MEET YOUR BEST FRIEND**

Somewhere you didn't expect to.

# Wisdom of the Crowds



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

**A:**

**B:** Peregrine Falcon

**C:** Humpback Whale

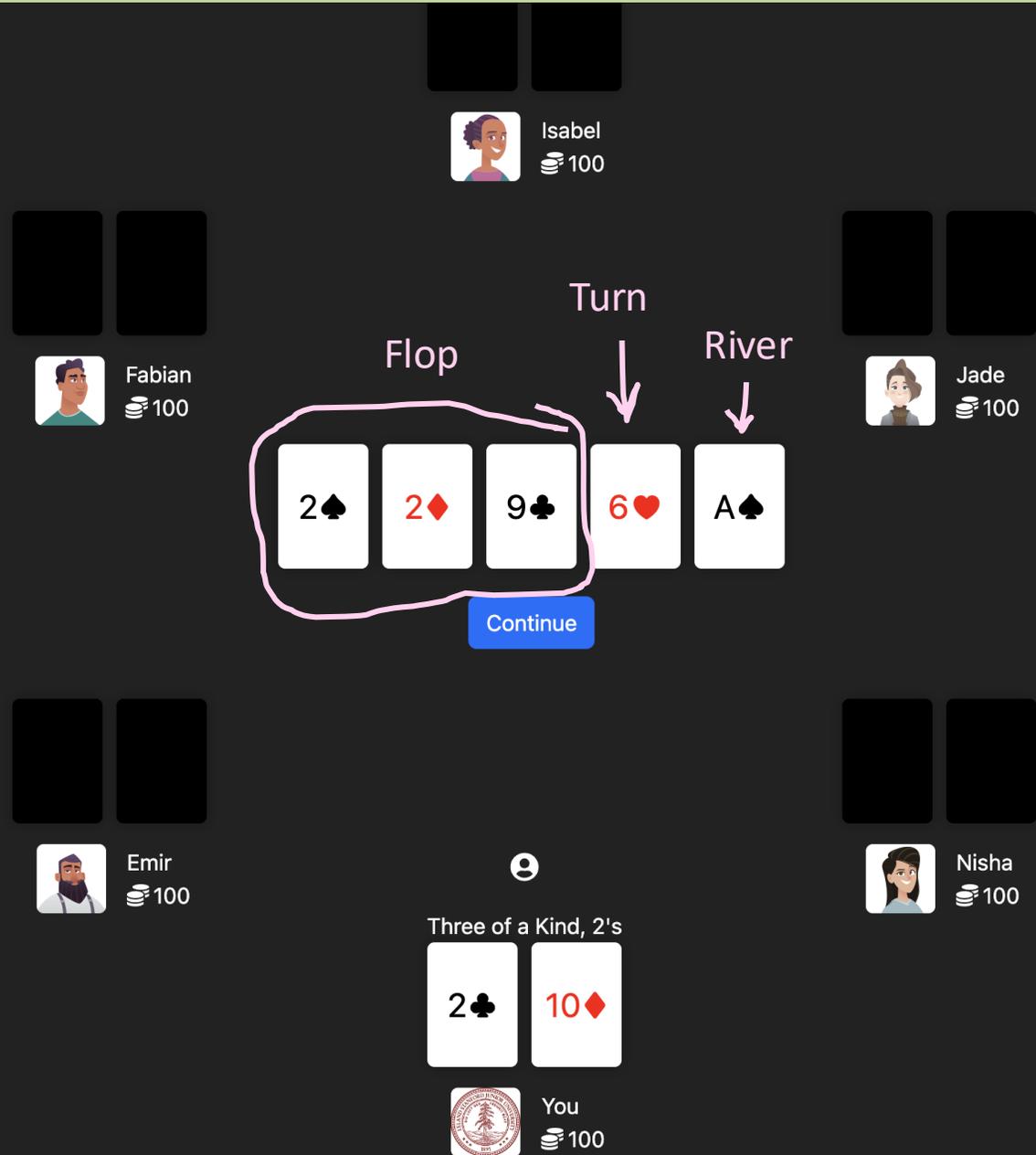
**D:**

# Magical Puppy Drugs



- Reference
  - Notation Reference
  - Core Probability Reference
  - Random Variable Reference
  - Python Reference
  - Calculus Reference
  - Calculators
  - Language Model Tool
- Part 1: Core Probability
  - Probability
    - Equally Likely Outcomes
    - Axioms of Probability
    - Probability of or
    - Conditional Probability
    - Law of Total Probability
    - Bayes' Theorem
    - Independence
    - Probability of and
    - De Morgan's Law
    - Log Probabilities
    - Many Coin Flips
    - Counting
    - Combinatorics
  - Stories
    - Bacteria Evolution
    - Google Rain Prediction
    - Random Walks
    - Binomial with Different Probs
    - Netflix Genres
    - Poker

# Poker Information Theory

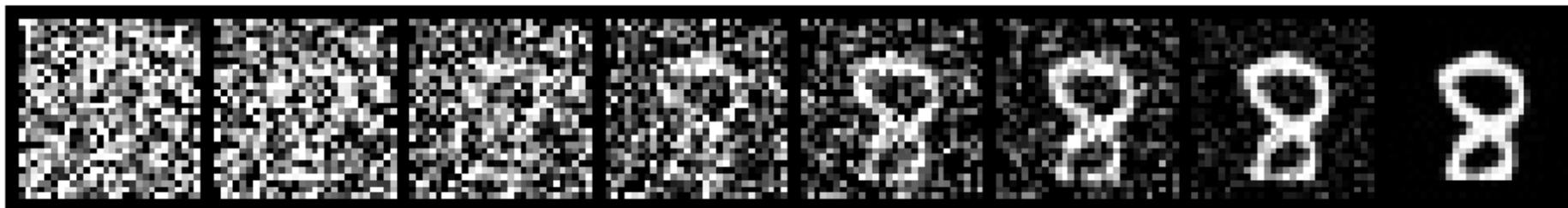


Let  $X$  be a random variable for the set of 7 cards at the end. What is  $H(X)$  after each event?

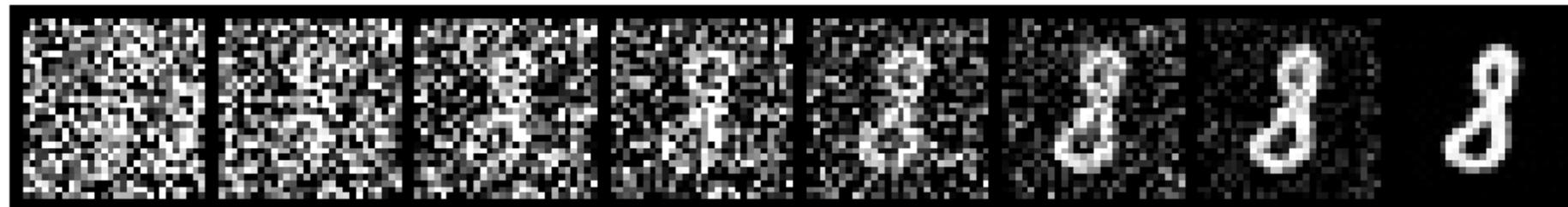
There are 52 unique cards. Each card is equally likely.

# Diffusion!

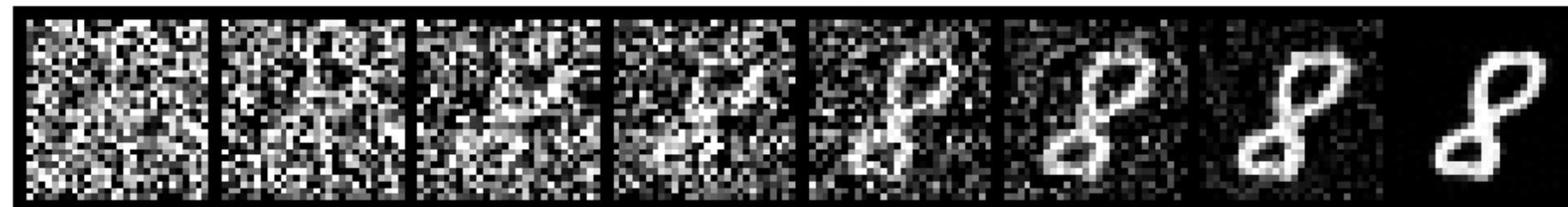
Reverse denoising (class 8)



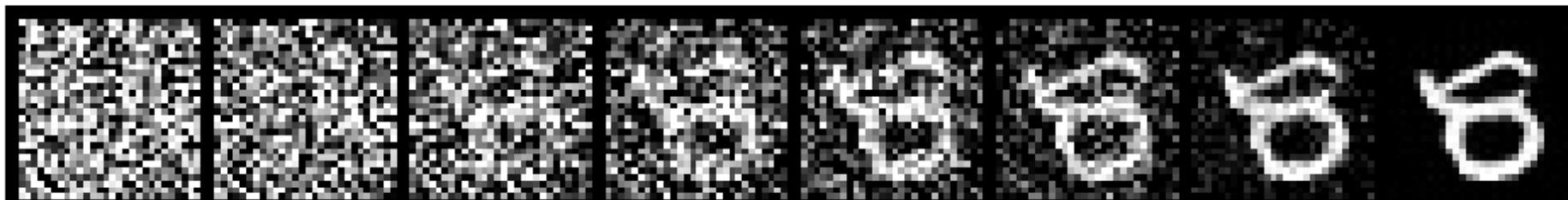
Reverse denoising (class 8)



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  $n \rightarrow \infty$

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

# 928 EdStem Questions

The screenshot shows the EdStem website interface. The top navigation bar is purple with the 'ed' logo and the course title 'CS 109 Winter 2026 - Ed Discussion'. The left sidebar contains a 'New Thread' button, a search bar, and a list of categories including 'General', 'Lectures', 'Sections', 'Problem Sets', 'Exams', 'PEP', and 'Social'. The main content area displays a discussion thread titled 'Beyond CS109 #375' by Juliette Woodrow (STAFF) posted 2 hours ago. The thread includes a heart icon, a description of the discussion, and two answers. Answer 1 discusses the teaching team's perspective on the question, and Answer 2 provides a warning about the course's difficulty. The bottom of the screenshot shows a browser status bar with the URL: <https://edstem.org/us/courses/90458/discussion/7799469>.

ed CS 109 Winter 2026 - Ed Discussion

Search

Filter

Beyond CS109  
General Juliette Woodrow STAFF 2h

Week 9 Recap  
General Juliette Woodrow STAFF 3d

Week 8 Recap  
General Juliette Woodrow STAFF 1w 1

Sign up for PEP for Final  
PEP Juliette Woodrow STAFF 1w 6

Image Format for PSET App  
General Saanvi Chawla STAFF 2w 1

Week 7 Recap  
General Juliette Woodrow STAFF 2w 1

PSet 6 Megathreads  
Problem Sets Isabel Michel STAFF 3w

Week 6 Recap  
General Juliette Woodrow STAFF 3w 1

PSet 5 Megathreads  
Problem Sets Isabel Michel STAFF 3w 1

Beyond CS109 #375

Juliette Woodrow STAFF  
2 hours ago in General

UNPIN STAR WATCHING VIEWS 7

I'll talk more about this in lecture today, but in the meantime I thought I would also share a discussion among CS109 TAs about what classes to consider after takings CS109.

### Answer 1

Ooh I love this question! We (the teaching team) were just talking about this!

A couple from me:

**CS238/AA228: Decision-Making Under Uncertainty.**

Bayes Nets are a big player here, and you'll get an intro to things like reinforcement learning (with applications into things like autonomous aviation/navigation, for example, which is why it's cross-listed with AA). Mykel's a great lecturer, the content is interesting, and the pace is approachable and chill (quizzes and a final project as opposed to exams).

**CS265: Randomized Algorithms with Probabilistic Analysis.**

A *HUGE* warning label on this rec: I would probably not recommend this to people who don't have a very strong math background (well beyond the normal SoE reqs) unless you're willing to push yourself and take an academic risk. The material is really, interesting, but the prereqs are misleading. (I thought that — because I'm into Theory and crushed 109, 161, and the SoE math reqs — I'd be prepared. I wasn't.)

That said, I survived — and if I'd waited another year or two to take this class, I think I would've gotten a lot more out of it!

### Answer 2

Open <https://edstem.org/us/courses/90458/discussion/7799469> in a new tab and focus it

# New Course Reader Chapters

New Chapters

probabilitycoders.stanford.edu

## What is a Large Language Model?

Large Language Models (LLMs) like ChatGPT, Claude, and Gemini are some of the most powerful AI tools to emerge in recent years. LLMs are called generative AI because they generate text. You might be wondering, what are they really doing under the hood? At their core, LLMs are **probability machines**. They don't "know facts" in the way a textbook does. Instead, they predict the probability of a next "token" given text seen so far. *That's it? Just probability?*

Probability is a powerful thing. Especially when the model has been trained on vast amounts of text (like the entire written) and can capture rich relationships in language. This allows it to make predictions, even though underneath it's always "just" calculating probabilities.

### Rejection Sampling Demo

This demo is based on the same Bayesian network that is at the top of this handout.

#### Computational Inference

Query: P(Flu | Fever, Tired)

Resample N=10,000

1. Sample 10,000    2. Reject    3. Count Target From Remaining

ICON MAPPING

- UG (blue)
- Fever (yellow)
- Flu (red)
- Tired (purple)

INFERENCE STATS

Total N: 10,000  
Matches Evidence: 1,131  
Target (Flu+Evid): 852

ESTIMATED PROBABILITY

# 0.753

model asks: **Given what I have already seen, what is the most probable thing to come**

are strings of text by repeatedly asking this question over and over for each **token**. Tokens are strings of text that an LLM considers. If you are curious about tokens check out the section at the

### abilities

ready seen the text "After CS109 we went to":

"After CS109 we went to"

$T_1$   $T_2$   $T_3$   $T_4$   $T_5$   $T_6$

the text into bite sized strings called tokens. In order to choose the next token, the model uses a probability distribution over all possible next tokens, given what it has seen so far. The dictionary below shows the relationship between possible next tokens, and the probability that the token will come next:

next tokens.  
responding probability

- Serendipity
- Core Probability Practice
- Part 2: Random Variables
  - Random Variables
  - Probability Mass Functions
  - Expectation
  - Variance
  - Bernoulli Distribution

probabilitycoders.stanford.edu

### INCLUSION-EXCLUSION

$E$  as the union of all processes:

$$R_1 \cup R_2 \cup \dots \cup R_n$$

terms to calculate:

# 7

(0.90)  
(0.90)  
(0.90)  
 $R_2$  (0.81)  
 $R_3$  (0.81)  
 $R_3$  (0.81)  
 $R_2 \cap R_3$  (0.73)

### METHOD 2: COMPLEMENT

We define  $E^c$  as the event that no functional path exists.

#### 1. LAW OF TOTAL PROBABILITY

$$P(E) = 1 - P(E^c)$$

#### 2. DE MORGAN'S LAW

A path fails only if all routers fail:

$$E^c = R_1^c \cap R_2^c \cap \dots \cap R_n^c$$

#### 3. APPLY INDEPENDENCE

Since routers are independent, intersections become products:

$$P(E^c) = P(R_1^c) \cdot P(R_2^c) \dots P(R_n^c)$$
$$P(E) = 1 - (1 - p)^n$$

# $1 - (1 - 0.9)^3 = 0.999000$

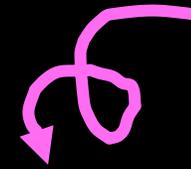
For  $n = 3$ , the complement method is **7x** faster to write out manually.

# 31+ Personal Challenges



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

My dream job !!!



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

Wild time. Incredible school at  
which to study probability



```
P> █
```

```
THANK YOU ALL SO MUCH █
```

```
[Model: CS109-GPT; Latency: 45ms; Response generated; press Ctrl+C to exit]
```

