

General Inference

CS109, Stanford University

Announcements

- (1) Midterm Exam is Tuesday 2/10 7-9pm.**
- (2) If you have an academic conflict with the exam or if you have OAE accommodations, fill out the form on Ed by end of class on Weds (form will be released very soon).
- (3) Location info will be announced later this week. We are in the AIWG proctoring pilot so we will assign you rooms and actual seats in the room as well.
- (4) You may bring 3 pieces of paper – 6 sides if you count front and back – of notes. Can be typed, handwritten, pictures, etc.
- (5) Leave phones at home if possible! If not – we will collect them before exam starts.
- (6) Review session on Friday at 4:30pm (location TBD). No lecture on Monday 2/9

Why You Need a Model

WebMD®

Why You Need a Model

WebMD Symptom Checker WITH BODY MAP

INFO **SYMPTOMS** QUESTIONS CONDITIONS DETAILS TREATMENT

What are your symptoms?

Type your main symptom here



CONTINUE

Multiple Random Variables. Start of Digital Revolution

Conditions that match your symptoms

UNDERSTANDING YOUR RESULTS 

Migraine Headache (Adult)



STRONG match



Tension Headache



Moderate match



Benign Paroxysmal Positional Vertigo (BPPV)



Fair match



Gender Female

Age 26

[Edit](#)

My Symptoms

[Edit](#)

dizziness , one sided headache



Start Over

Surprisingly Simple (if you can code)

Code



Probability

Three Guiding Questions

1. How do people actually define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

But first some review.

Did you know today is Groundhog's day?

Sees shadow = 6 more weeks of winter.

Doesn't see shadow = early spring.

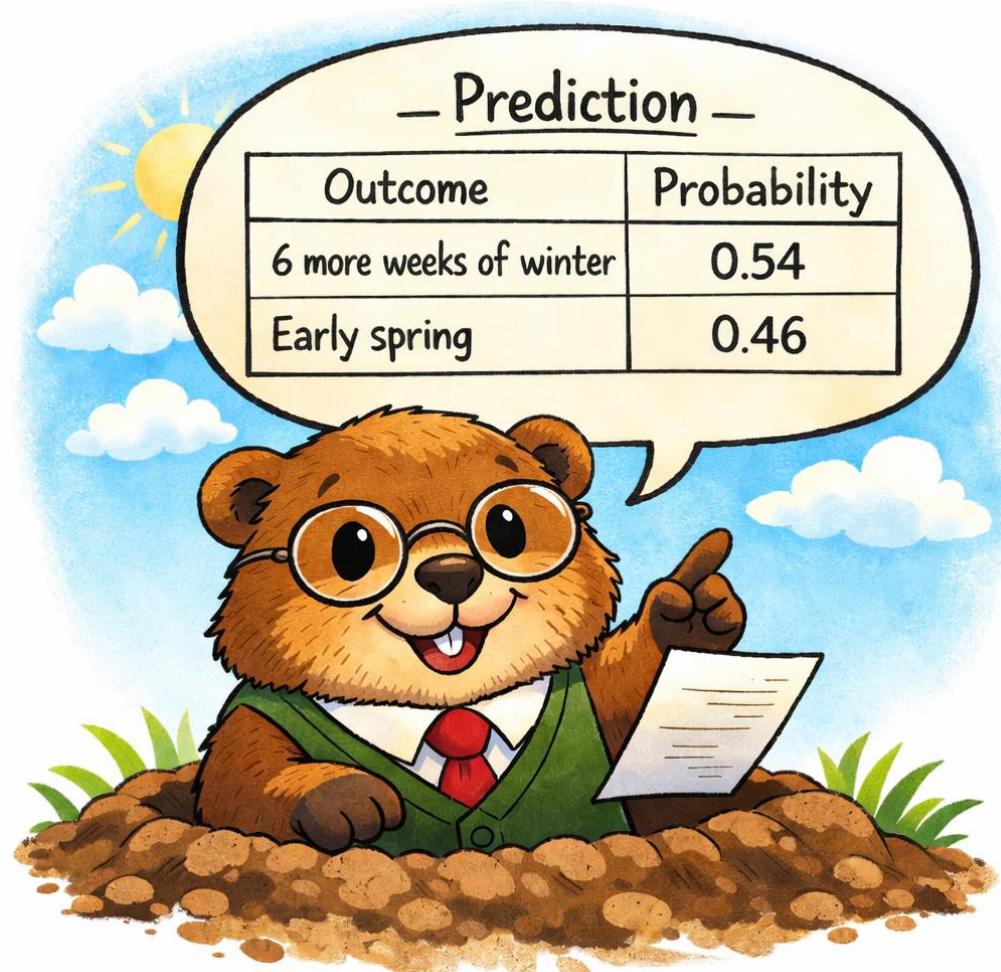
Based on historical data:

- When the groundhog sees shadow, 70% chance of 6 more weeks of winter.
- When the groundhog doesn't see shadow, 40% chance of 6 more weeks of winter.
- Before we observe the groundhog, we think it is equally likely to be an early spring or 6 more weeks of winter.



Let W = 6 more weeks of winter, and S = sees shadow.

$$P(W | S) = \frac{P(S | W)P(W)}{P(S | W)P(W) + P(S | W^c)P(W^c)}$$
$$= \frac{0.7 \cdot 0.5}{0.7 \cdot 0.5 + 0.6 \cdot 0.5} = \frac{0.35}{0.65} \approx 0.54$$



At this point you know inference with
two random variables

Today: Five New Real + Exciting Problems

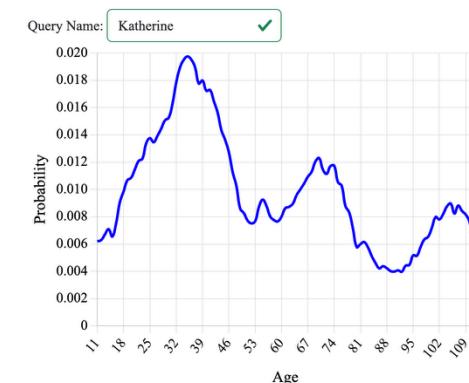
Age from C14



Updated Delivery Prob



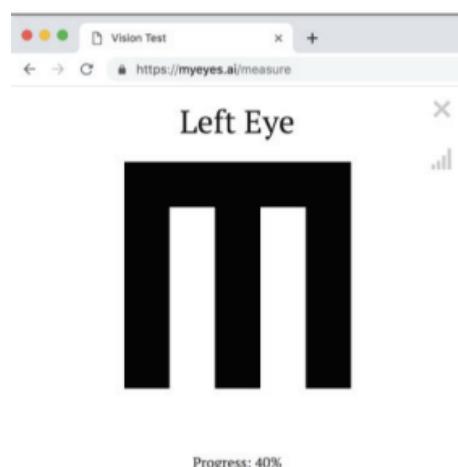
Age from Name



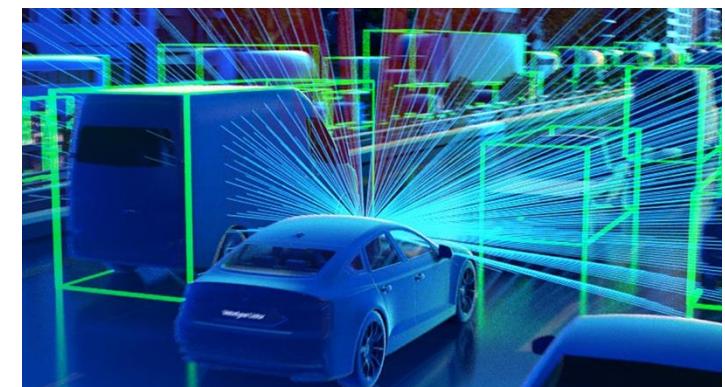
Hidden Chambers



Stanford Eye Test



Updating Lidar Belief



Today: Five New Real + Exciting Problems

Simple Joint

Interesting Likelihood

Age from C14



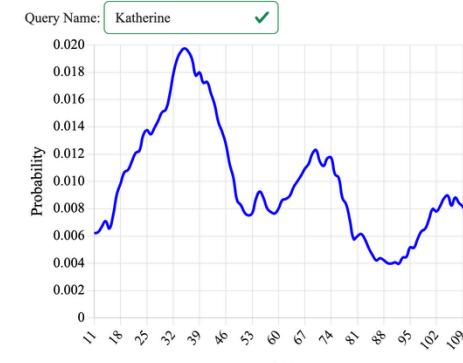
Hidden Chambers



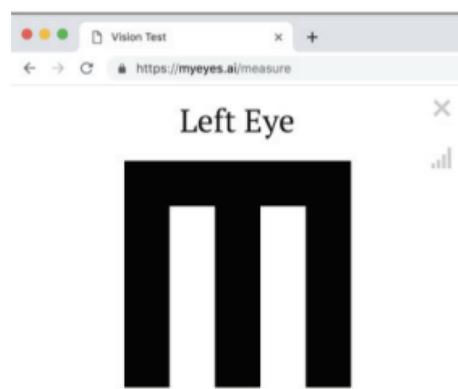
Updated Delivery Prob



Age from Name



Stanford Eye Test



Repeat Observations

Updating Lidar Belief



Continuous

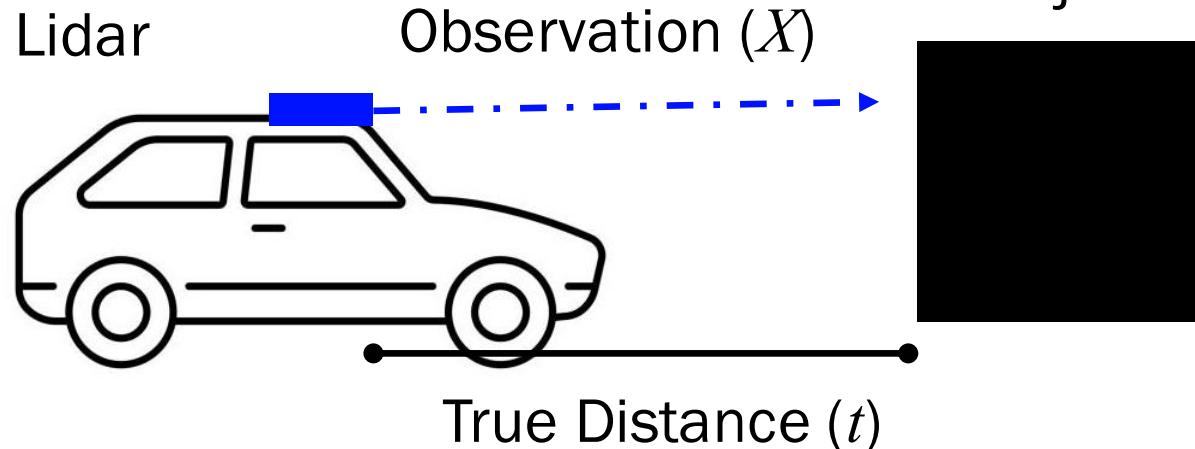
Updating Lidar Belief

Your prior belief in true distance

$$T \sim N(\mu = 1, \sigma^2 = 3)$$

Your sensor has uncertainty

$$X \sim N(\mu = t, \sigma^2 = 1.5)$$



Observe: $X = 4$

$$f(T = t | X = 4) \propto f(X = 4 | T = t) \cdot f(T = t)$$

$$\propto \frac{1}{\sqrt{2\pi \cdot 3}} \exp\left[-\frac{1}{2} \frac{(4-t)^2}{3}\right] \cdot \frac{1}{\sqrt{2\pi \cdot 1.5}} \exp\left[-\frac{1}{2} \frac{(t-1)^2}{1.5}\right]$$

Optional
but neat

$$\begin{aligned} &\propto \exp\left[-\frac{1}{2} \left(\frac{(4-t)^2}{3} + \frac{(t-1)^2}{1.5}\right)\right] \\ &\propto \exp\left[-\frac{1}{2}(t^2 - 6t)\right] \\ &\propto \exp\left[-\frac{1}{2}(t-3)^2\right] \end{aligned}$$

$$\sim N(\mu = 3, \sigma^2 = 1)$$

Bayes theorem

Plug in normal PDF

Drop constants +
Combine exponents

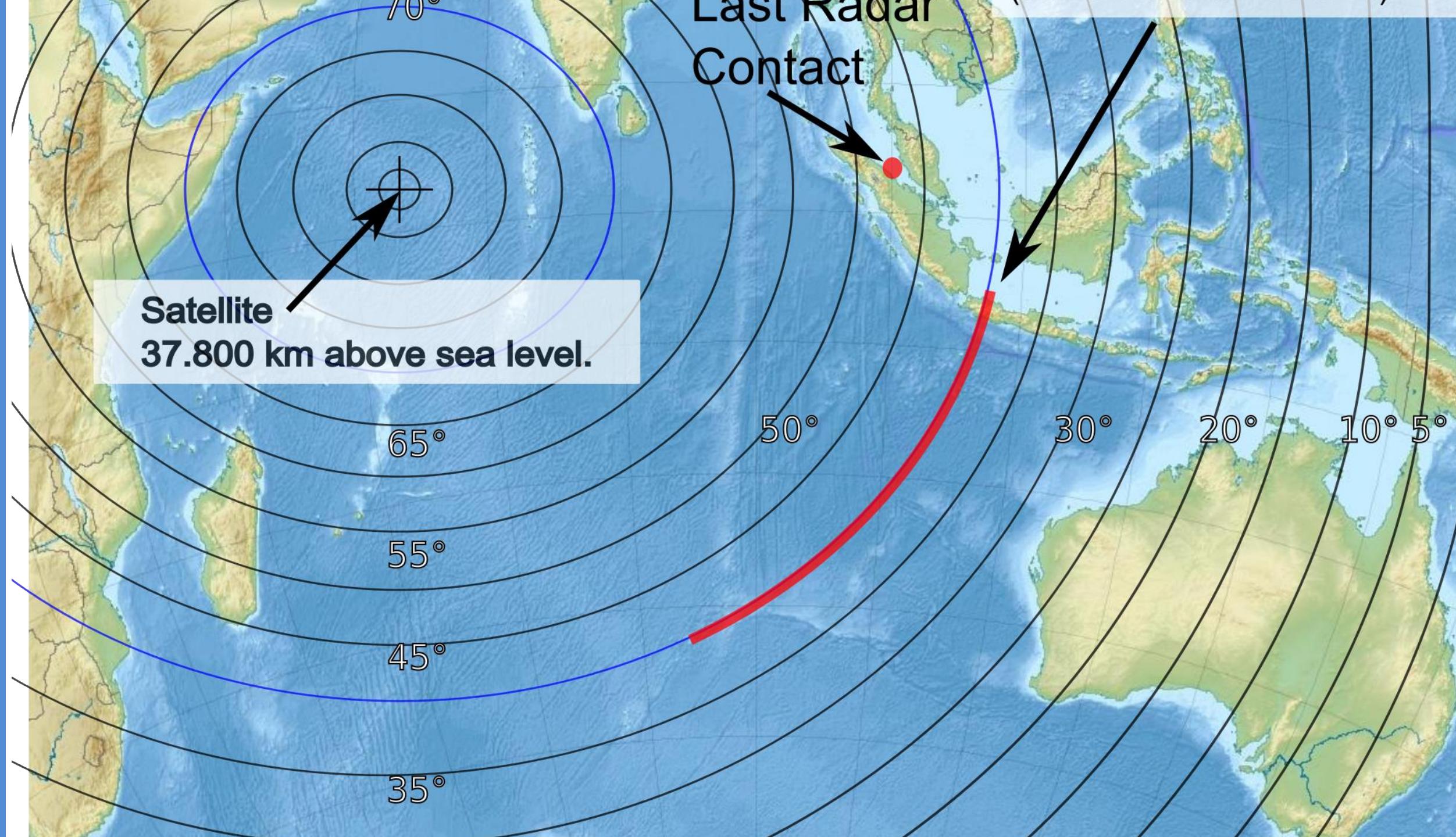
After simplifying

Complete the square

Pro tip: keep the $-1/2$ factored out

Tracking in 2D Space?







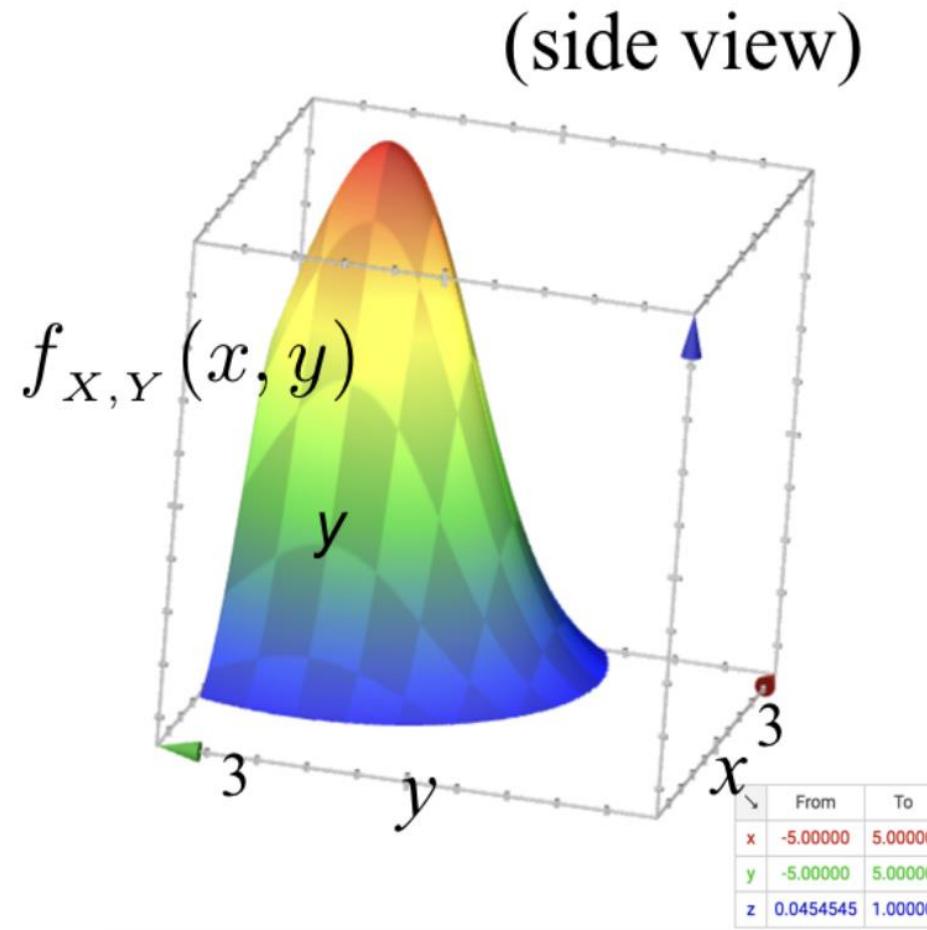
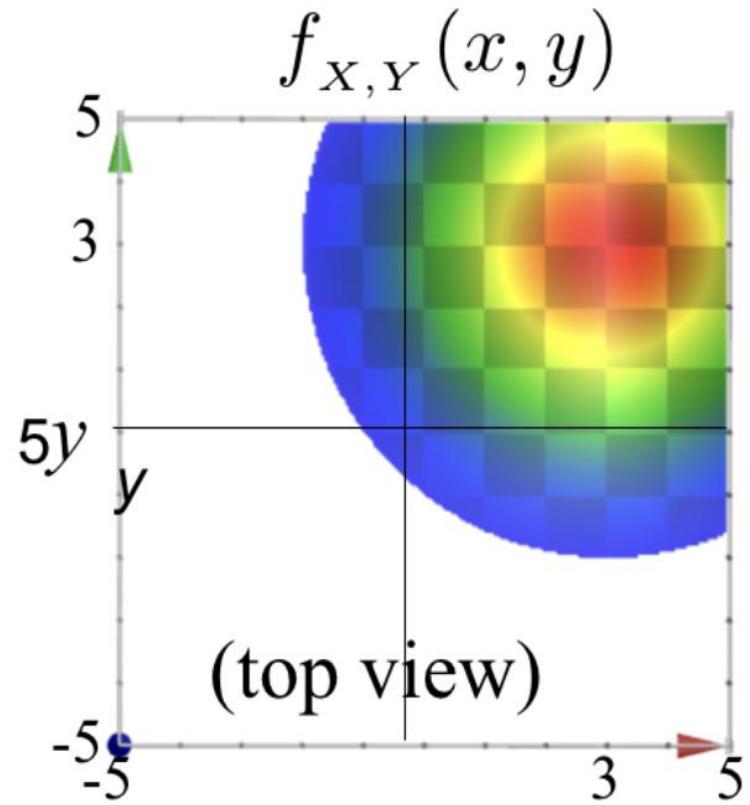
EXPERTS ONLY

Expect unmarked cliffs,
rocks, avalanche debris
and other alpine hazards.

GATE

Prior

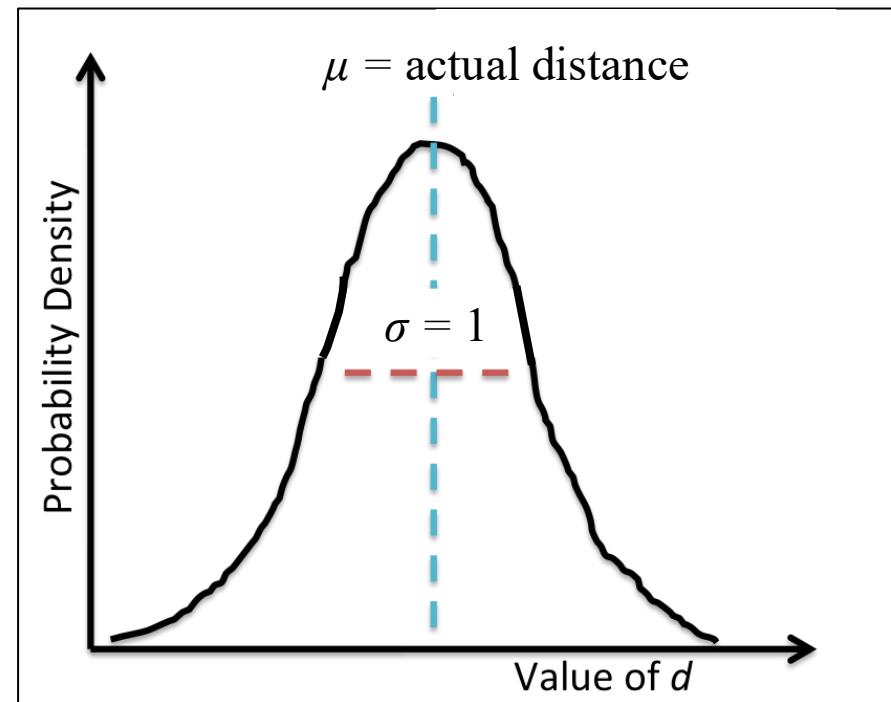
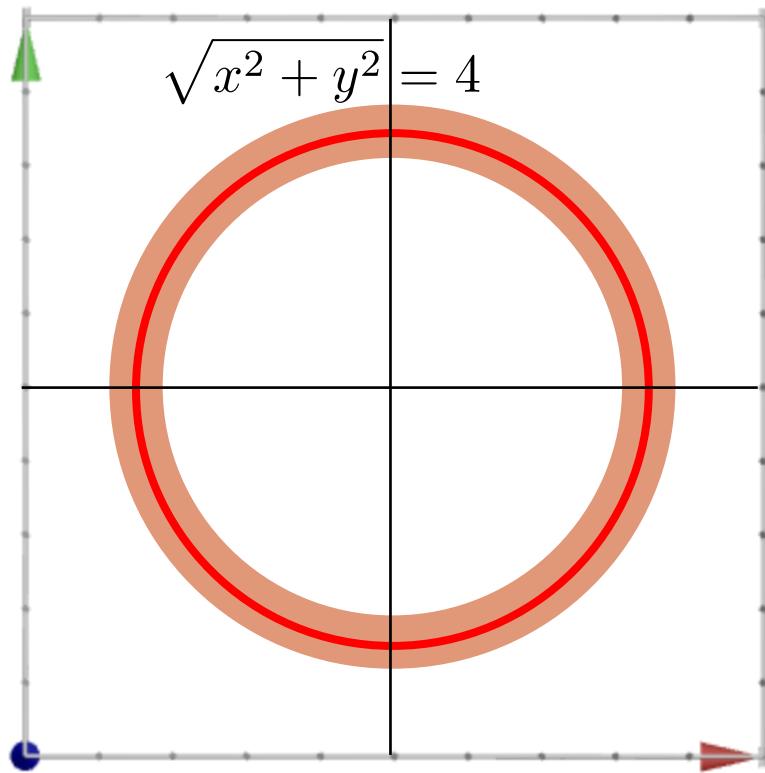
Prior belief: $f(X = x, Y = y) = \frac{1}{8 \cdot \pi} \cdot e^{-\frac{(x-3)^2 + (y-3)^2}{8}}$



Likelihood

You now observe a noisy distance reading from a sensor at (0,0).
It says that your object is distance $D = 4$ away

$$D|X, Y \sim N(\mu = \sqrt{x^2 + y^2}, \sigma^2 = 1)$$



Put it all Together

$$\begin{aligned} & f(X = x, Y = y \mid D = 4) \\ &= \frac{f(D = 4 \mid X = x, Y = y) \cdot f(X = x, Y = y)}{f(D = 4)} \\ &= \frac{K_1 \cdot e^{-\frac{(4-\sqrt{x^2+y^2})^2}{2}} \cdot K_2 \cdot e^{-\frac{(x-3)^2+(y-3)^2}{8}}}{f(D = 4)} \\ &= \frac{K_1 \cdot K_2}{f(D = 4)} \cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]} \\ &= K_3 \cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]} \end{aligned}$$

Bayes using densities

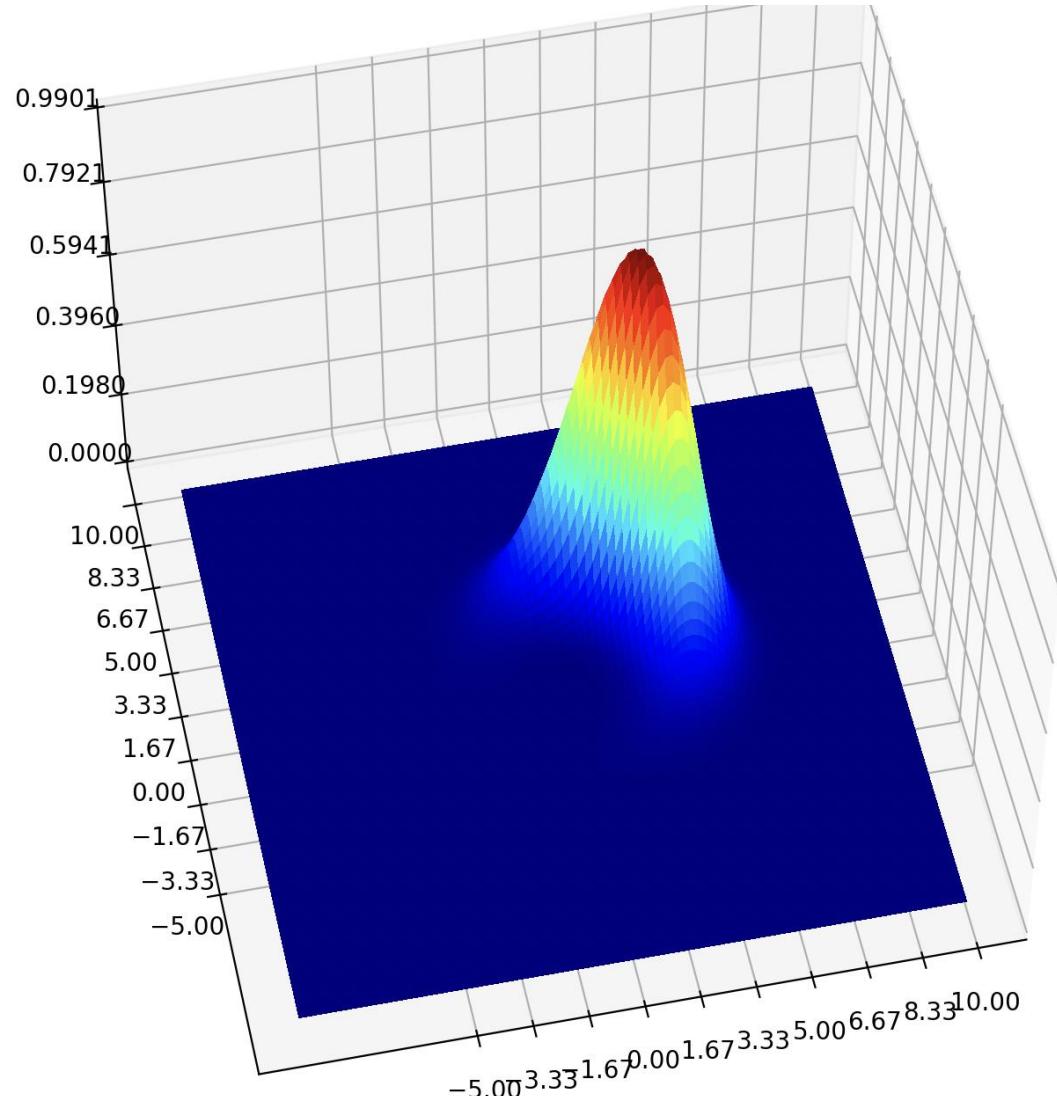
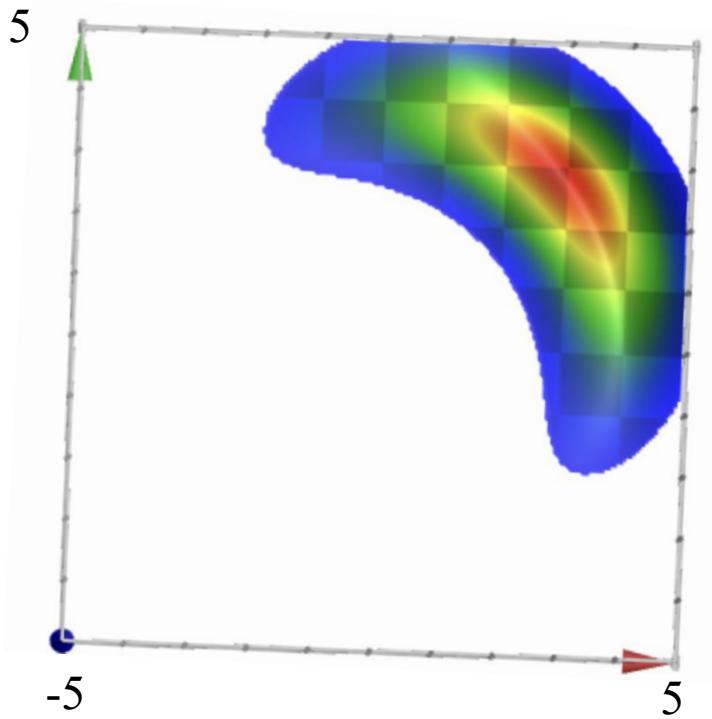
Substitute

$f(D = 4)$ is a constant w.r.t. (x, y)

K_3 is a new constant

Tracking Posterior

$$f(X = x, Y = y \mid D = 4)$$



Many real world problems have way
more than two random variables...

Why You Need a Model

WebMD®

Why You Need a Model

WebMD Symptom Checker WITH BODY MAP

INFO **SYMPTOMS** QUESTIONS CONDITIONS DETAILS TREATMENT

What are your symptoms?

Type your main symptom here



CONTINUE

Multiple Random Variables. Start of Digital Revolution

Conditions that match your symptoms

UNDERSTANDING YOUR RESULTS 

Migraine headache (adult)

 Moderate match



Acute Sinusitis

 Fair match



Stroke

 Fair match



Gender **Male**

Age **30**

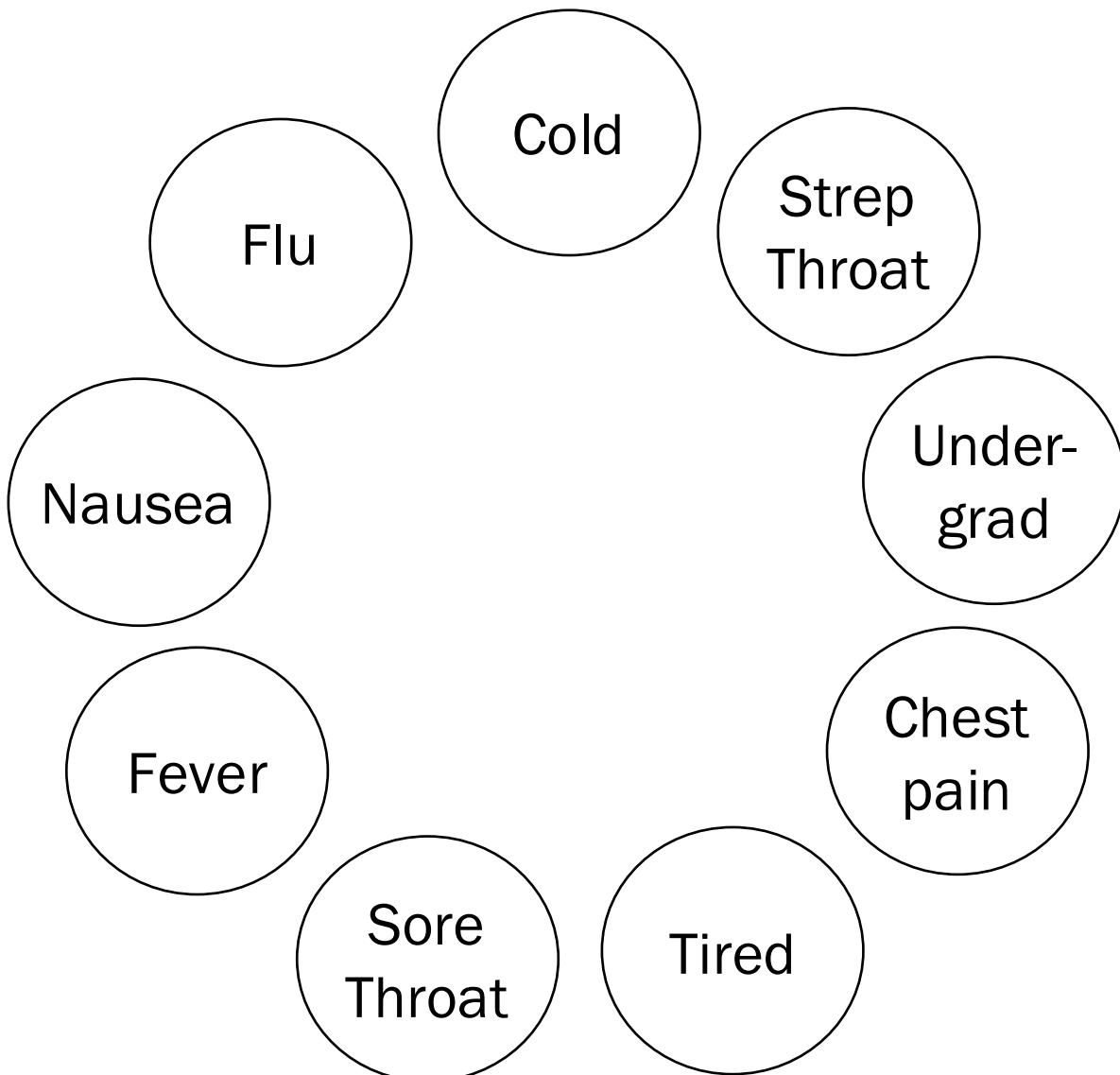
[Edit](#)

My Symptoms

[Edit](#)

dizziness, one sided headache

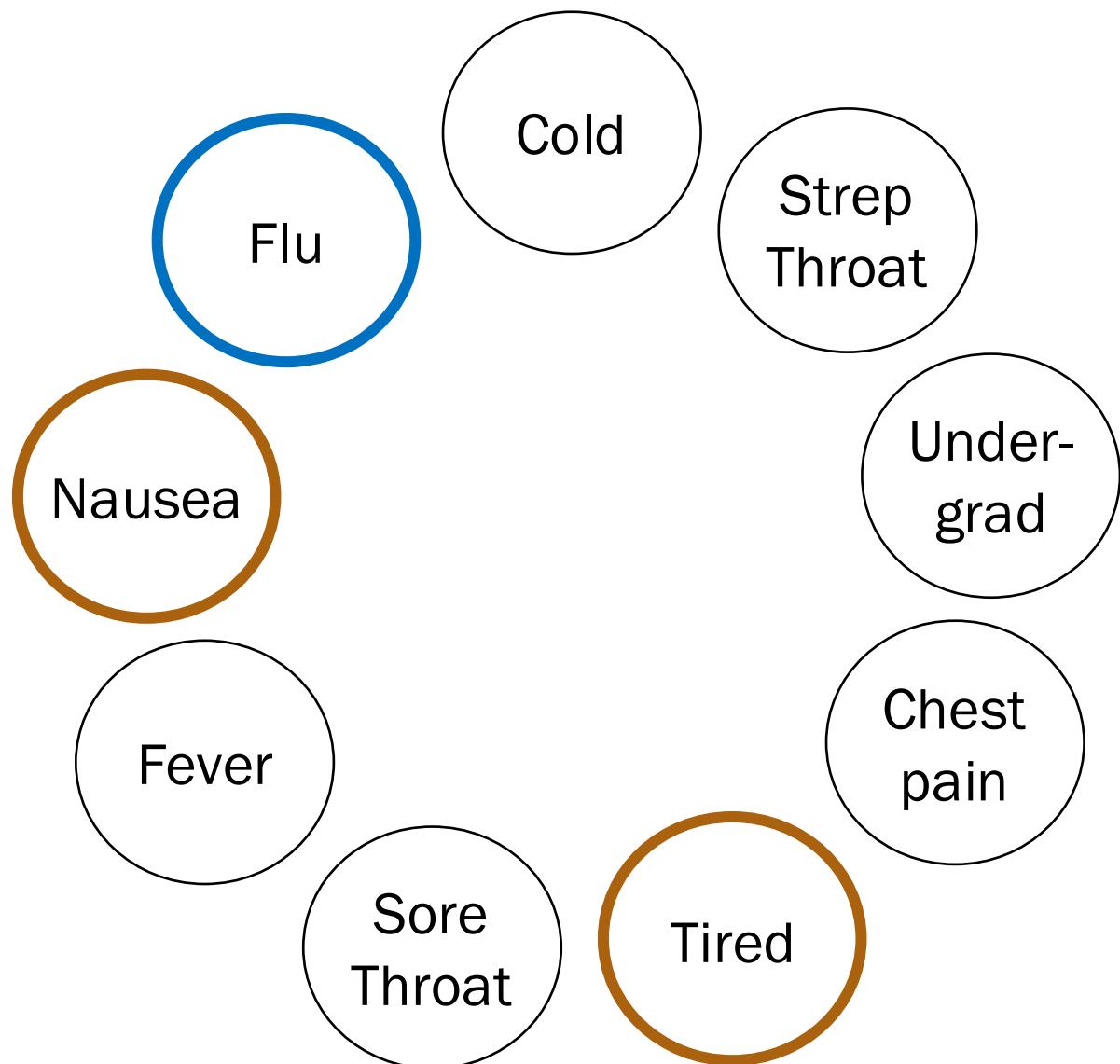
Challenge #1: Many Inference Questions



Inference question:

Given the values of some random variables, what are the conditional distributions of some other random variables?

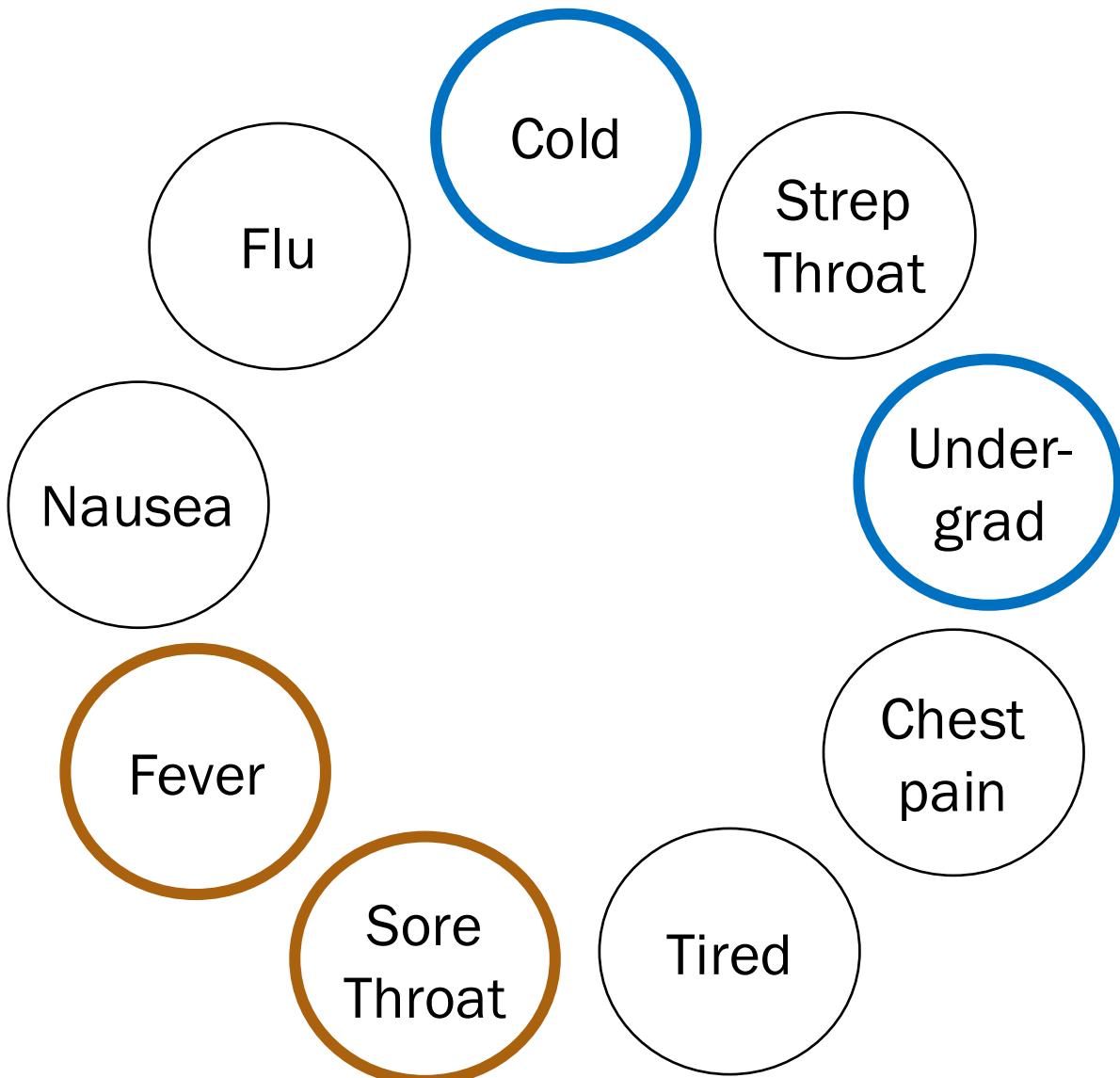
Challenge #1: Many Inference Questions



One inference question:

$$P(F = 1 | N = 1, T = 1)$$

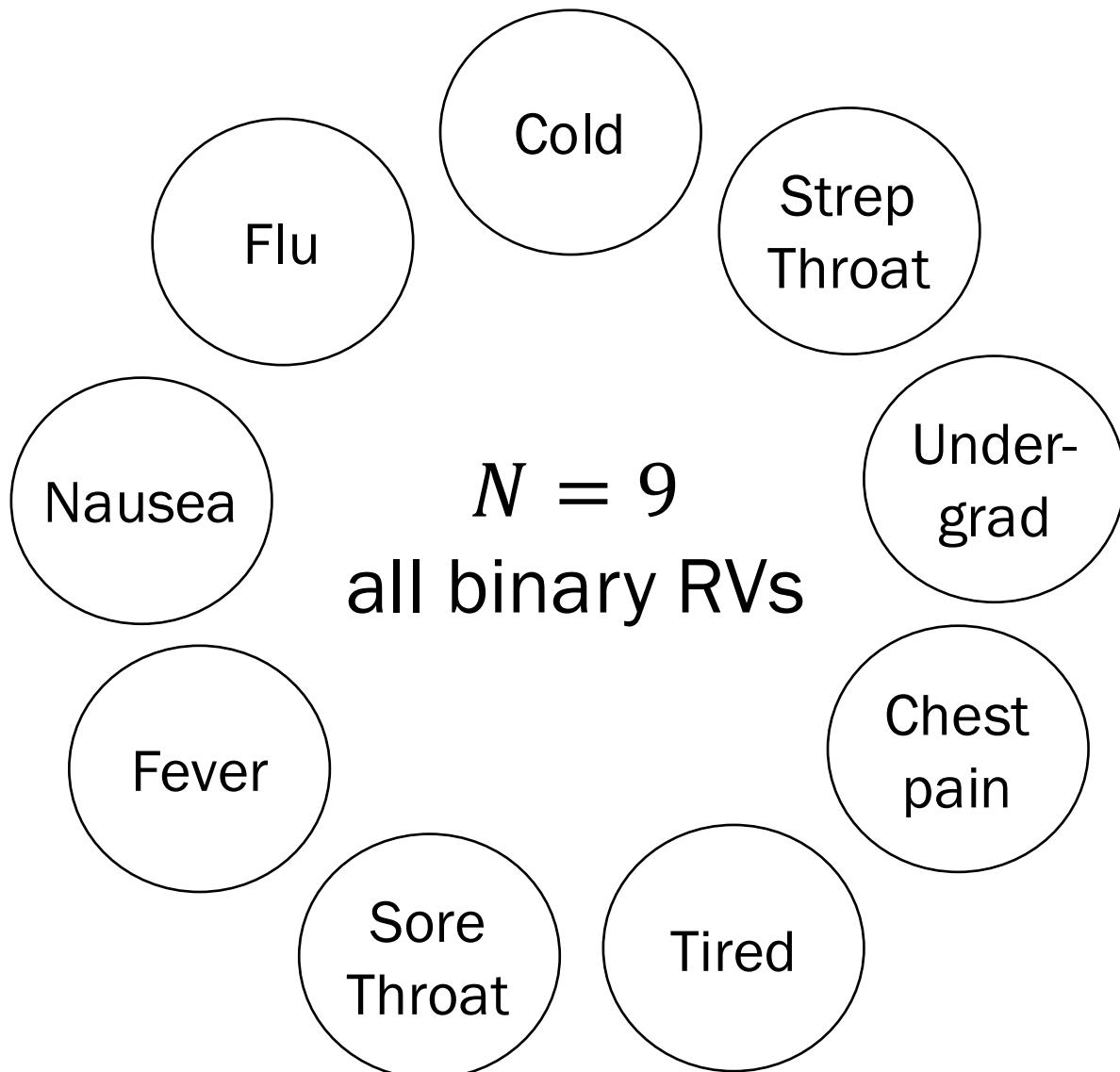
Challenge #1: Many Inference Questions



Another inference question:

$$P(C_o = 1, U = 1 | S = 0, F_e = 0)$$

Challenge #2: Joint is Large



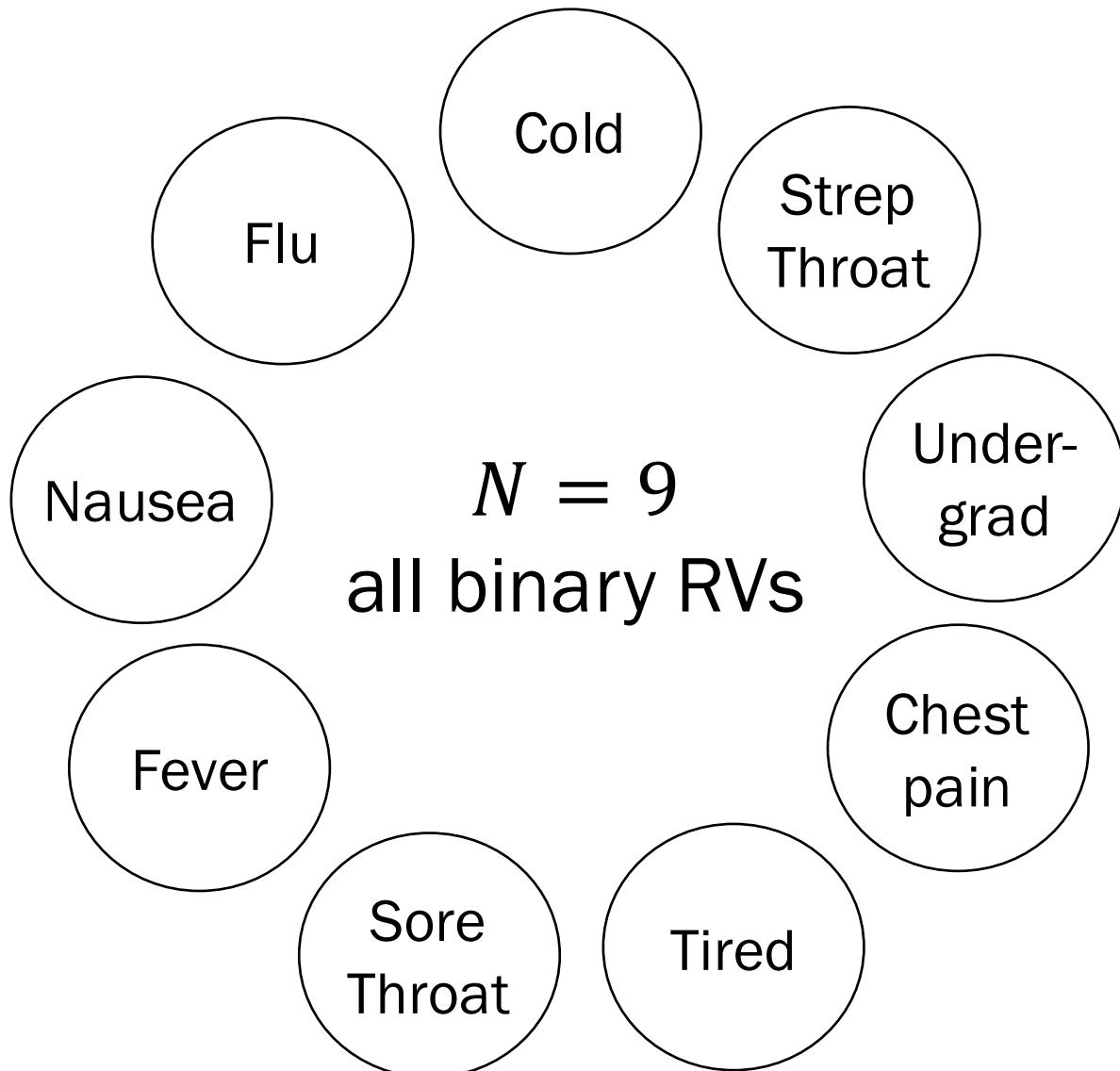
If we knew the **joint distribution**, we can answer all probabilistic inference questions.

What is the size of the joint probability table?

- A. 2^{N-1} entries
- B. N^2 entries
- C. 2^N entries
- D. None/other/don't know



Challenge #2: Joint is Large



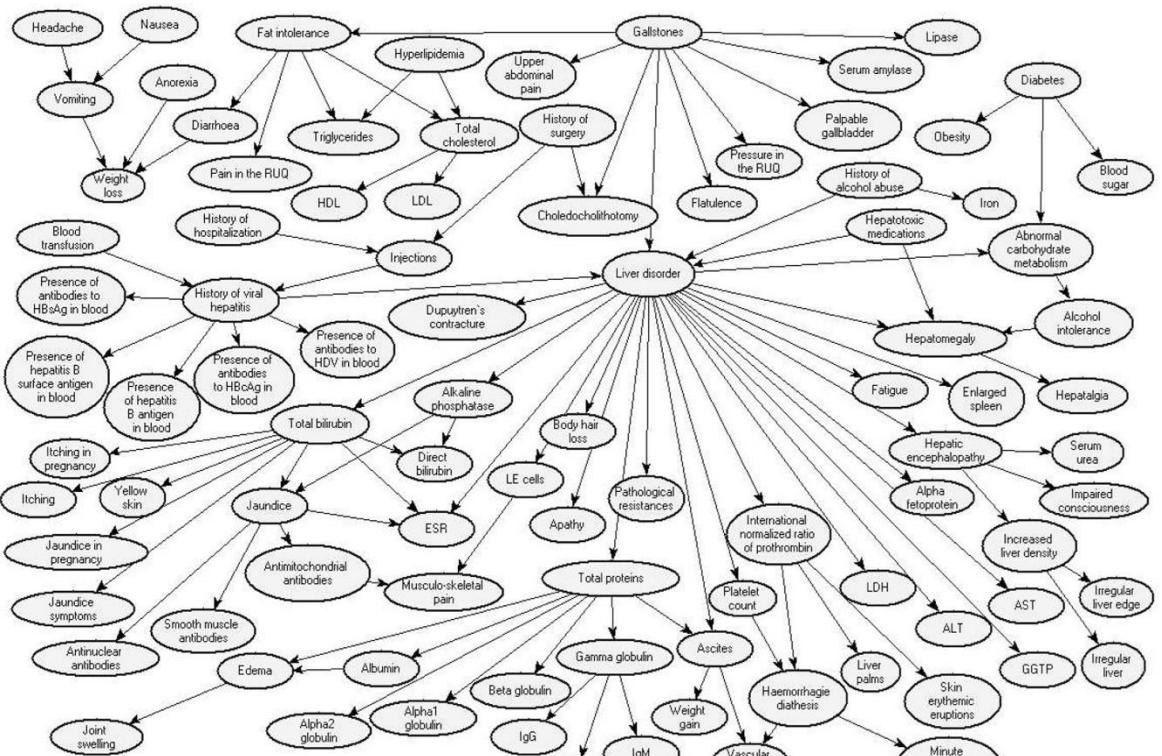
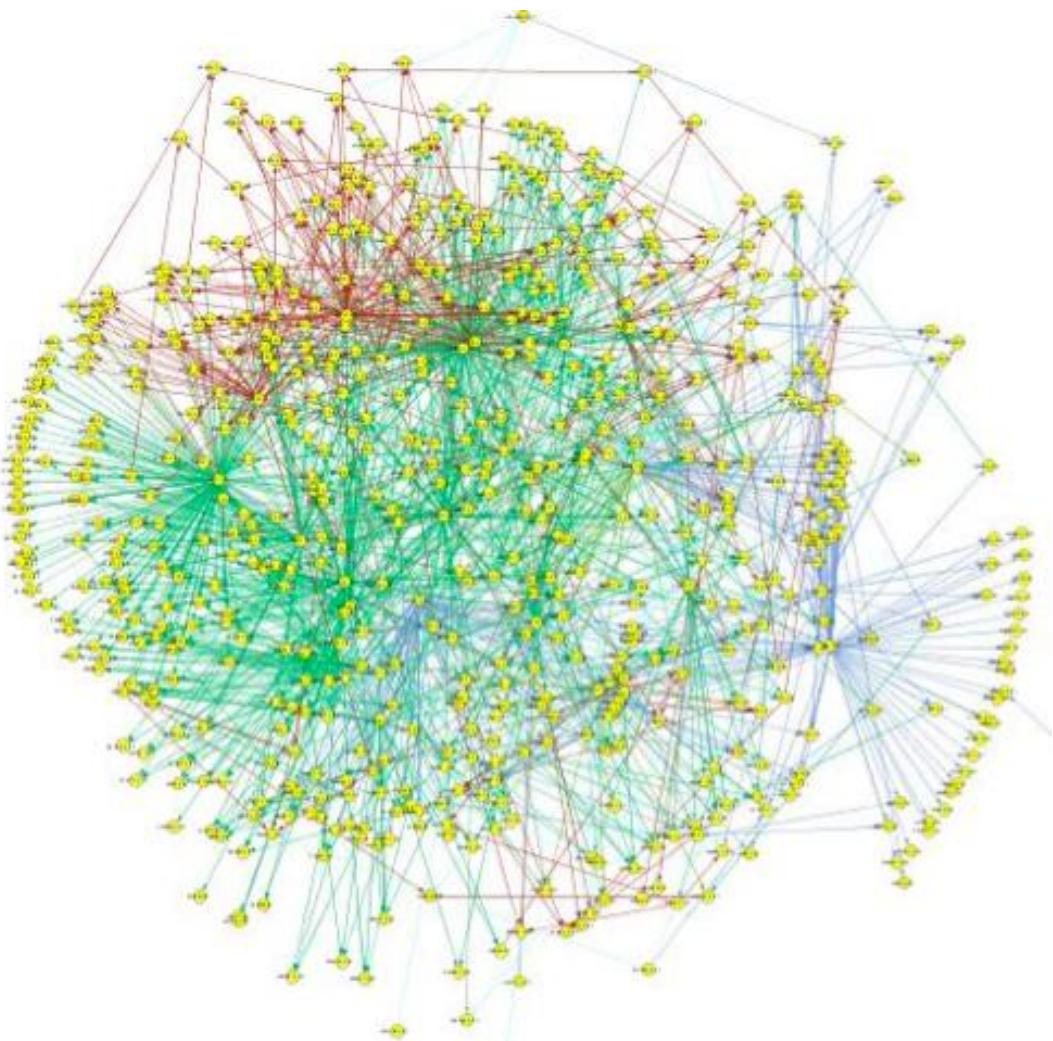
If we knew the **joint distribution**, we can answer all probabilistic inference questions.

What is the size of the joint probability table?

- A. 2^{N-1} entries
- B. N^2 entries
- C. 2^N entries
- D. None/other/don't know

Naively specifying a joint distribution is, in general, intractable.

N can be large...



Three Guiding Questions

1. How do people actually define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

Three Guiding Questions

1. How do people actually define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

Why You Need a Model

WebMD®

A simpler WebMD

Flu

Under-
grad

Fever

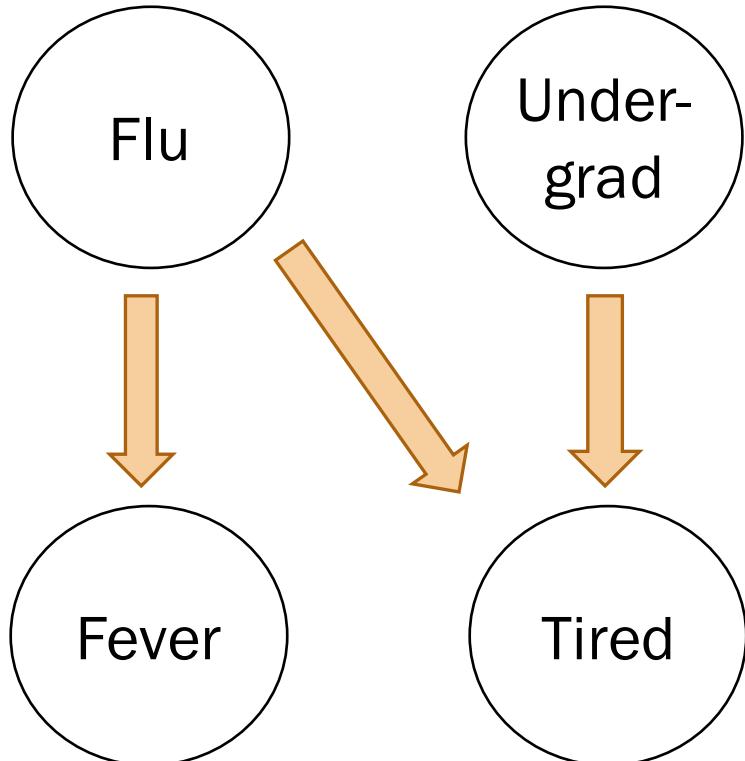
Tired

Great! Just specify $2^4 = 16$ joint probabilities...?

$$P(F_{lu} = a, F_{ev} = b, U = c, T = d)$$

We can compress the joint if we know the generative story...

Constructing a Bayesian Network

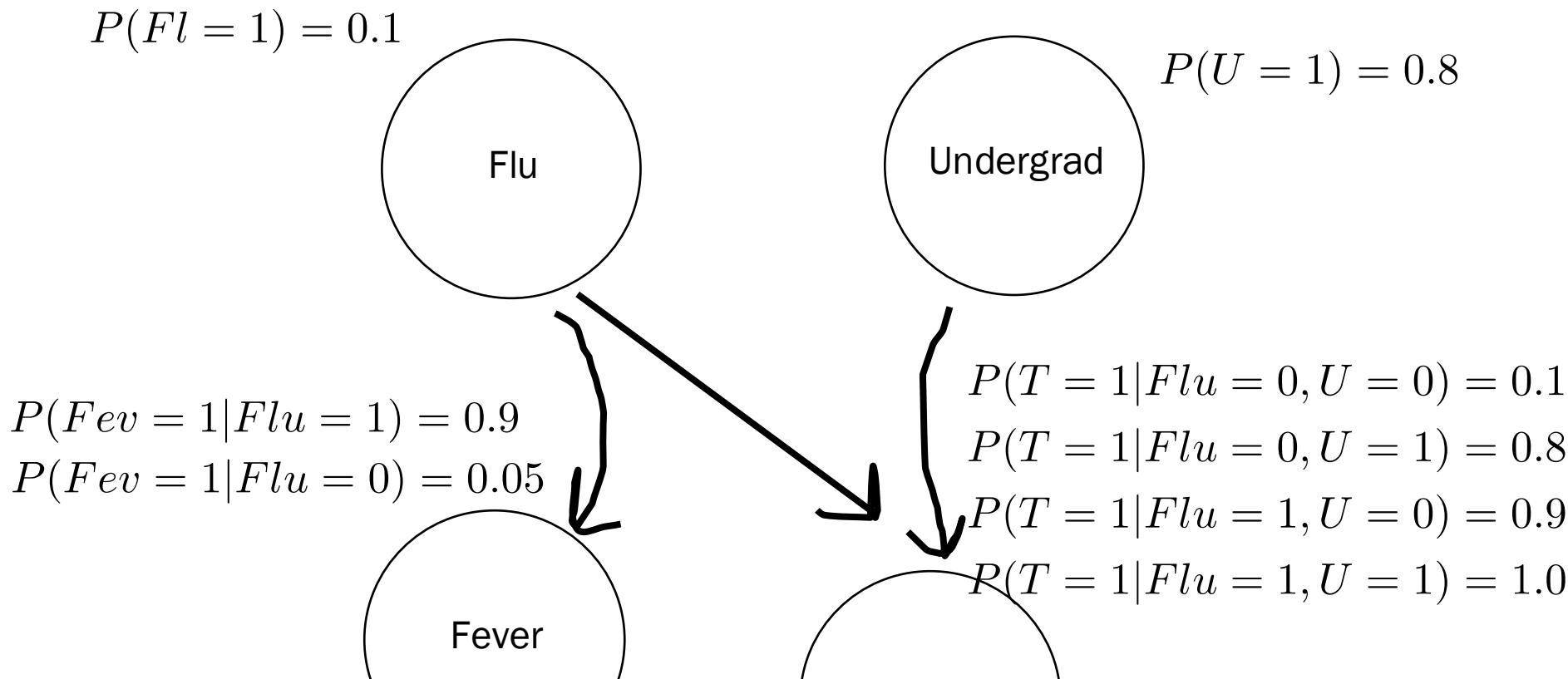


What would a Stanford flu expert do?

- 1. Describe the causality.
- 2. Provide $P(\text{values}|\text{causal parents})$ for each random variable
- 3. Implicitly assumes independences.

Recall: Probabilistic Model

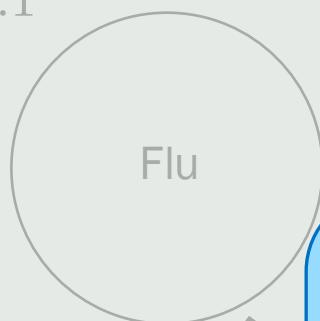
2. Provide $P(\text{values}|\text{causal parents})$ for each random variable



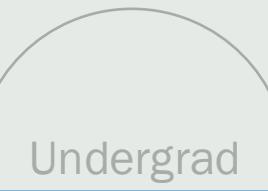
Recall: Probabilistic Model

 2. Provide $P(\text{values}|\text{causal parents})$ for each random variable

$$P(Fl = 1) = 0.1$$

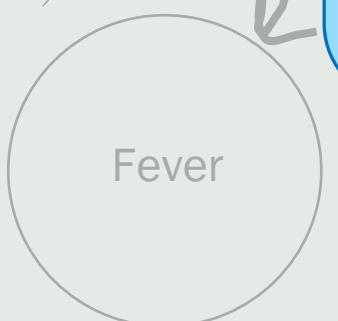


$$P(U = 1) = 0.8$$



$$P(Fev = 1|Flu = 1) = 0.9$$

$$P(Fev = 1|Flu = 0) = 0.05$$



Check your understanding:

What is $P(\text{Fev} = 0|\text{Flu} = 1)$



Could we write a python program which makes a fake person from this joint?

To the Code



Midjourney 2023. Prompt: “a lot of excited pixar characters running off to computers”



ChatGPT 5.2 2026. Prompt: “a lot of cute animated characters running off to computers to solve a problem”

```
3  def make_sample():
4      """
5      Make Sample
6      -----
7      choose a single sample from the joint distribution
8      """
9      # prior on causal factors
10     flu = bern(0.1)
11     undergrad = bern(0.8)
12
13     # choose fever based on flue
14     if flu == 1: fever = bern(0.9)
15     else:         fever = bern(0.05)
16
17     # choose tired based on (undergrade and flu)
18     if undergrad == 1 and flu == 1: tired = bern(1.0)
19     elif undergrad == 1 and flu == 0: tired = bern(0.8)
20     elif undergrad == 0 and flu == 1: tired = bern(0.9)
21     else:                         tired = bern(0.1)
22
23     # a sample from the joint has an
24     # assignment to *all* random variables
25     return {
26         'flu':flu,
27         'undergrad':undergrad,
28         'fever':fever,
29         'tired':tired
30     }
```

```
3  def make_sample():
4      """
5      Make Sample
6      -----
7      choose a single sample from the joint distribution
8      """
9      # prior on causal factors
10     flu = bern(0.1)
11     undergrad = bern(0.8)
12
13     # choose fever based on flu
14     if flu == 1: fever = bern(0.9)
15     else:         fever = bern(0.05)
16
17     # choose tired based on (undergrad and flu)
18     if undergrad == 1 and flu == 1: tired = bern(1.0)
19     elif undergrad == 1 and flu == 0: tired = bern(0.8)
20     elif undergrad == 0 and flu == 1: tired = bern(0.9)
21     else:                         tired = bern(0.1)
22
23     # a sample from the joint has an
24     # assignment to *all* random variables
25     return {
26         'flu':flu,
27         'undergrad':undergrad,
28         'fever':fever,
29         'tired':tired
30     }
```

```
3  def make_sample():
4      """
5      Make Sample
6      -----
7      choose a single sample from the joint distribution
8      """
9      # prior on causal factors
10     flu = bern(0.1)
11     undergrad = bern(0.8)
12
13     # choose fever based on flu
14     if flu == 1: fever = bern(0.9)
15     else:         fever = bern(0.05)
16
17     # choose tired based on (undergrad and flu)
18     if undergrad == 1 and flu == 1: tired = bern(1.0)
19     elif undergrad == 1 and flu == 0: tired = bern(0.8)
20     elif undergrad == 0 and flu == 1: tired = bern(0.9)
21     else:                         tired = bern(0.1)
22
23     # a sample from the joint has an
24     # assignment to *all* random variables
25     return {
26         'flu':flu,
27         'undergrad':undergrad,
28         'fever':fever,
29         'tired':tired
30     }
```

```
3  def make_sample():
4      """
5      Make Sample
6      -----
7      choose a single sample from the joint distribution
8      """
9      # prior on causal factors
10     flu = bern(0.1)
11     undergrad = bern(0.8)
12
13     # choose fever based on flu
14     if flu == 1: fever = bern(0.9)
15     else:         fever = bern(0.05)
16
17     # choose tired based on (undergrad and flu)
18     if undergrad == 1 and flu == 1: tired = bern(1.0)
19     elif undergrad == 1 and flu == 0: tired = bern(0.8)
20     elif undergrad == 0 and flu == 1: tired = bern(0.9)
21     else:                         tired = bern(0.1)
22
23     # a sample from the joint has an
24     # assignment to *all* random variables
25     return {
26         'flu':flu,
27         'undergrad':undergrad,
28         'fever':fever,
29         'tired':tired
30     }
```

```
3  def make_sample():
4      """
5      Make Sample
6      -----
7      choose a single sample from the joint distribution
8      """
9      # prior on causal factors
10     flu = bern(0.1)
11     undergrad = bern(0.8)
12
13     # choose fever based on flu
14     if flu == 1: fever = bern(0.9)
15     else:         fever = bern(0.05)
16
17     # choose tired based on (undergrade and flu)
18     if undergrad == 1 and flu == 1: tired = bern(1.0)
19     elif undergrad == 1 and flu == 0: tired = bern(0.8)
20     elif undergrad == 0 and flu == 1: tired = bern(0.9)
21     else:                         tired = bern(0.1)
22
23     # a sample from the joint has an
24     # assignment to *all* random variables
25     return {
26         'flu':flu,
27         'undergrad':undergrad,
28         'fever':fever,
29         'tired':tired
30     }
```

Can You Sample from the Joint?



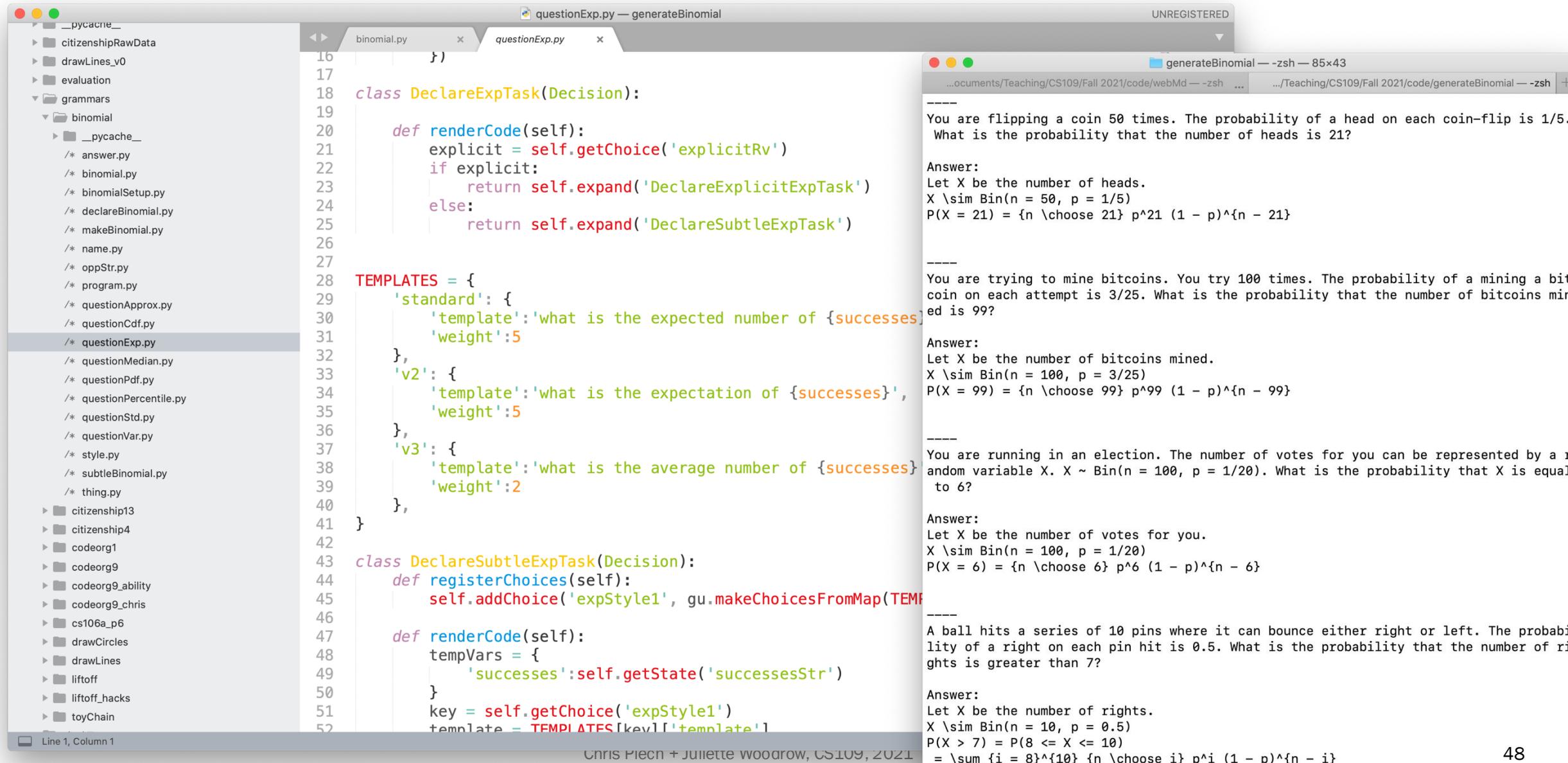
Writing a python program that can **sample** from the joint, is the same as defining the joint.

Make a *Generative* Model



A good probabilistic model is **generative**. It explains the process through which the joint is **created**.

Generative Model of Binomial Questions



The image shows a Mac OS X desktop with three windows:

- File Browser:** Shows a directory structure for a project named "generateBinomial". The "questionExp.py" file is selected.
- Code Editor:** Displays the content of "questionExp.py". The code defines two classes: `DeclareExpTask` and `DeclareSubtleExpTask`. The `DeclareExpTask` class has a `renderCode` method that returns either a template for an explicit RV or a subtle one based on a choice. The `TEMPLATES` dictionary contains three entries: `'standard'`, `'v2'`, and `'v3'`, each with a template string and a weight. The `DeclareSubtleExpTask` class has a `registerChoices` method and a `renderCode` method that uses a template and state variables.
- Terminal:** Shows a terminal session with three prompts (zsh) and their corresponding answers. Each prompt is a binomial probability problem with a solution provided below it.

```
questionExp.py — generateBinomial
UNREGISTERED
questionExp.py — generateBinomial — zsh — 85x43
.../Teaching/CS109/Fall 2021/code/webMd — zsh ... .../Teaching/CS109/Fall 2021/code/generateBinomial — zsh +

_____
You are flipping a coin 50 times. The probability of a head on each coin-flip is 1/5. What is the probability that the number of heads is 21?

Answer:
Let X be the number of heads.
 $X \sim \text{Bin}(n = 50, p = 1/5)$ 
 $P(X = 21) = \{n \choose 21\} p^{21} (1 - p)^{50 - 21}$ 

_____
You are trying to mine bitcoins. You try 100 times. The probability of a mining a bit coin on each attempt is 3/25. What is the probability that the number of bitcoins mined is 99?

Answer:
Let X be the number of bitcoins mined.
 $X \sim \text{Bin}(n = 100, p = 3/25)$ 
 $P(X = 99) = \{n \choose 99\} p^{99} (1 - p)^{100 - 99}$ 

_____
You are running in an election. The number of votes for you can be represented by a random variable X.  $X \sim \text{Bin}(n = 100, p = 1/20)$ . What is the probability that X is equal to 6?

Answer:
Let X be the number of votes for you.
 $X \sim \text{Bin}(n = 100, p = 1/20)$ 
 $P(X = 6) = \{n \choose 6\} p^6 (1 - p)^{100 - 6}$ 

_____
A ball hits a series of 10 pins where it can bounce either right or left. The probability of a right on each pin hit is 0.5. What is the probability that the number of rights is greater than 7?

Answer:
Let X be the number of rights.
 $X \sim \text{Bin}(n = 10, p = 0.5)$ 
 $P(X > 7) = P(8 \leq X \leq 10)$ 
 $= \sum_{i=8}^{10} \{n \choose i\} p^i (1 - p)^{10 - i}$ 
```

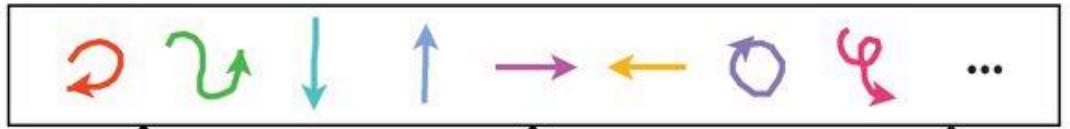
Line 1, Column 1

CHRIS PIECH + JULIETTE WOODROW, CS109, 2021

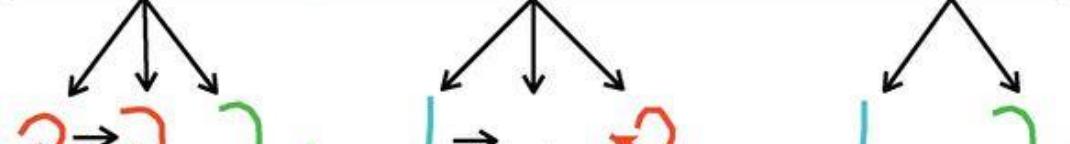
Generative Model of Hand Written Letters

A

i) primitives



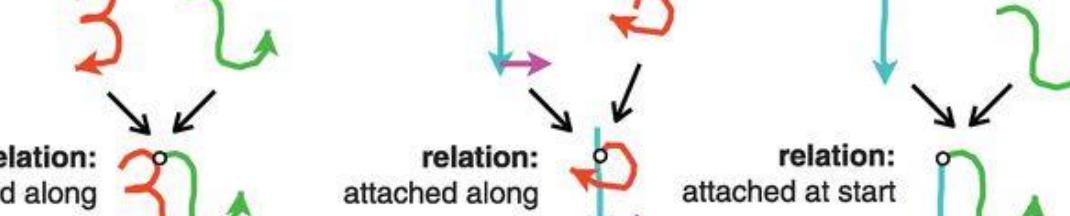
ii) sub-parts



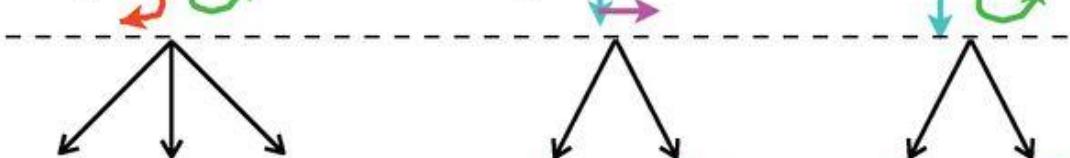
iii) parts



iv) object template
type level



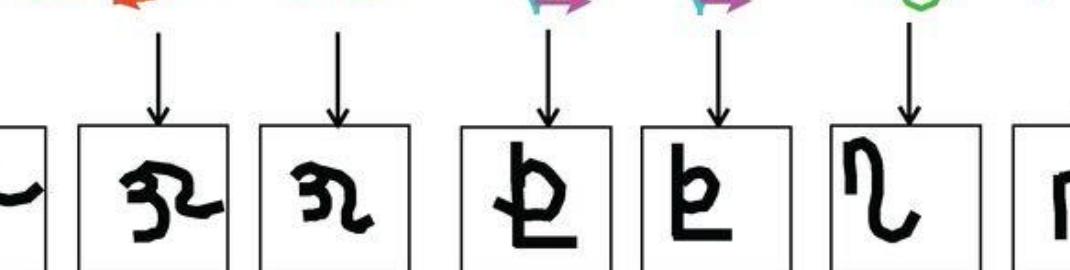
token level



v) exemplars



vi) raw data



B

procedure GENERATETYPE

```

 $\kappa \leftarrow P(\kappa)$                                 ▷ Sample number of parts
for  $i = 1 \dots \kappa$  do
     $n_i \leftarrow P(n_i|\kappa)$                           ▷ Sample number of sub-parts
    for  $j = 1 \dots n_i$  do
         $s_{ij} \leftarrow P(s_{ij}|s_{i(j-1)})$  ▷ Sample sub-part sequence
    end for
     $R_i \leftarrow P(R_i|S_1, \dots, S_{i-1})$            ▷ Sample relation
end for
 $\psi \leftarrow \{\kappa, R, S\}$ 
return @GENERATETOKEN( $\psi$ )                      ▷ Return program

```

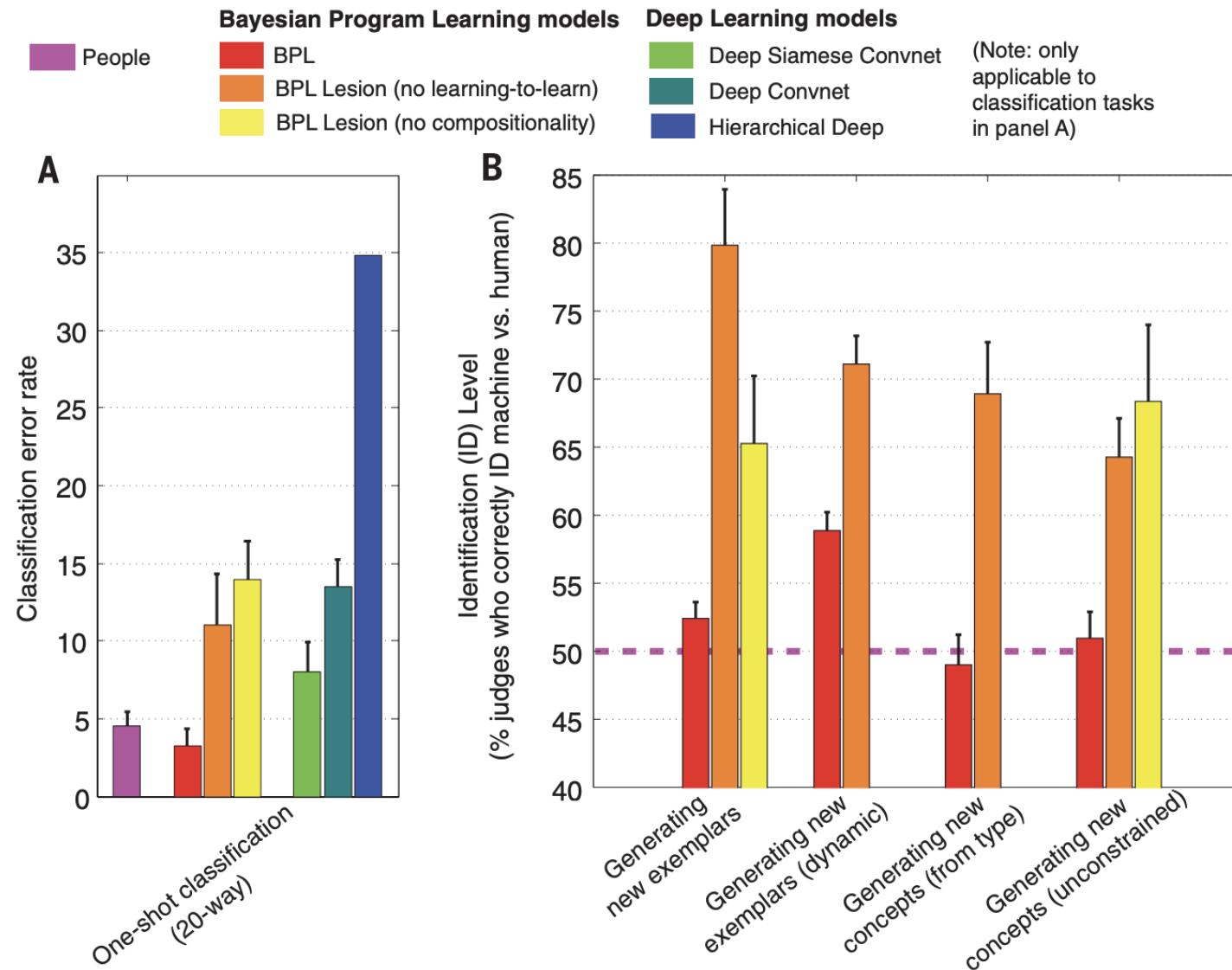
procedure GENERATETOKEN(ψ)

```

for  $i = 1 \dots \kappa$  do
     $S_i^{(m)} \leftarrow P(S_i^{(m)}|S_i)$                 ▷ Add motor variance
     $L_i^{(m)} \leftarrow P(L_i^{(m)}|R_i, T_1^{(m)}, \dots, T_{i-1}^{(m)})$  ▷ Sample part's start location
     $T_i^{(m)} \leftarrow f(L_i^{(m)}, S_i^{(m)})$  ▷ Compose a part's trajectory
end for
 $A^{(m)} \leftarrow P(A^{(m)})$                           ▷ Sample affine transform
 $I^{(m)} \leftarrow P(I^{(m)}|T^{(m)}, A^{(m)})$           ▷ Sample image
return  $I^{(m)}$ 

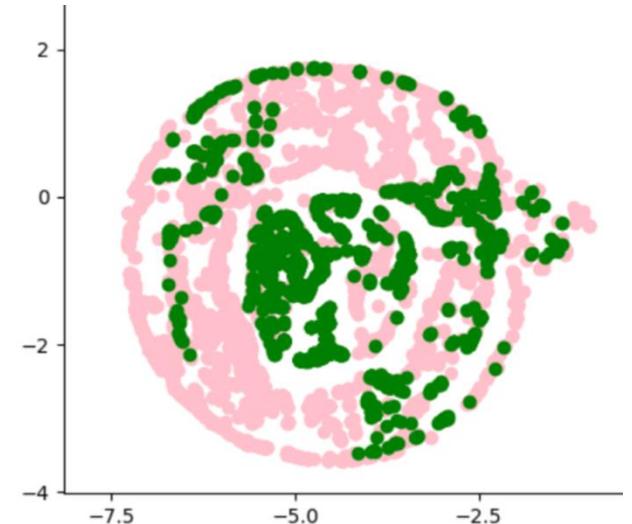
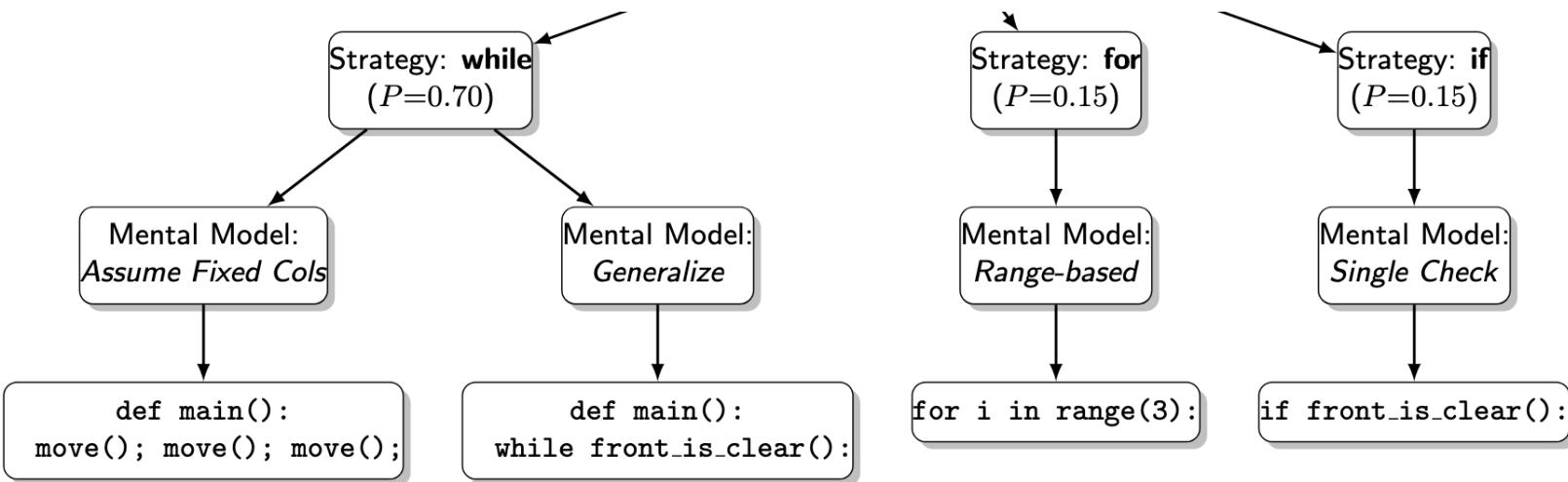
```

Human Level. And More!



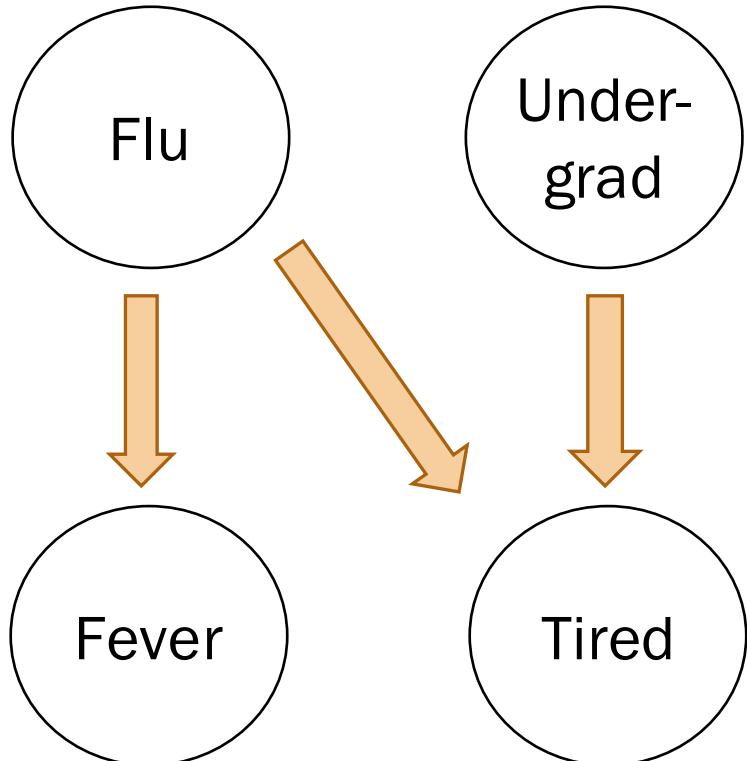
Generative Student Modeling

Juliette Woodrow, Chris Piech, 2021



Used generative grammars to simulate the most common buggy programs that TAs would see in LaIR.

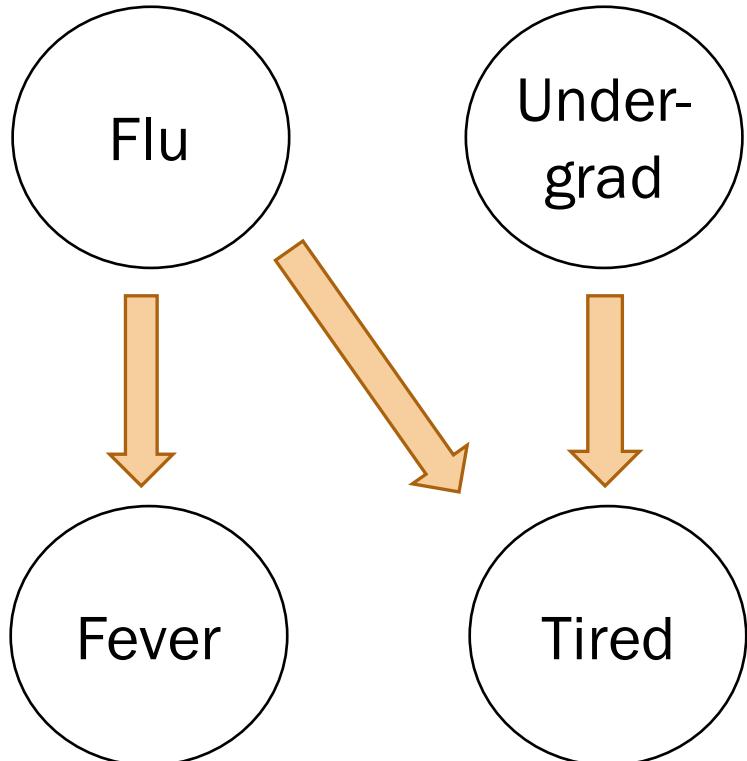
Generative Models make Independence Assumptions



What would a Stanford flu expert do?

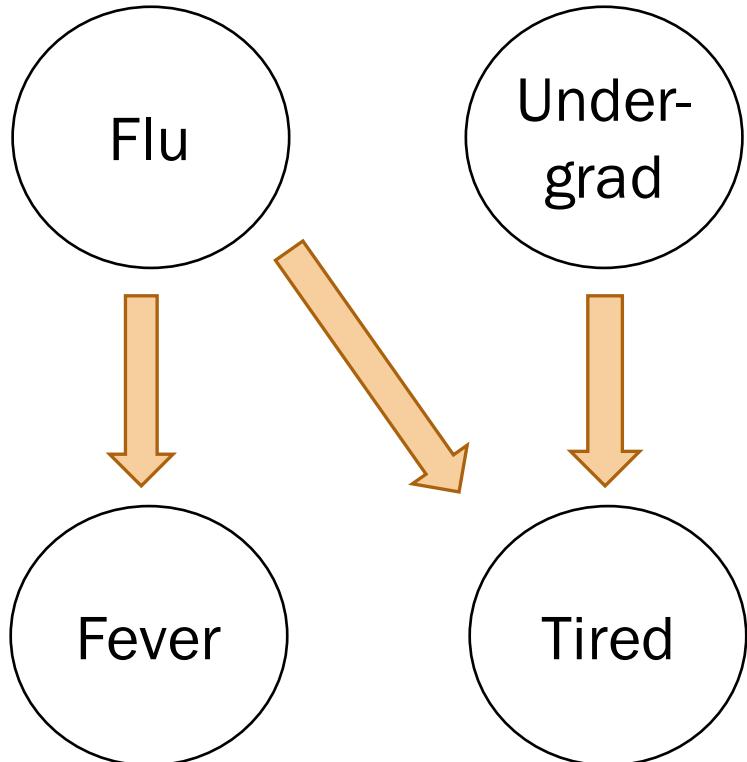
- 1. Describe the causality.
- 2. Provide $P(\text{values} | \text{causal parents})$ for each random variable
- 3. Implicitly assumes independences.

Generative Models make Independence Assumptions



Each random variable is **conditionally independent** of its causal non-descendants, **given its causal parents**.

Generative Models make Independence Assumptions

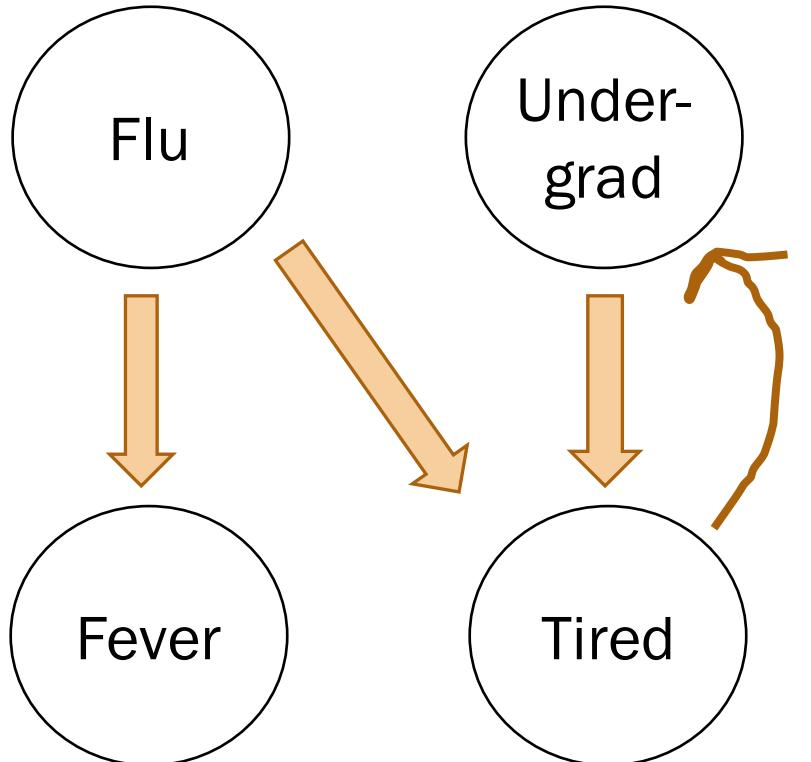


This model assumes that Flu and being an Undergraduate are independent.

Advanced: it also assumes that fever and tired are conditionally independent given Flu.

You need to tell a generative story. The independence assumptions come for free.

Bug: Constructing a Bayesian Network



Must be acyclic!

Three Guiding Questions

1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

Three Guiding Questions

1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

[suspense]

Computational Inference

Query: $P(\text{Flu} | \text{Fever, Tired})$

Resample N=10,000

1. Sample
10,000

2. Reject

3. Count
Target From
Remaining

ICON MAPPING

■ UG

■ Flu

■ Fever

■ Tired



INFERENCE STATS

Total N **10,000**

Matches Evidence **1,148**

Target (Flu+Evid) **870**

ESTIMATED PROBABILITY

0.758

Algorithm #2: Rejection Sampling

```
13  def main():
14      obs = get_observation()
15      samples = sample_a_ton()  
16      prob = prob_flu_given_obs(samples, obs)
17      print('Observation = ', obs)
18      print('Pr(Flu | Obs) = ', prob)
```

Algorithm #2: Rejection Sampling

```
13 def main():
14     obs = get_observation()
15     samples = sample_a_ton() # This line is highlighted
16     prob = prob_flu_given_obs(samples)
17     print('Observation = ', obs)
18     print('Pr(Flu | Obs) = ', prob)
```

? webMd — -zsh — 56x42

Algorithm #2: Rejection Sampling

```
13  def main():
14      obs = get_observation()
15      samples = sample_a_thon()
16      prob = prob_flu_given_obs(samples, obs)
17      print('Observation = ', obs)
18      print('Pr(Flu | Obs) = ', prob)
```

Algorithm #2: Rejection Sampling

```
35  def prob_flu_given_obs(samples, obs):  
36      """  
37          Calculate the probability of flu given many  
38          samples from the joint distribution and a set  
39          of observations to condition on.  
40      """  
41      # reject all samples which don't align  
42      # with condition  
43      keep_samples = []  
44      for sample in samples:  
45          if check_obs_match(sample, obs):  
46              keep_samples.append(sample)  
47  
48      # from remaining, simply count...  
49      flu_count = 0  
50      for sample in keep_samples:  
51          if sample['flu'] == 1:  
52              flu_count += 1  
53  
54      # counting can be so sweet...  
55      return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```
35  def prob_flu_given_obs(samples, obs):  
36      """  
37          Calculate the probability of flu given many  
38          samples from the joint distribution and a set  
39          of observations to condition on.  
40      """  
41      # reject all samples which don't align  
42      # with condition  
43      keep_samples = []  
44      for sample in samples:  
45          if check_obs_match(sample, obs):  
46              keep_samples.append(sample)  
47  
48      # from remaining, simply count...  
49      flu_count = 0  
50      for sample in keep_samples:  
51          if sample['flu'] == 1:  
52              flu_count += 1  
53  
54      # counting can be so sweet...  
55      return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```
35  def prob_flu_given_obs(samples, obs):  
36      """  
37          Calculate the probability of flu given many  
38          samples from the joint distribution and a set  
39          of observations to condition on.  
40      """  
41      # reject all samples which don't align  
42      # with condition  
43      keep_samples = []  
44      for sample in samples:  
45          if check_obs_match(sample, obs):  
46              keep_samples.append(sample)  
47  
48      # from remaining, simply count...  
49      flu_count = 0  
50      for sample in keep_samples:  
51          if sample['flu'] == 1:  
52              flu_count += 1  
53  
54      # counting can be so sweet...  
55      return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```
35 def prob_flu_given_obs(samples, obs):
36     """
37     Calculate the probability of flu given many
38     samples from the joint distribution and a
39     of observations to condition on.
40     """
41     # reject all samples which don't align
42     # with condition
43     keep_samples = []
44     for sample in samples:
45         if check_obs_match(sample, obs):
46             keep_samples.append(sample)
47
48     # from remaining, simply count...
49     flu_count = 0
50     for sample in keep_samples:
51         if sample['flu'] == 1:
52             flu_count += 1
53
54     # counting can be so sweet...
55     return float(flu_count) / len(keep_samples)
```

```
webMd --zsh-- 53x25
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
-----
Observation = {'flu': None, 'undergrad': 1, 'fever': None, 'tired': 1}
Pr(Flu | Obs) = 0.1228646517739816
piech@Chriss-MBP-5 webMd %
```

Lets try it!

BACK 
TO THE **CODE**

Rejection sampling algorithm

Inference
question:

What is $P(F_{lu} = 1 | U = 1, T = 1)$?

$$\text{probability} \approx \frac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$$

Why would this definition of approximate probability make sense?



Why would this approximate probability make sense?

Inference
question:

What is $P(F_{lu} = 1 | U = 1, T = 1)$?

$$\text{probability} \approx \frac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$$

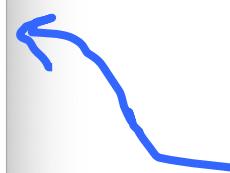
Recall our definition of
probability as a frequency:

$$P(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n} \quad \begin{aligned} n &= \# \text{ of total trials} \\ n(E) &= \# \text{ trials where } E \text{ occurs} \end{aligned}$$

```
webMd --zsh -- 53x25
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 0}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
-----
Observation =  {'flu': None, 'undergrad': 1, 'fever': None, 'tired': 1}
Pr(Flu | Obs) =  0.1228646517739816
piech@Chriss-MBP-5 webMd %
```



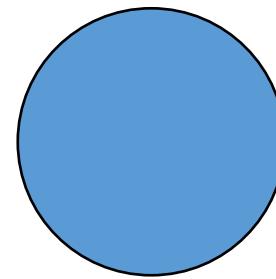
If you can sample enough
from the joint distribution,
you can answer any
probability question



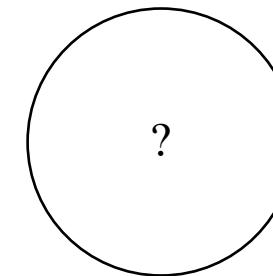
Each one of these is one joint
sample

Lets try another question

You observe that someone has a **recessive** gene.
What is the probability that their **cousin** has the same recessive gene?
Each person has a 1/20 chance of having the recessive gene.

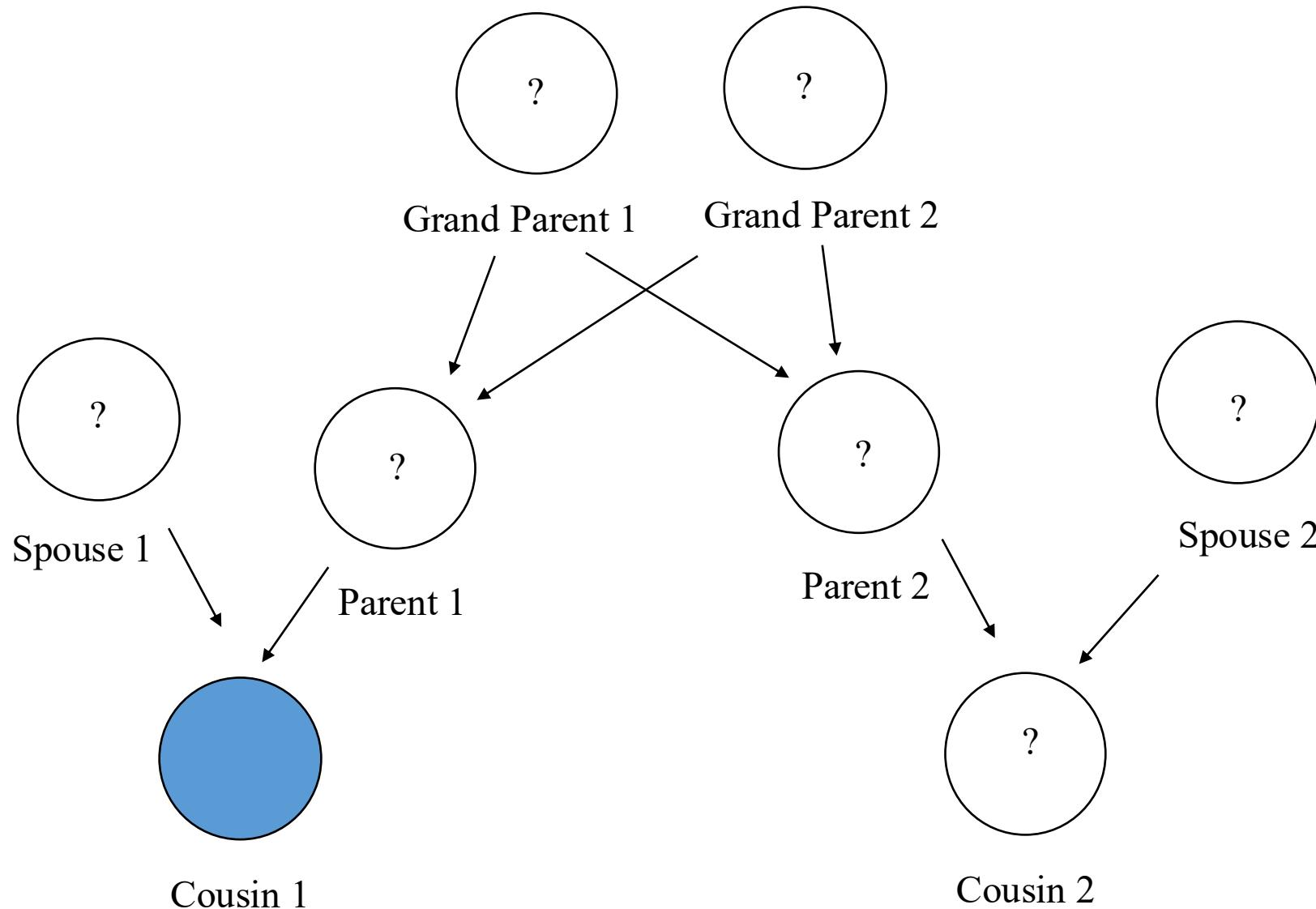


Cousin 1

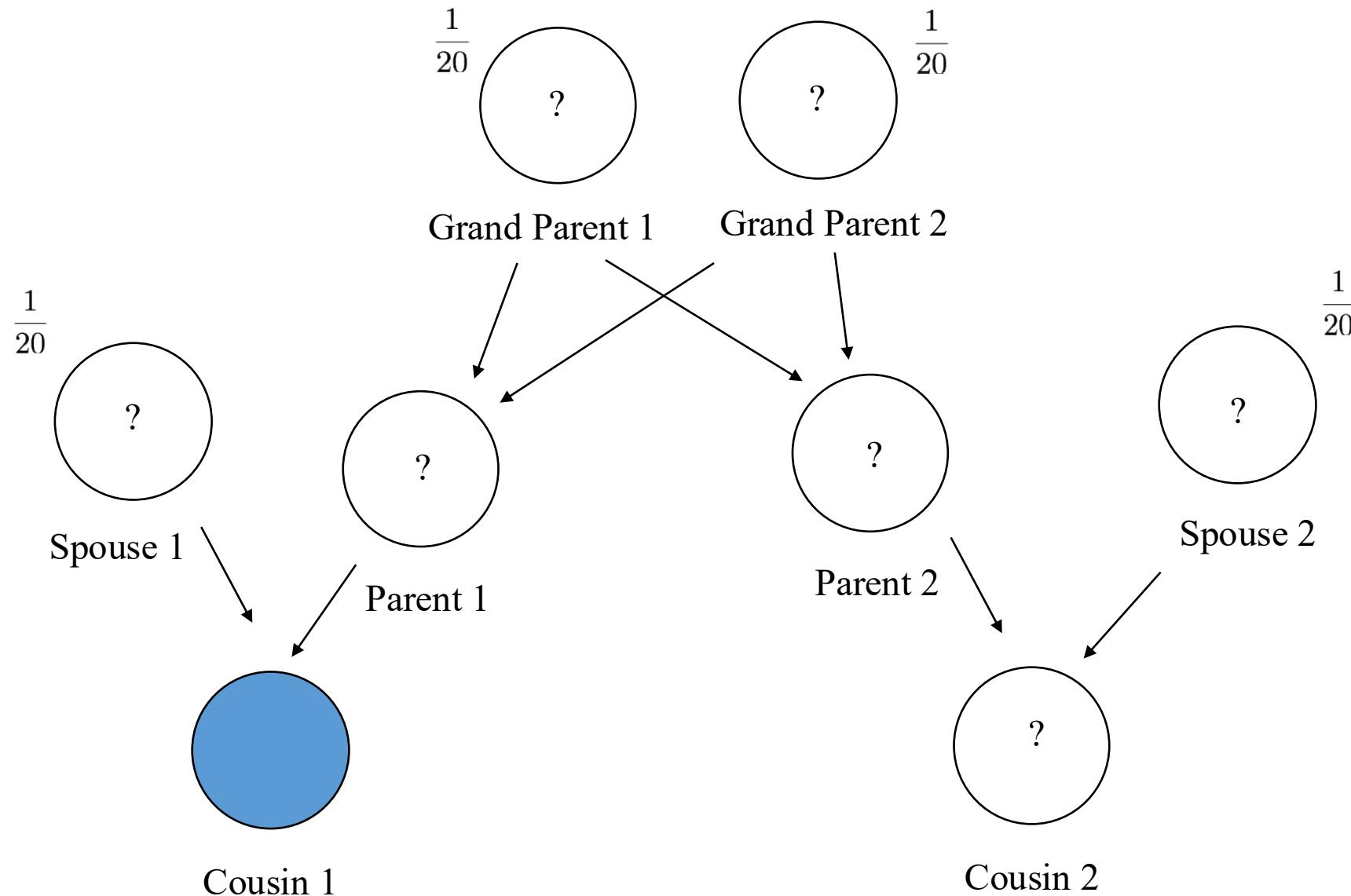


Cousin 2

You observe that someone has a **recessive** gene.
What is the probability that their **cousin** has the same recessive gene?

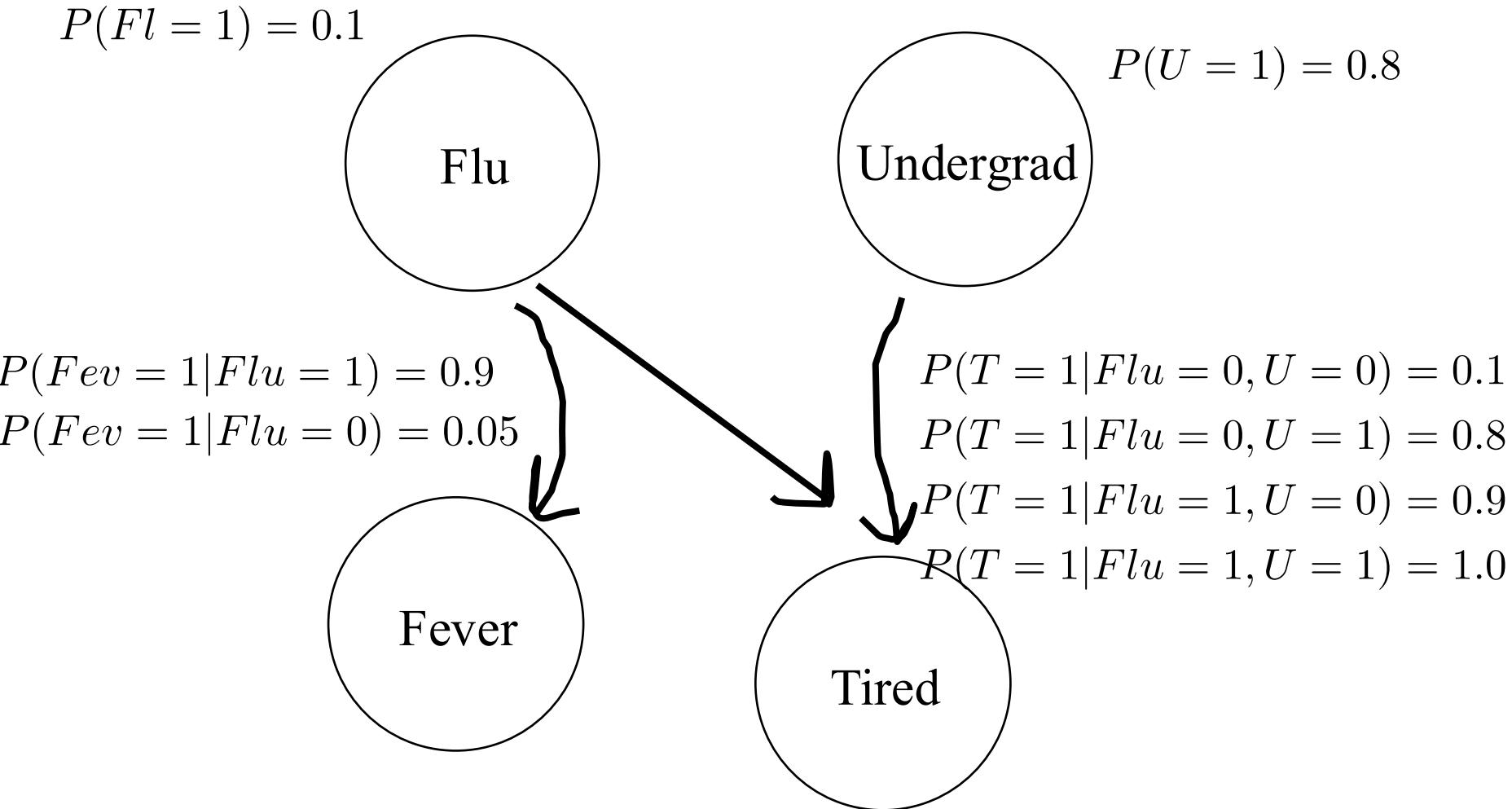


You observe that someone has a **recessive** gene.
What is the probability that their **cousin** has the same recessive gene?

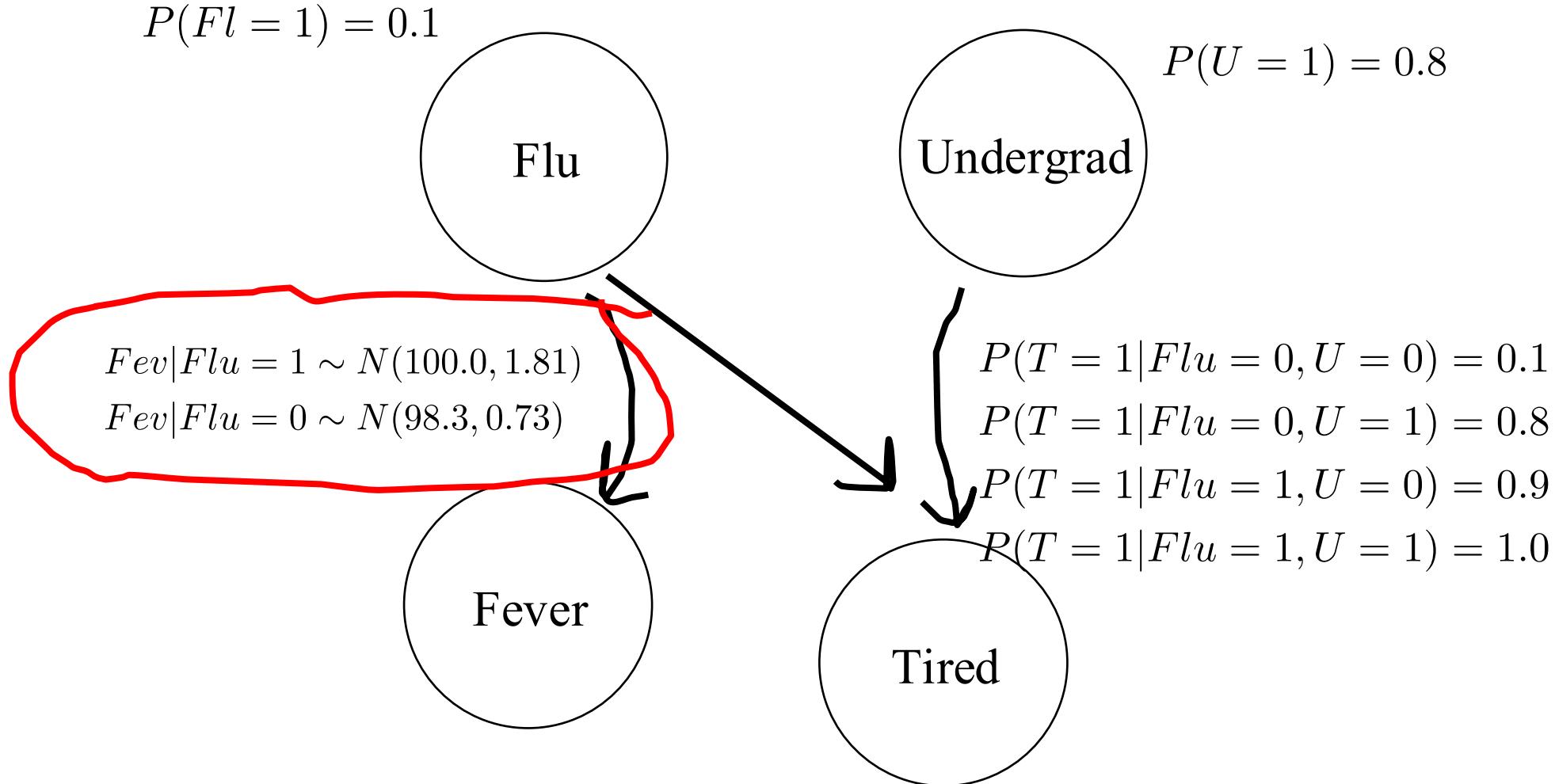


What's the matter with
rejection sampling?

Probabilistic Model



Probabilistic Model



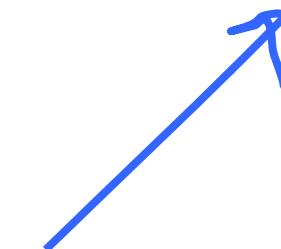
Back to the code !!

Many Algorithms

Markov Chain



MCMC



Monte Carlo

Many Algorithms

```
webMd -- bash -- 20x20
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[0, 1, 101.0, 0]
[0, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
Pr(Flu) = 0.9773
>
```

MCMC is a way to sample
with conditioned variables
fixed

Each one of these
is one posterior
sample:



[Flu, Undergrad, Fever, Tired]

Many Algorithms

Rejection
Sampling



MCMC



Pyro



Idea2Text



Three Guiding Questions

1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

Three Guiding Questions

1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

ROCK

The Sound: Vigorous, defiant, energetic, inventive

The Roots: Rhythm & blues, country

The Pioneers: Bill Haley, Chuck Berry, Fats Domino, Little Richard, Buddy Holly, Elvis Presley

The Places: Cleveland, New Orleans, Detroit, New York City

The Ensemble: Electric guitar, bass, drums, keyboard, vocals

"We're a rock group. We're young, ready, sensational and wild."

— Angus Young (R. 1990)

HIP-HOP Rap

The Sound: Rhythmic, unvarnished, adaptable, streetwise

The Roots: Rhythm & blues, soul, funk, reggae

The Pioneers: Afrika Bambaataa, Kool Herc, DJ Hollywood, Grandmaster Flash, Kurtis Blow, Grandmaster Caz

The Places: New York City (South Bronx)

The Ensemble: Vinyl, turntable, vocals

"The beautiful thing about hip-hop is it's like the audio version of graffiti. You can take any found music and do it in your own way and it'll be a hip-hop song."

— Salt N' Pepa (R. 1991)

LATIN American

The Sound: Syncopated, enthusiastic, diverse, vibrant

The Roots: Spain, Africa, Caribbean, South America

The Pioneers: Arsenio Rodriguez, Machito, Perez Prado, Tito Puente, Celia Cruz, Johnny Pacheco

The Places: Cuba, Puerto Rico, Mexico, Miami, New York

The Ensemble: Congas, bongos, maracas, güiro, guitar, vocals

"The emphasis was dancing and rhythm. I grew up with an emphasis on Latin dancing that were familiar to people in Latin America—and everybody identified with the songs."

— Ruben Blades (R. 1992)

Folk

The Sound: Grassroots, narrative, sincere, lyrical

The Roots: Ballads, immigrant folklore, spirituals, cowboy songs

The Pioneers: Lead Belly, Odetta, Woody Guthrie, Pete Seeger, Bob Dylan, Joan Baez

The Places: Appalachia, Deep South, Western frontier

The Ensemble: Guitar, banjo, fiddle, accordion, vocals

"I liked the rhythms [of folk music]. I liked the melodies, time-tested by generations of singers. Above all, I liked the words ... they seemed fresh, straightforward, honest."

— Pete Seeger (R. 2014)

CLASSICAL

The Sound: Genuine, uncomplicated, nostalgic, informal

The Roots: European ballads, folk and gospel songs

The Pioneers: Uncle Dave Macon, the Carter Family, Jimmie Rodgers, Roy Acuff, Gene Autry, Bill Monroe

The Places: Appalachia, Nashville, Chicago, Western U.S.

The Ensemble: Fiddle, banjo, guitar, harmonica, accordion, vocals

CLASSICAL

The Sound: Intricate, polished, structured, harmonious

The Roots: Sacred music, choral chants, madrigals, dance rhythms

The Pioneers: J.S. Bach, Handel, Haydn, Mozart, Beethoven, Brahms

The Places: Austria, Germany, France, Italy

The Ensemble: Strings, woodwinds, brass, percussion, vocals

"I carry my thoughts about with me a long time ... before writing them down, I change many things, discuss others, and try again and again until I am satisfied."

— Ludwig van Beethoven (1770-1827)

AutoSave OFF

Home Insert Page Layout Formulas Data Share

Clipboard Font Alignment Number

Conditional Formatting Format as Table Cell Styles Cells Editing

C15

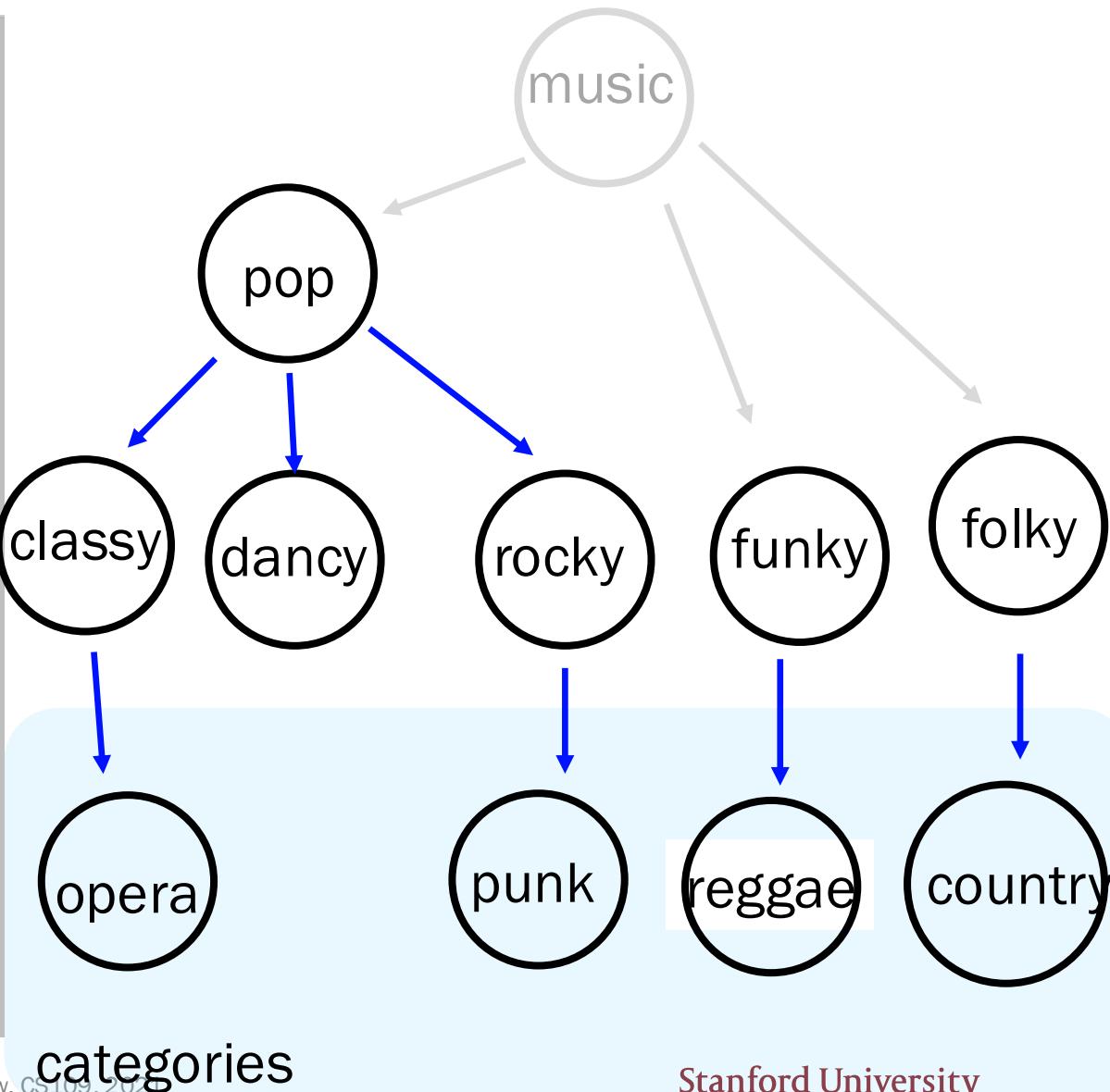
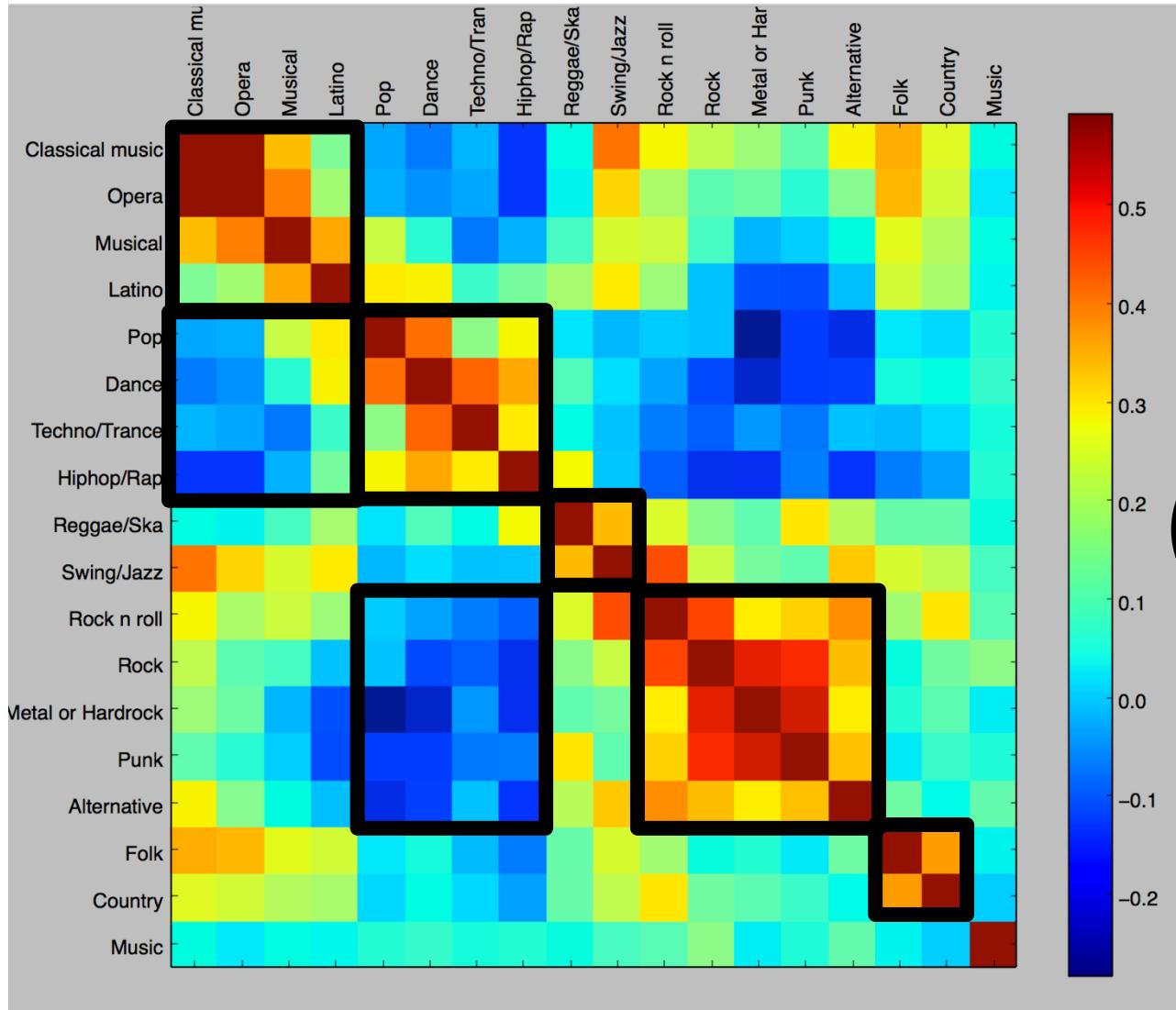
A B C D E F G H

	Music	Dance	Folk	Country	Classical music	Musical	Pop	Rock	Me
1	5	2	1	2	2	1	5	5	5
2	4	2	1	1	1	1	2	3	5
3	5	2	2	3	1	1	4	3	5
4	5	4	3	2	2	4	3	5	3
5	5	2	1	1	1	1	1	2	2
6	5	4	3	2	2	4	3	5	3
7	5	2	3	2	2	3	3	2	5
8	5	5	3	1	1	2	2	2	5
9	5	3	2	1	1	2	2	2	4
10	5	3	1	1	1	1	2	4	5
11	5	2	5	2	2	2	5	3	5
12	5	3	2	1	1	2	3	4	3
13	5	1	1	1	1	4	1	2	5
14	5	1	2	1	1	4	3	3	5
15	5	5	3	2	1	5	5	2	5
16	5	2	1	1	1	2	3	4	5
17	1	2	2	3	4	3	3	3	5
18	5	3	1	1	1	1	2	4	4
19	5	3	3	3	3	2	2	4	4
20	5	5	4	3	4	4	5	5	4
21	5	3	3	2	2	4	2	2	4
22	5	3	2	3	4	3	3	2	5
23	5	1	1	3	2	2	2	2	5
24	5	3	2	3	3	3	3	4	5
25	5	4	2	2	2	2	4	4	5
26	5	3	1	1	1	4	3	3	5
27	5	4	2	1	2	3	5	1	1
28	5	5	5	4	5	3	4	4	4
29	4	3	4	1	1	3	2	2	4
30	5	5	1	1	1	1	1	3	4
31	5	3	4	2	3	3	3	3	4
32	4	4	3	3	3	3	3	4	4
33	4	4	1	3	2	3	5	3	3
34	5	3	1	3	2	3	3	3	4
35	5	2	2	3	4	5	4	4	3

music

Ready

From Correlation to Bayes Net!



Why is it harder to
find independences
here than for bat DNA
expression?

AutoSave OFF > Q Search Sheet

Home Insert Page Layout Formulas Data > Share

Clipboard Font Alignment Number Conditional Formatting
Format as Table
Cell Styles Cells Editing

C15 X ✓ fx 3

	A	B	C	D	E	F	G	H	M
1	Music	Dance	Folk	Country	Classical music	Musical	Pop	Rock	Me
2	5	2	1	2	2	1	1	5	5
3	4	2	1	1	1	1	2	3	5
4	5	2	2	3	4	5	3	5	5
5	5	2	1	1	1	1	1	2	2
6	5	4	3	2	4	3	5	3	3
7	5	2	3	2	3	3	2	5	5
8	5	5	3	1	2	2	2	5	3
9	5	3	2	1	2	2	2	4	5
10	5	3	1	1	2	4	3	5	5
11	5	2	5	2	2	5	3	5	5
12	5	3	2	1	2	3	4	3	3
13	5	1	1	1	4	1	2	5	5
14	5	1	2	1	4	3	3	5	5
15	5	5	3	2	1	5	5	2	2
16	5	2	1	1	2	3	4	5	5
17	1	2	2	3	4	3	3	5	5
18	5	3	1	1	1	2	4	4	4
19	5	3	3	3	2	2	2	4	4
20	5	5	4	3	4	5	5	4	4
21	5	3	3	2	4	2	2	4	4
22	5	3	2	3	4	3	2	5	5
23	5	1	1	3	2	2	2	2	5
24	5	3	2	3	3	3	3	4	4
25	5	4	2	2	2	4	4	5	5
26	5	3	1	1	4	3	3	5	5
27	5	4	2	1	2	3	5	1	1
28	5	5	5	4	5	3	4	4	4
29	4	3	4	1	3	2	2	4	4
30	5	5	1	1	1	1	1	3	4
31	5	3	4	2	3	3	3	3	4
32	4	4	3	3	3	3	3	4	4
33	4	4	1	3	2	3	5	3	3
34	5	3	1	3	2	3	3	4	4
35	5	2	2	3	4	5	4	4	3

music +

Ready

100%

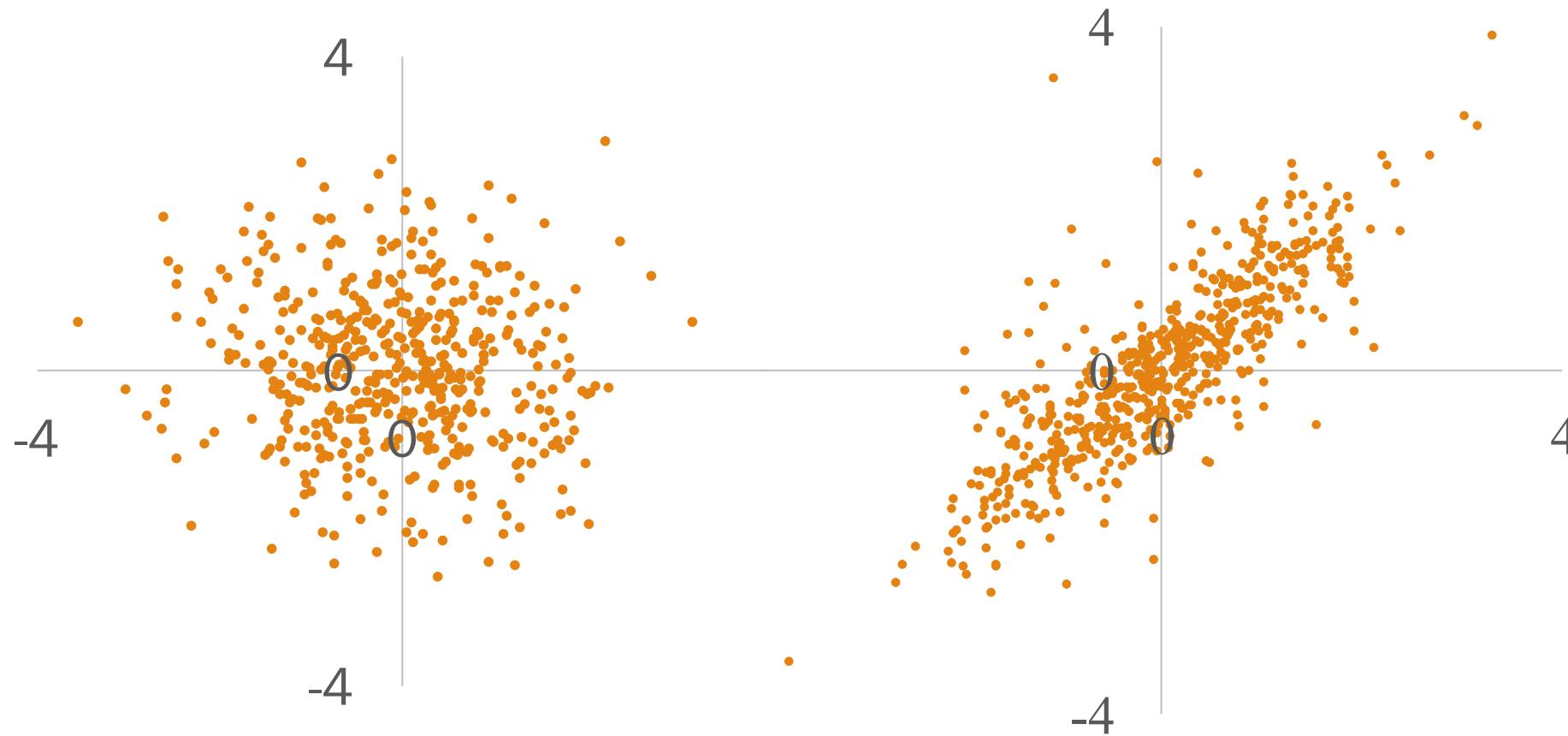
Bat Data

Gene1	Gene2	Gene3	Gene4	Gene5	Trait
TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
TRUE	FALSE	TRUE	TRUE	TRUE	FALSE
FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
FALSE	TRUE	FALSE	TRUE	FALSE	FALSE
TRUE	TRUE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	TRUE	TRUE	TRUE	FALSE
FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
...					
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE

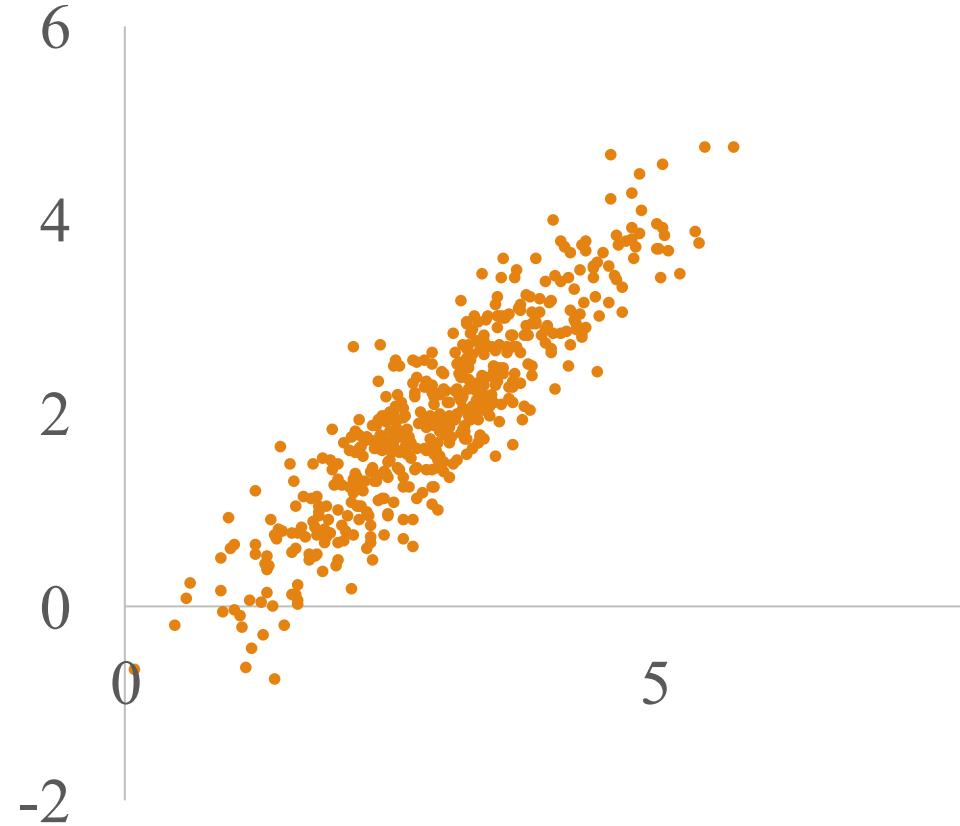
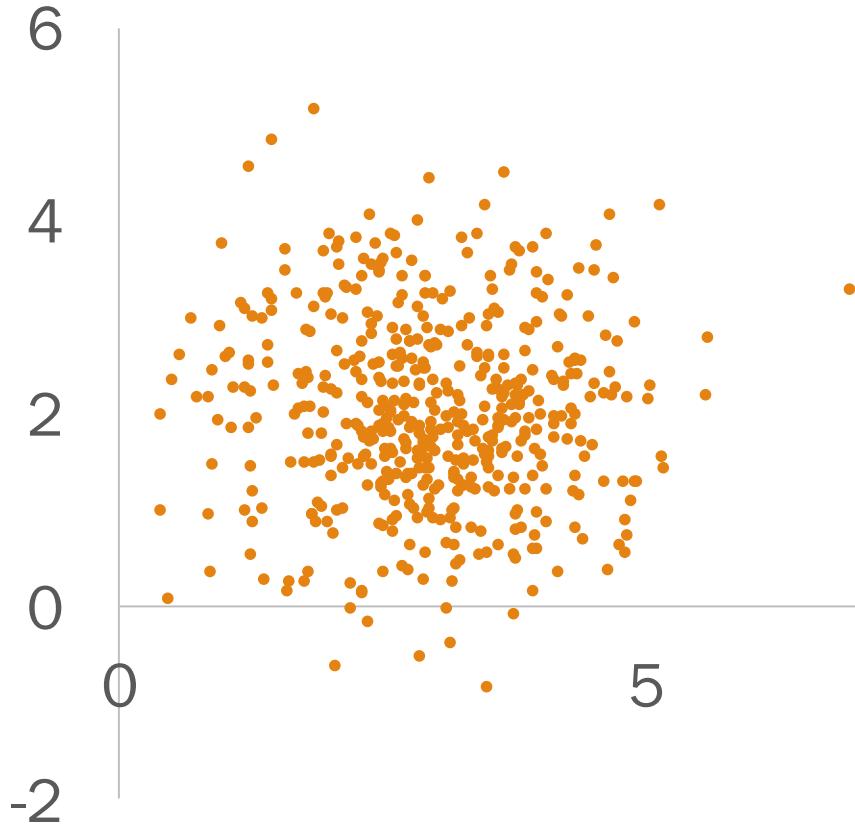
Expression Amount

Gene5	Trait
0.76	0.83
0.94	0.85
0.82	0.03
0.94	0.32
0.50	0.10
0.40	0.53
0.90	0.67
0.29	0.71
0.72	0.25
0.15	0.24
0.79	0.98
0.68	0.77
0.71	0.37
0.36	0.18
0.62	0.08
0.59	0.38
0.82	0.76

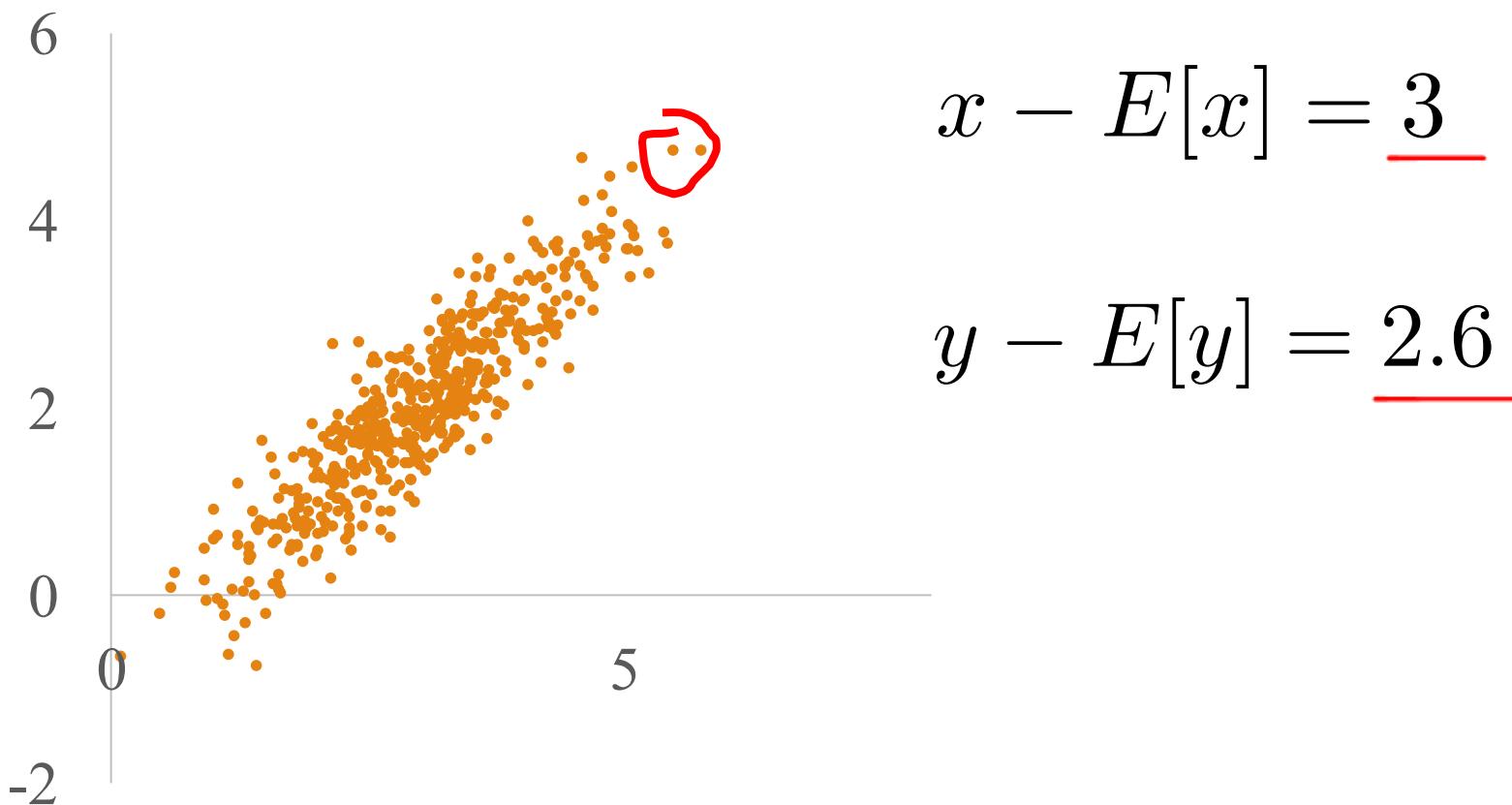
Spot The Difference



Spot The Difference

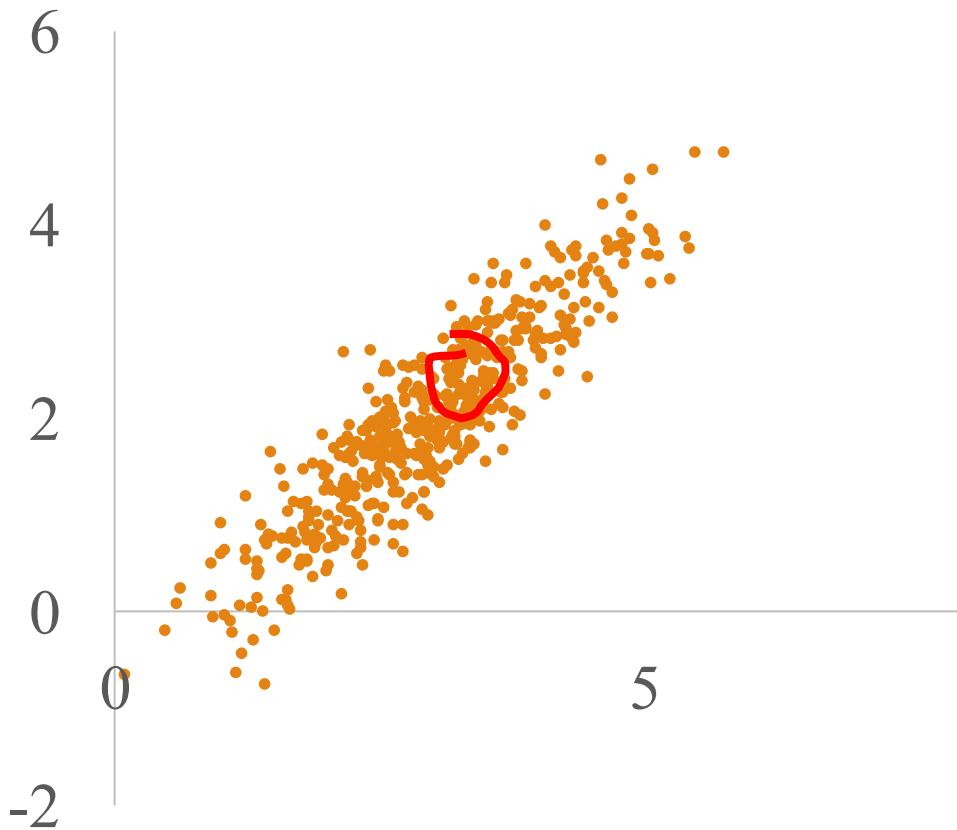


Vary Together



$$(x - E[x])(y - E[y]) = 7.8$$

Vary Together

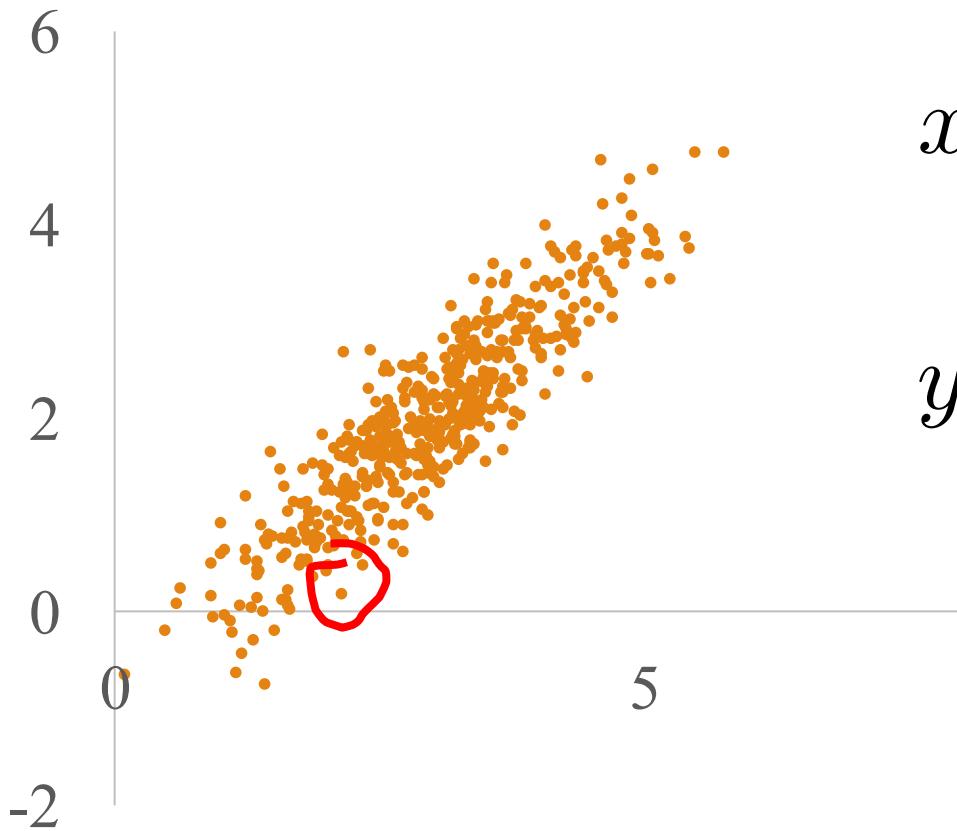


$$x - E[x] \approx 0$$

$$y - E[y] \approx 0$$

$$(x - E[x])(y - E[y]) = 0$$

Vary Together

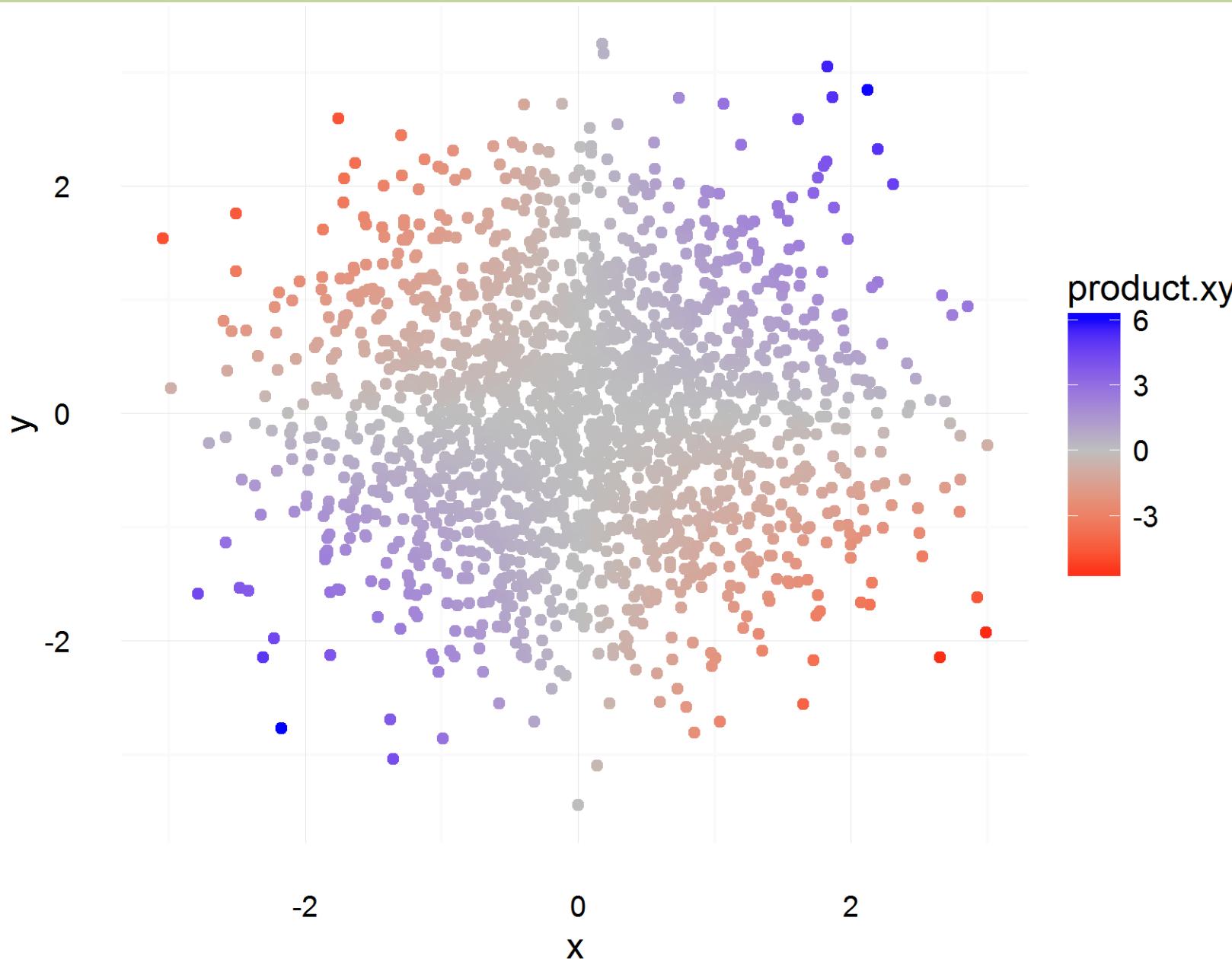


$$x - E[x] = -1.1$$

$$y - E[y] = -2.8$$

$$(x - E[x])(y - E[y]) \approx 3.1$$

Understanding Covariance



The Dance of the Covariance

Say X and Y are arbitrary random variables

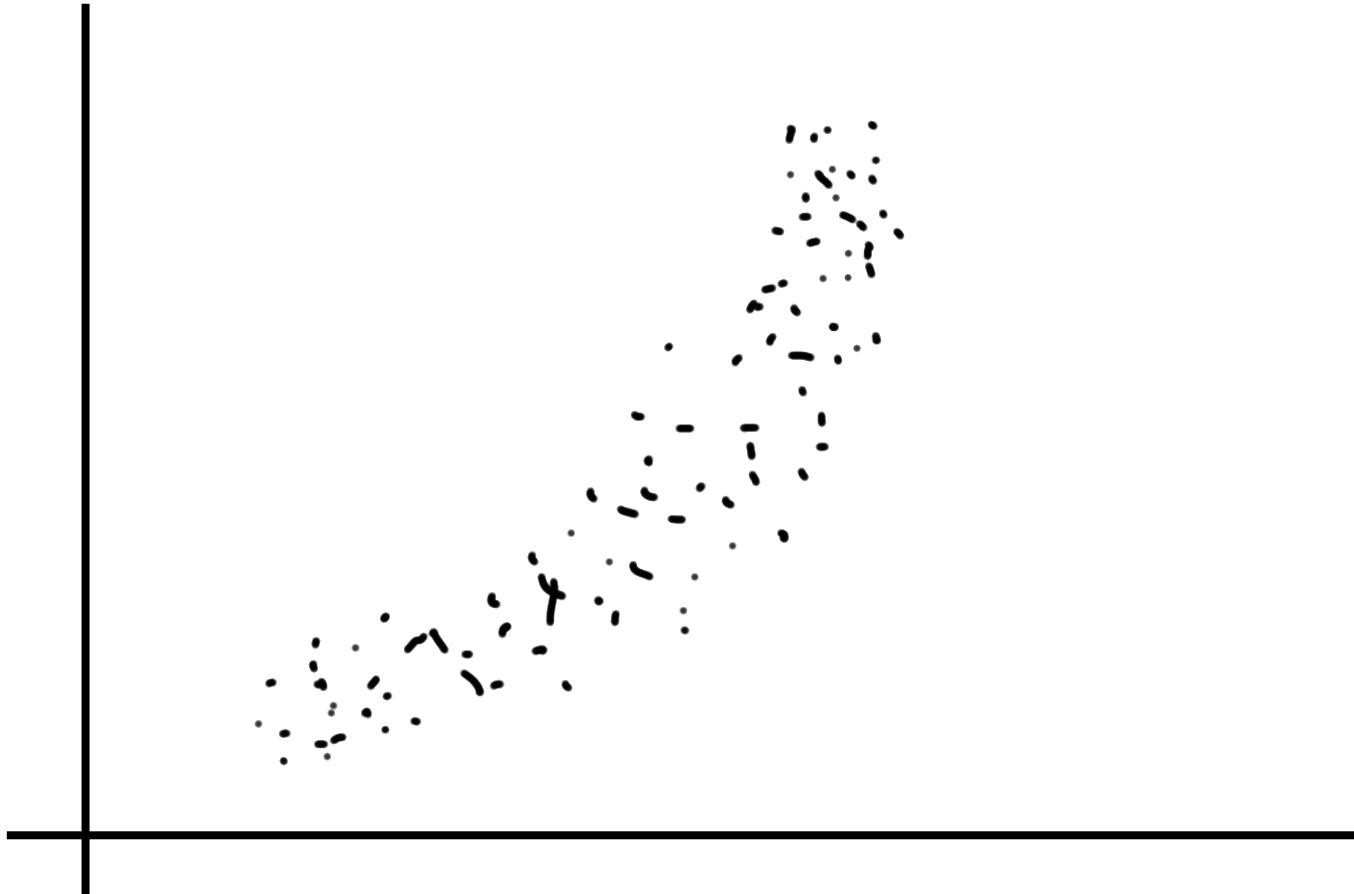
Covariance of X and Y :

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

x	y	$(x - E[X])(y - E[Y])p(x,y)$
Above mean	Above mean	Positive
Below mean	Below mean	Positive
Below mean	Above mean	Negative
Above mean	Below mean	Negative

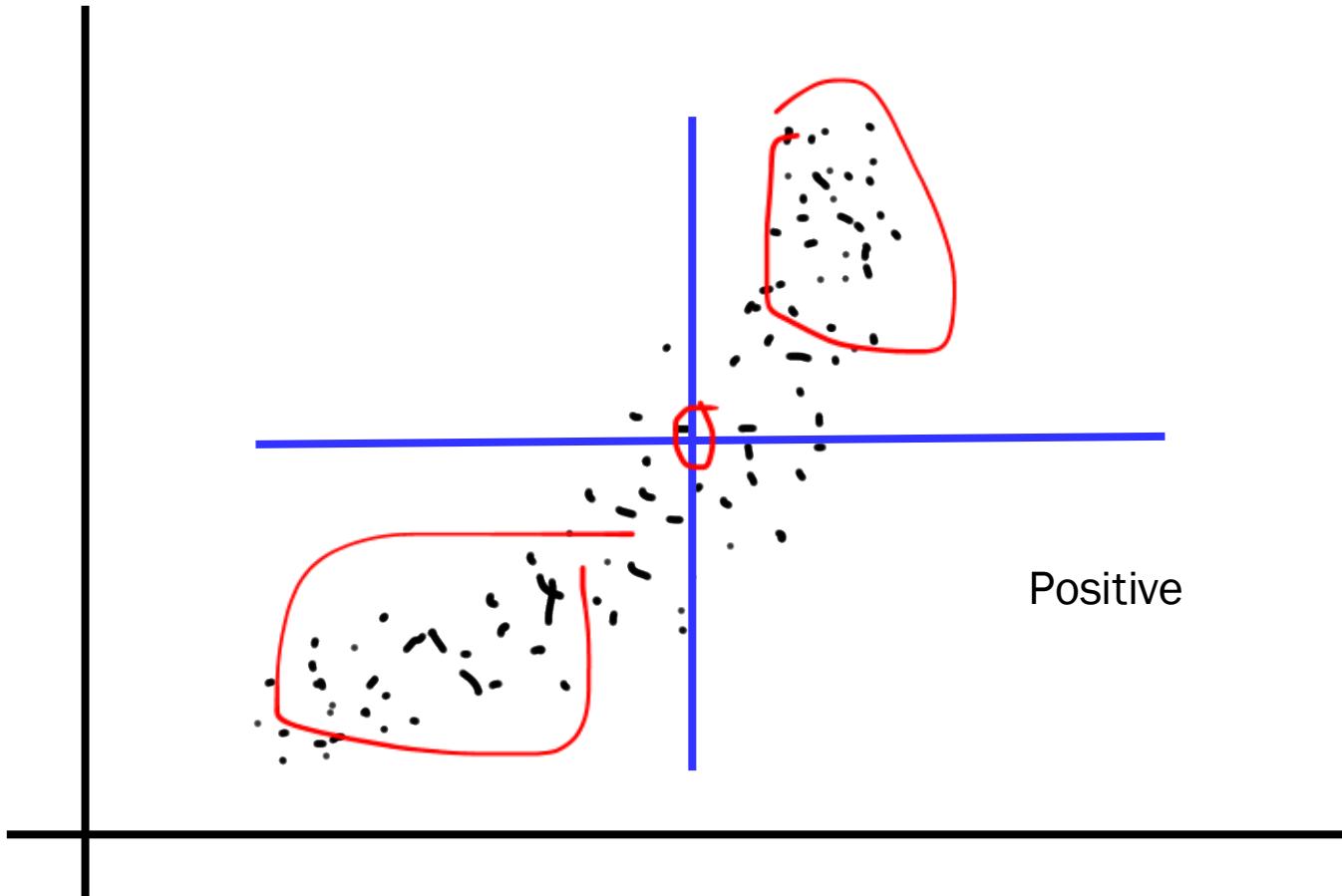
Covariance

Poll: (a) positive, (b) negative, (c) zero



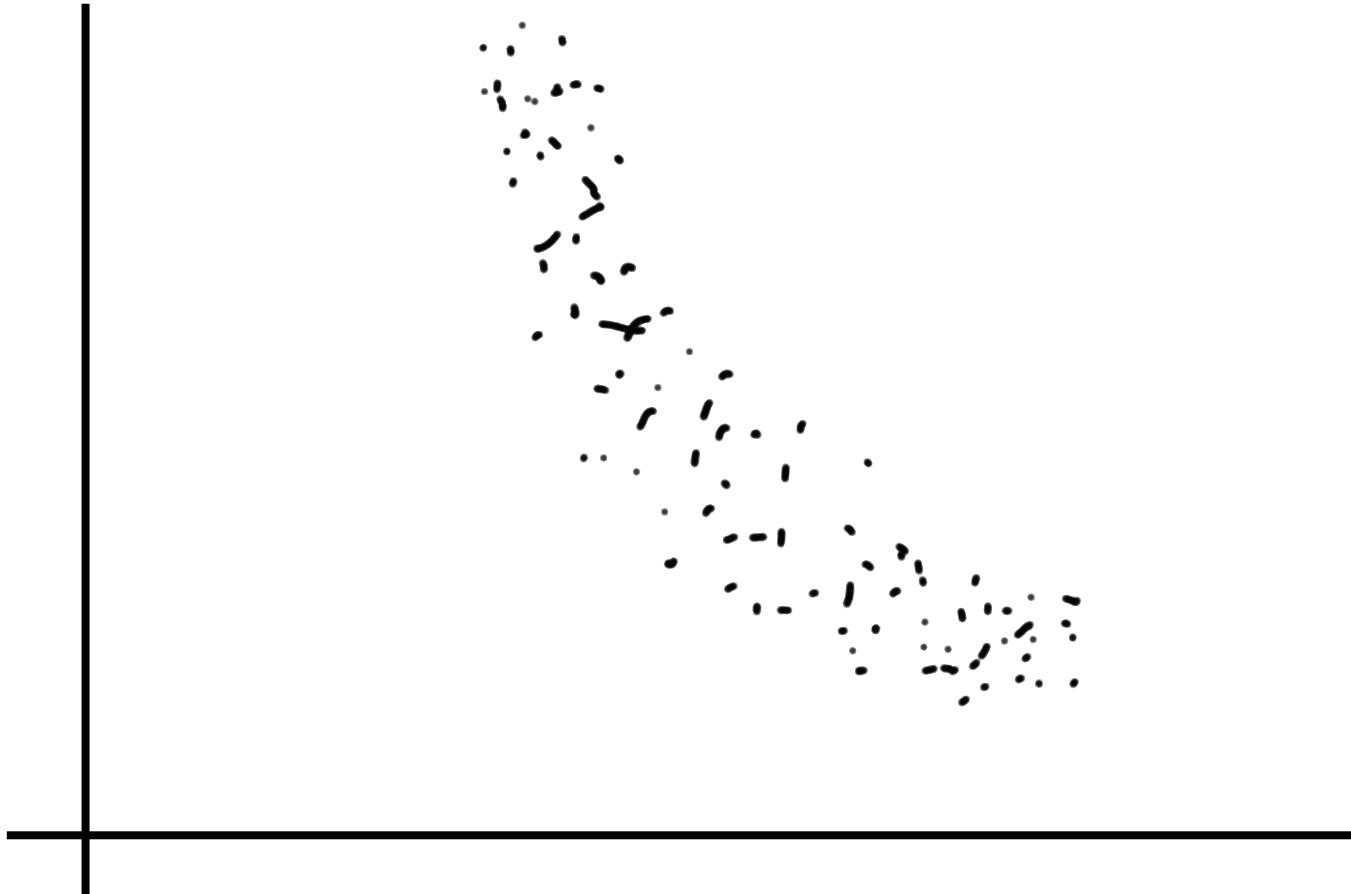
Covariance

Is the Covariance: (a) positive, (b) negative, (c) zero



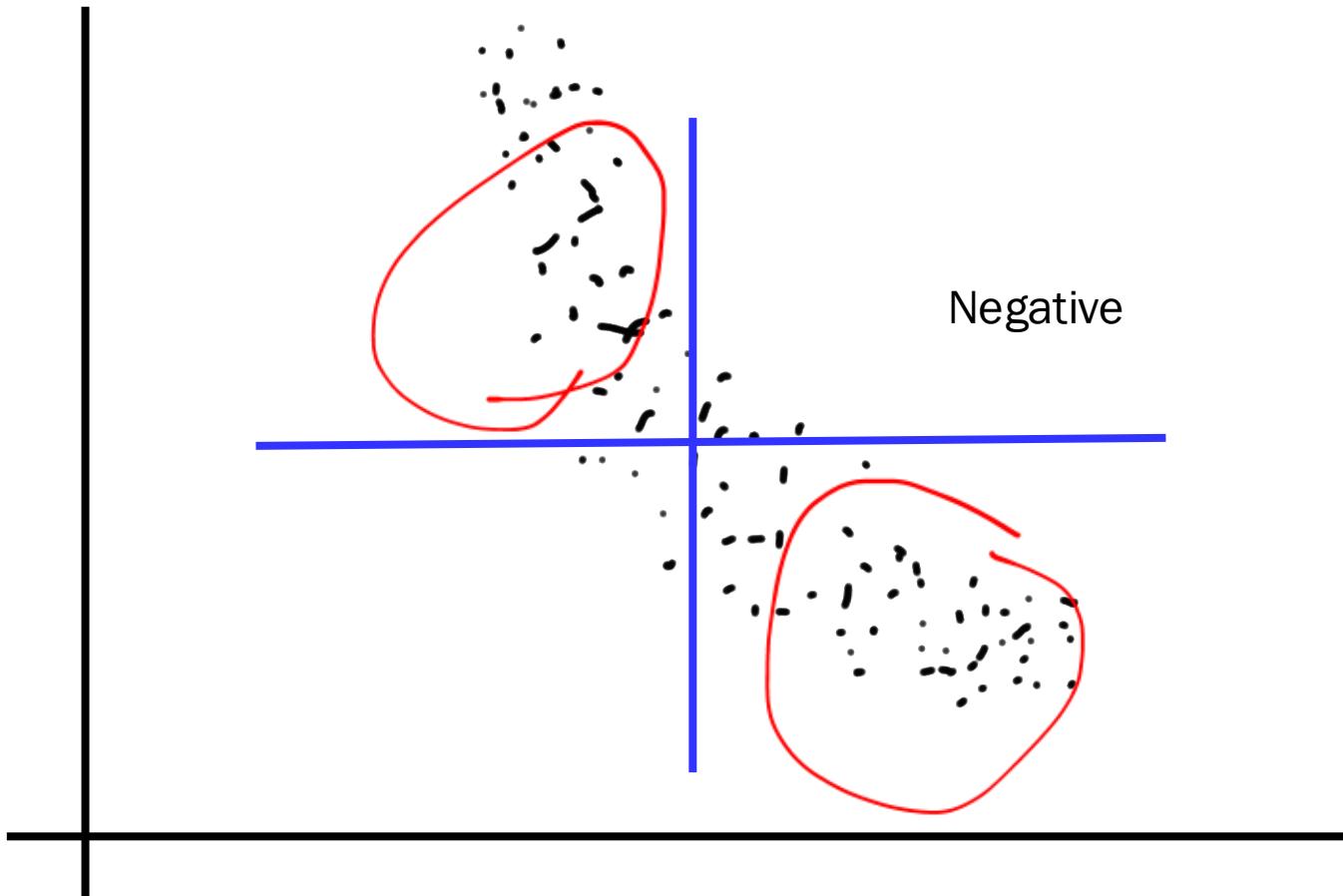
Covariance

Is the Covariance: (a) positive, (b) negative, (c) zero



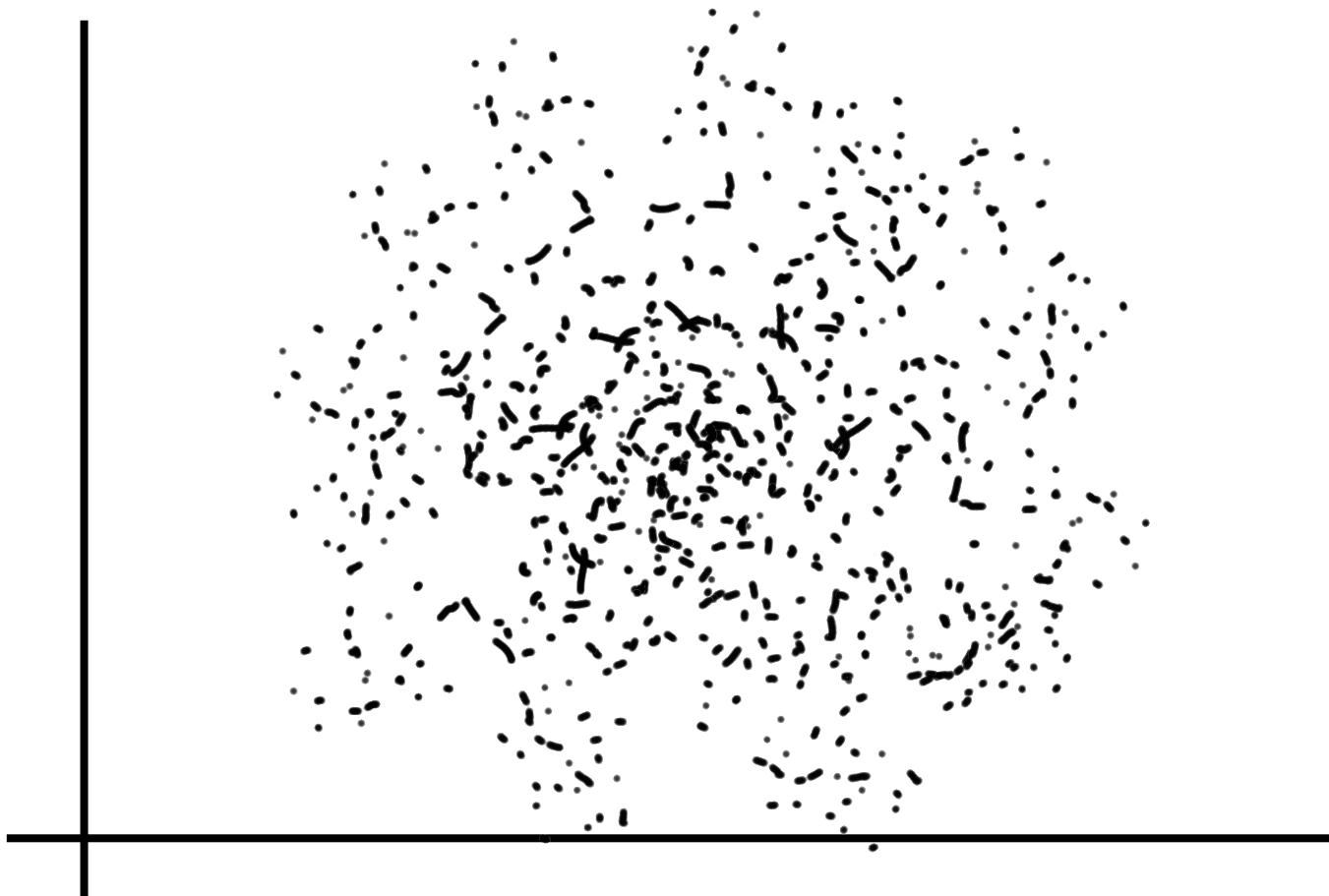
Covariance

Is the Covariance: (a) positive, (b) negative, (c) zero



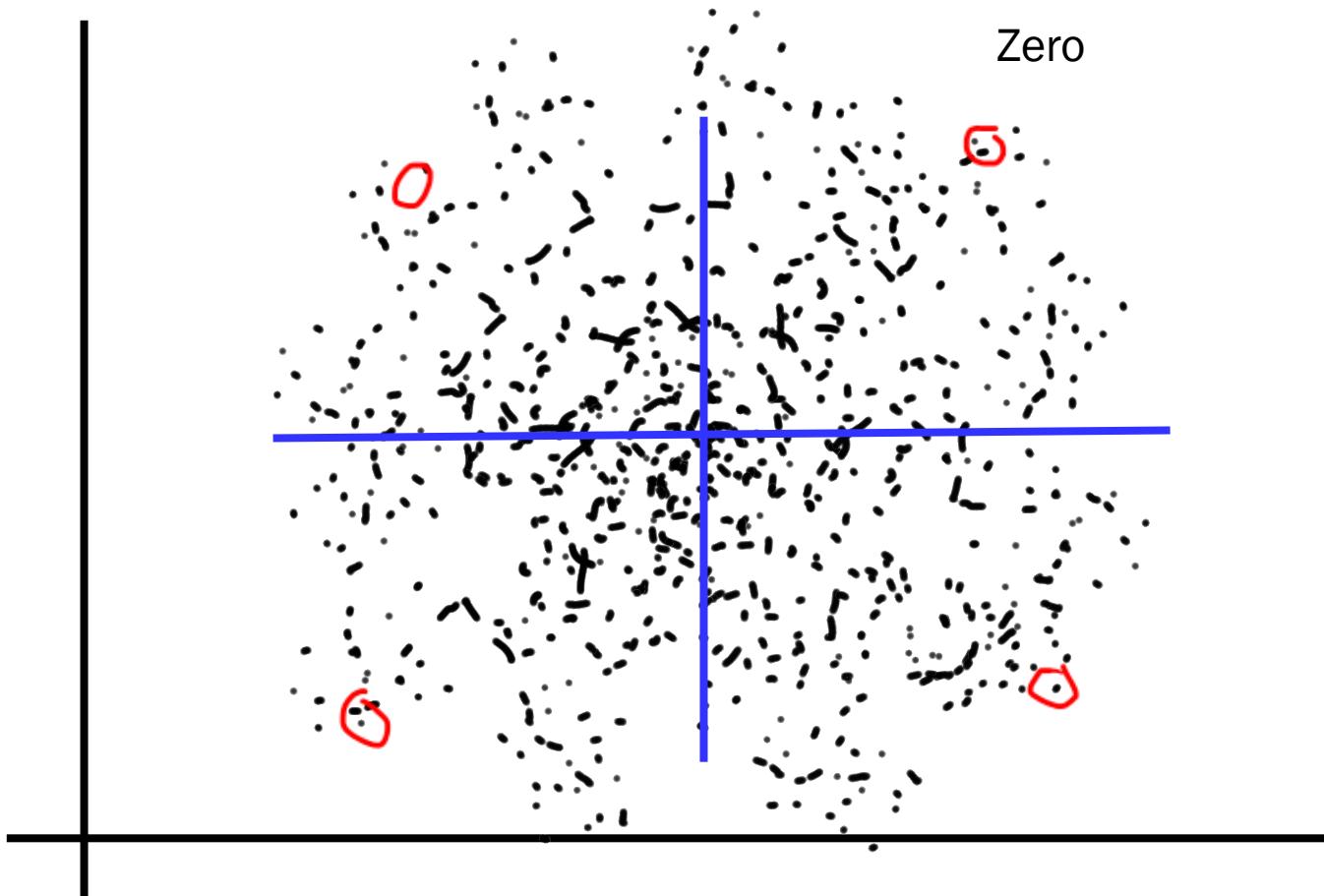
Covariance

Is the Covariance: (a) positive, (b) negative, (c) zero



Covariance

Is the Covariance: (a) positive, (b) negative, (c) zero



The Dance of the Covariance

Say X and Y are arbitrary random variables

Covariance of X and Y :

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

Equivalently:

$$\begin{aligned}\text{Cov}(X, Y) &= E[XY - E[X]Y - XE[Y] + E[Y]E[X]] \\ &= E[XY] - E[X]E[Y] - E[X]E[Y] + E[X]E[Y] \\ &= E[XY] - E[X]E[Y]\end{aligned}$$

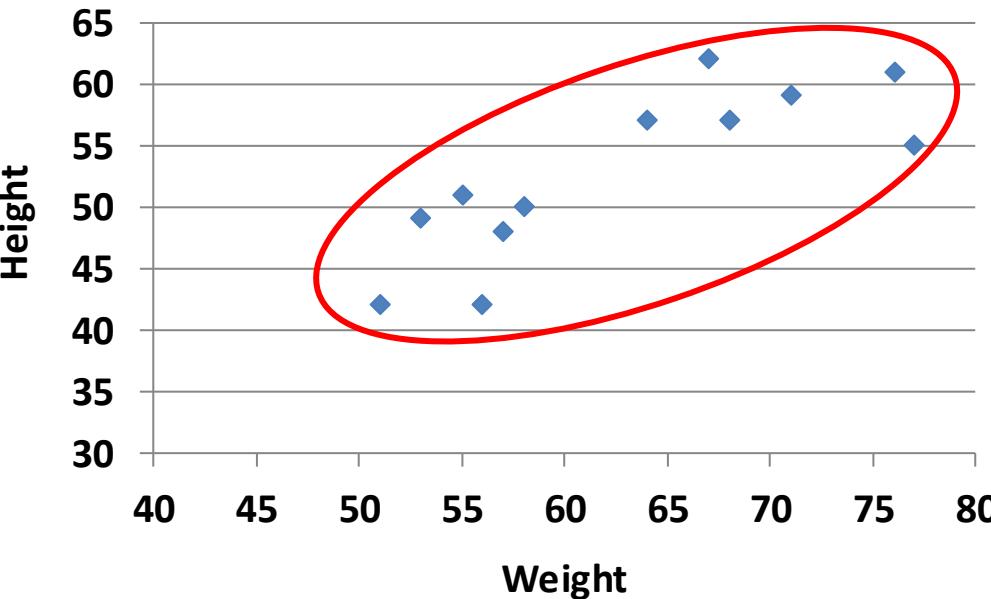
- X and Y independent $\rightarrow \text{Cov}(X, Y) = 0$
- But $\text{Cov}(X, Y) = 0$ does not imply X and Y independent!

Covariance and Data

Consider the following data:

Weight	Height	Weight * Height
64	57	3648
71	59	4189
53	49	2597
67	62	4154
55	51	2805
58	50	2900
77	55	4235
57	48	2736
56	42	2352
51	42	2142
76	61	4636
68	57	3876

$E[W]$	$E[H]$	$E[W*H]$
$= 62.75$	$= 52.75$	$= 3355.83$



$$\begin{aligned}\text{Cov}(W, H) &= \underline{E[W*H]} - \underline{E[W]E[H]} \\ &= 3355.83 - (62.75)(52.75) \\ &= 45.77\end{aligned}$$

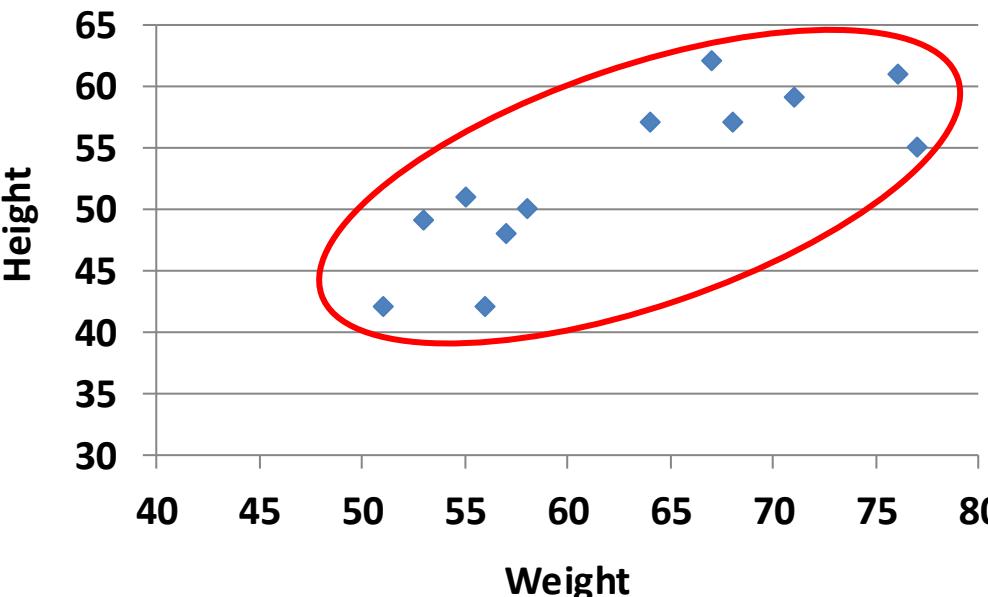
Correlation

What is Wrong With This?

Consider the following data:

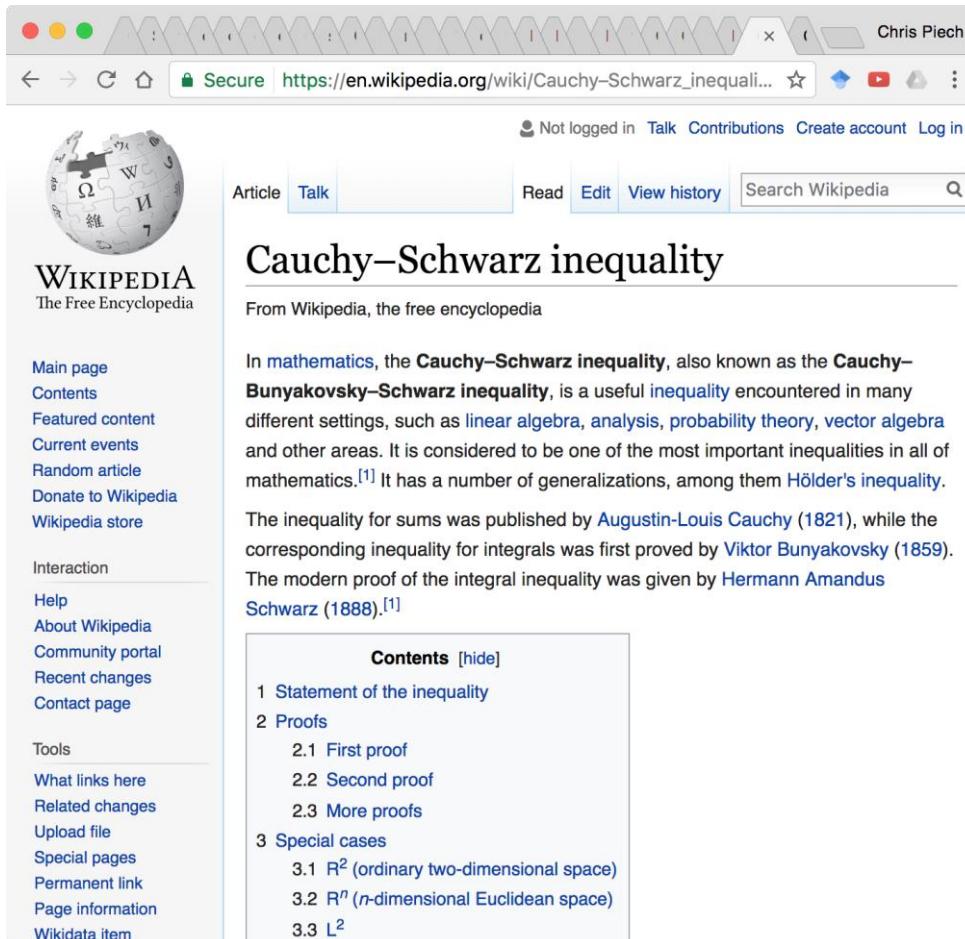
Weight	Height	Weight * Height
64	57	3648
71	59	4189
53	49	2597
67	62	4154
55	51	2805
58	50	2900
77	55	4235
57	48	2736
56	42	2352
51	42	2142
76	61	4636
68	57	3876

$E[W]$	$E[H]$	$E[W*H]$
$= 62.75$	$= 52.75$	$= 3355.83$



$$\begin{aligned}\text{Cov}(W, H) &= E[W*H] - E[W]E[H] \\ &= 3355.83 - (62.75)(52.75) \\ &= 45.77\end{aligned}$$

Cauchy Schwarz, a great way to normalize!



The screenshot shows a web browser window with the following details:

- Address Bar:** https://en.wikipedia.org/wiki/Cauchy-Schwarz_inequality
- User Information:** Not logged in, Chris Piech
- Page Title:** Cauchy–Schwarz inequality
- Page Content Summary:** From Wikipedia, the free encyclopedia
- Text Content:** In mathematics, the Cauchy–Schwarz inequality, also known as the Cauchy–Bunyakovsky–Schwarz inequality, is a useful inequality encountered in many different settings, such as linear algebra, analysis, probability theory, vector algebra and other areas. It is considered to be one of the most important inequalities in all of mathematics.^[1] It has a number of generalizations, among them Hölder's inequality.
- Text Content (continued):** The inequality for sums was published by Augustin-Louis Cauchy (1821), while the corresponding inequality for integrals was first proved by Viktor Bunyakovsky (1859). The modern proof of the integral inequality was given by Hermann Amandus Schwarz (1888).^[1]
- Table of Contents:**
 - 1 Statement of the inequality
 - 2 Proofs
 - 2.1 First proof
 - 2.2 Second proof
 - 2.3 More proofs
 - 3 Special cases
 - 3.1 \mathbb{R}^2 (ordinary two-dimensional space)
 - 3.2 \mathbb{R}^n (n -dimensional Euclidean space)
 - 3.3 L^2

$$-\text{Std}(X)\text{Std}(Y) \leq \text{Cov}(X, Y) \leq \text{Std}(X)\text{Std}(Y)$$

Viva La Correlatióñ

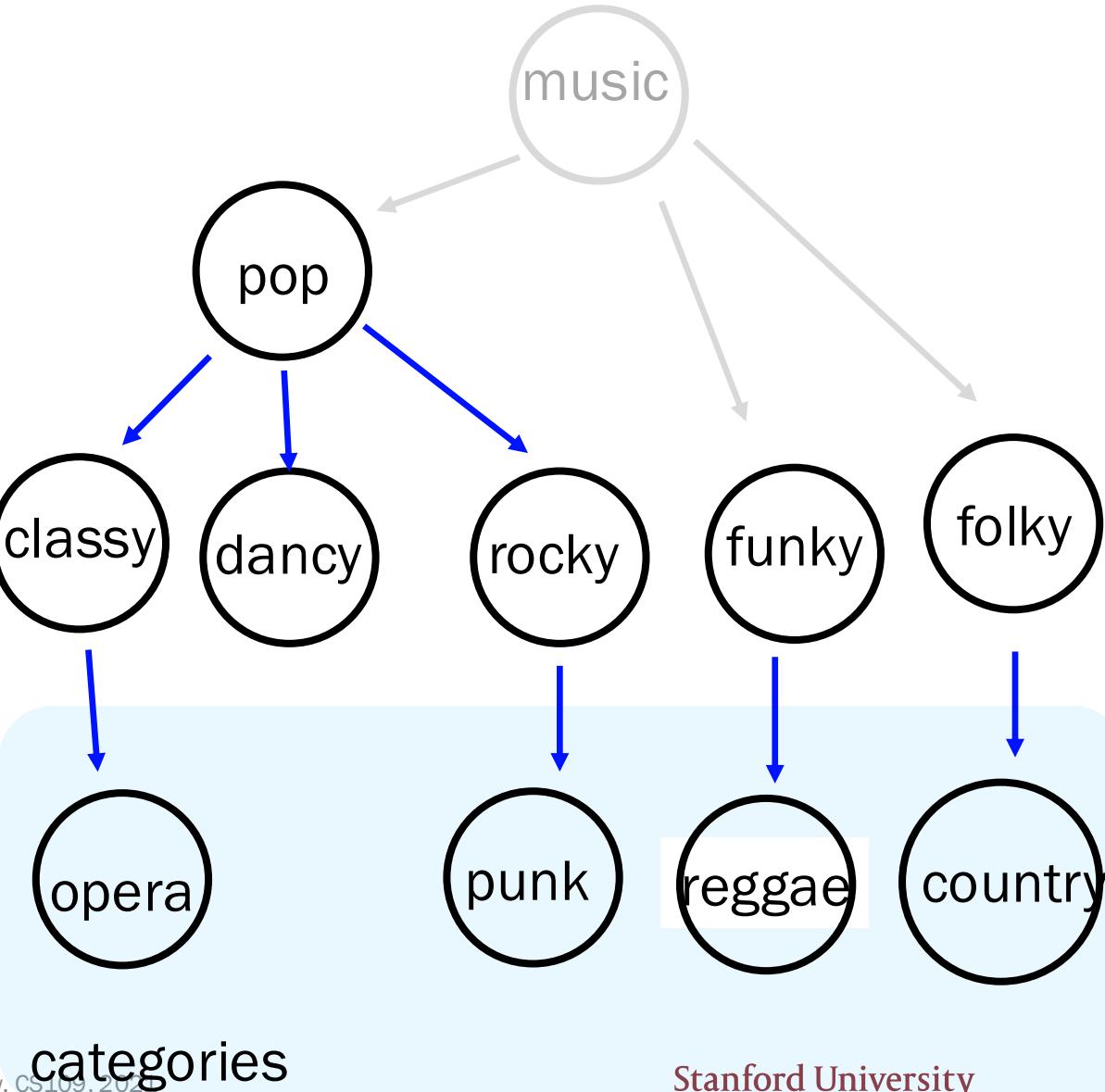
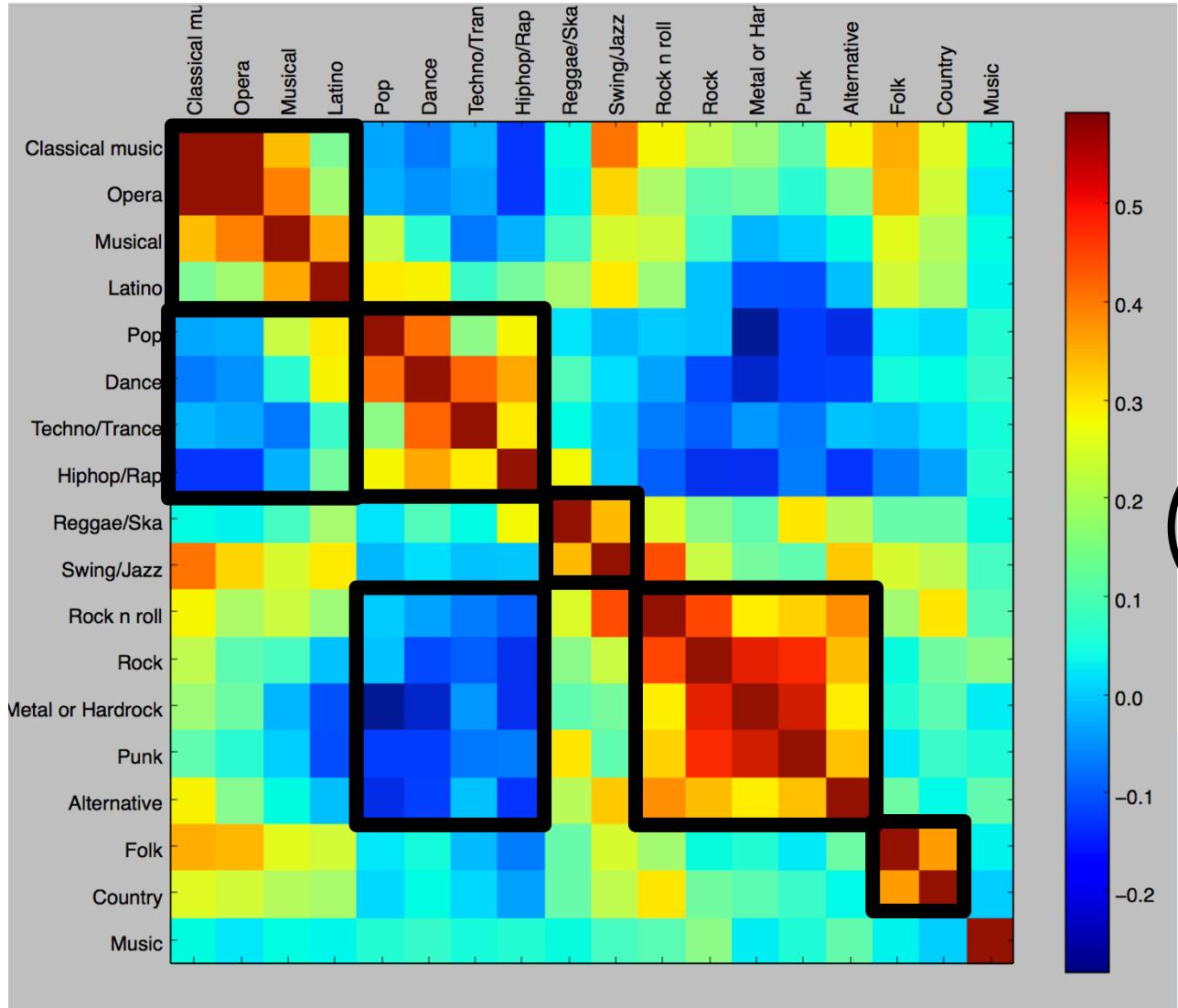
Say X and Y are arbitrary random variables

- Correlation of X and Y , denoted $\rho(X, Y)$:

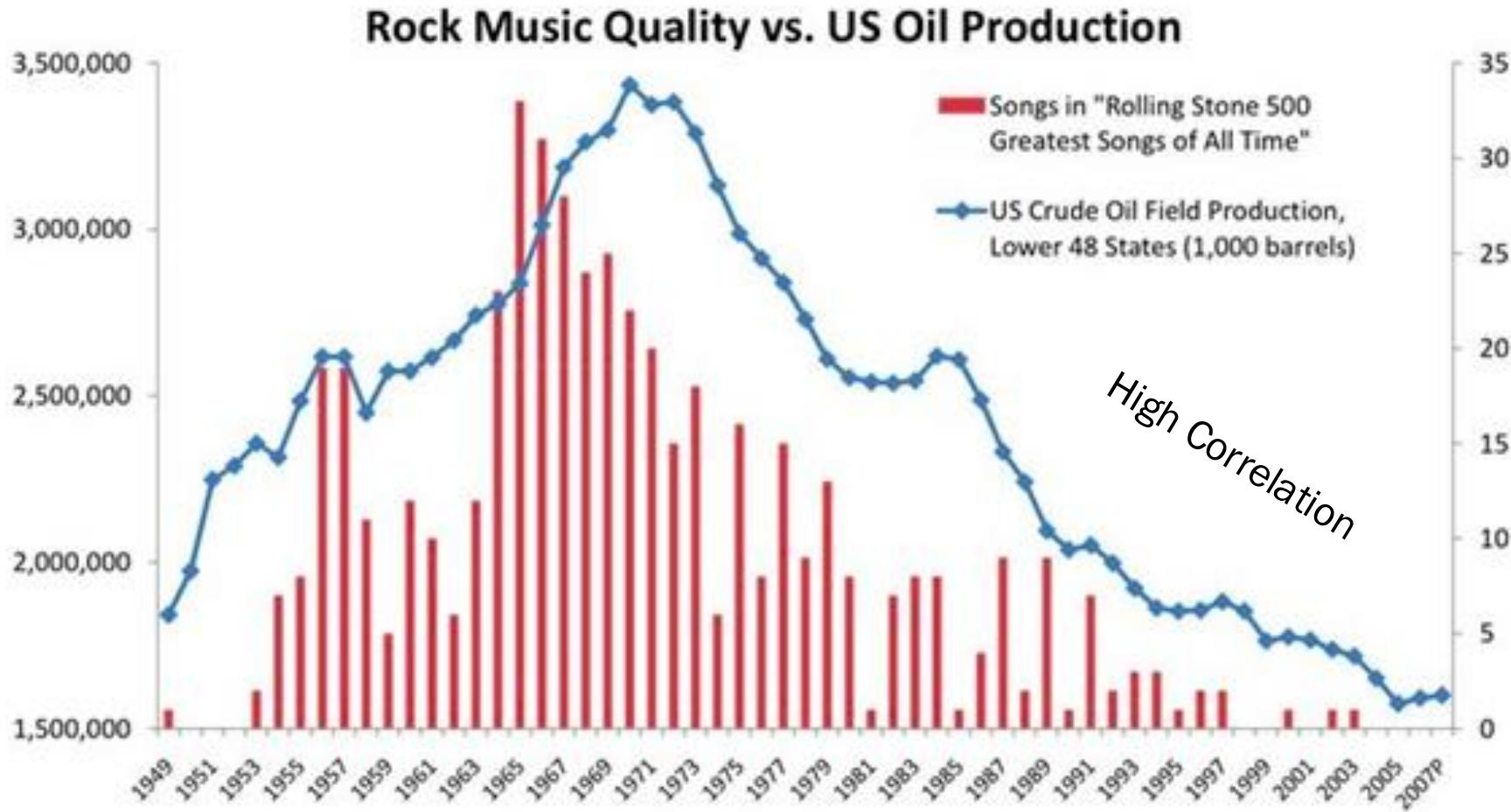
$$\underline{\rho}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

- Note: $-1 \leq \rho(X, Y) \leq 1$
- $\rho(X, Y) = 1 \Rightarrow$ perfectly correlated
- $\rho(X, Y) = -1 \Rightarrow$ perfectly negatively correlated
- $\rho(X, Y) = 0 \Rightarrow$ absence of linear relationship
 - But, X and Y can still be related in some other way!
- If $\rho(X, Y) = 0$, we say X and Y are “uncorrelated”

Recall: It is a useful starting point

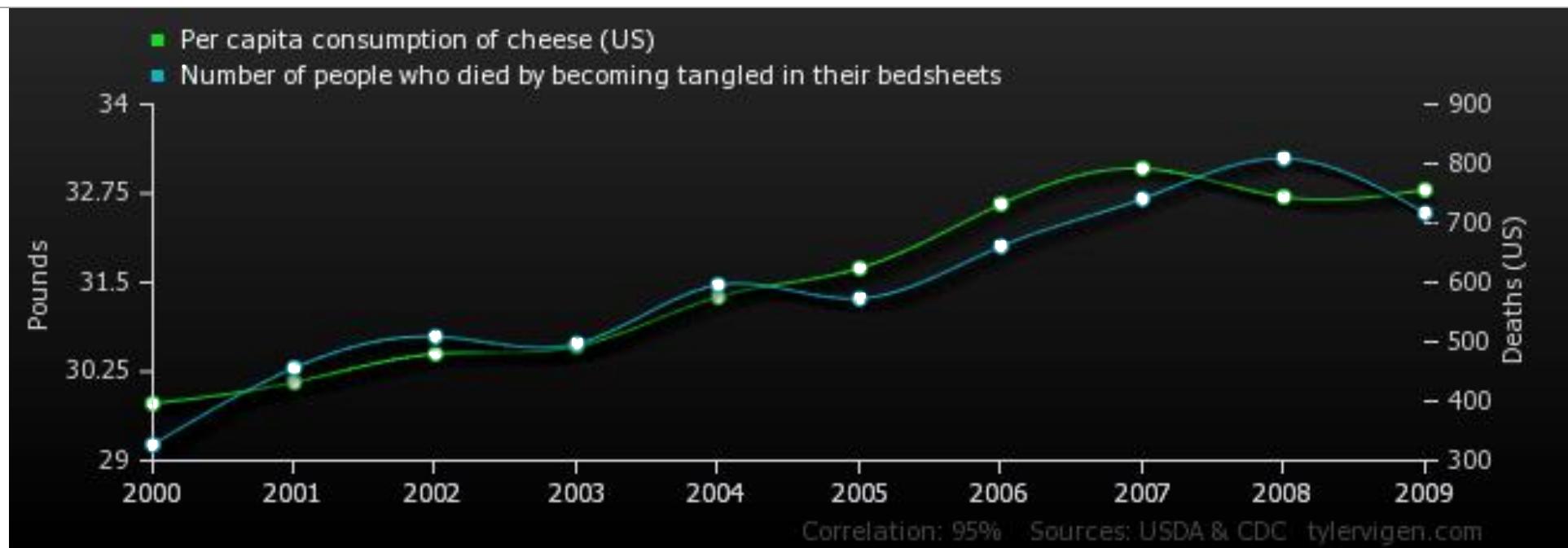


Rock Music Vs Oil?



Hubbert Peak Theory

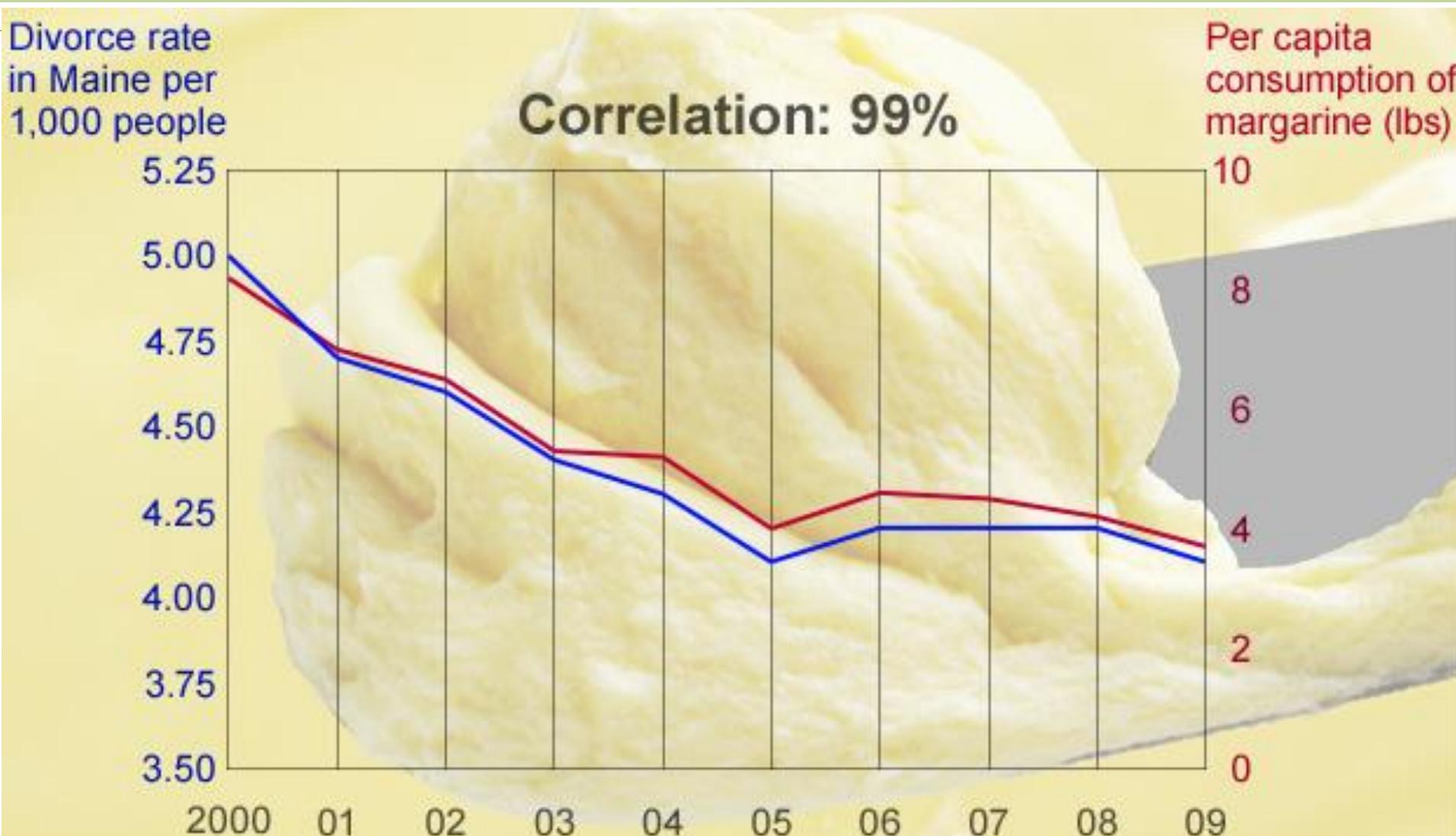
Tell your friends!



	<u>2000</u>	<u>2001</u>	<u>2002</u>	<u>2003</u>	<u>2004</u>	<u>2005</u>	<u>2006</u>	<u>2007</u>	<u>2008</u>	<u>2009</u>
<i>Per capita consumption of cheese (US)</i> Pounds (USDA)	29.8	30.1	30.5	30.6	31.3	31.7	32.6	33.1	32.7	32.8
<i>Number of people who died by becoming tangled in their bedsheets</i> Deaths (US) (CDC)	327	456	509	497	596	573	661	741	809	717

Correlation: 0.947091

Divorce Vs Butter?



Source: US Census, USDA, tylervigen.com

SPL

Three Guiding Questions

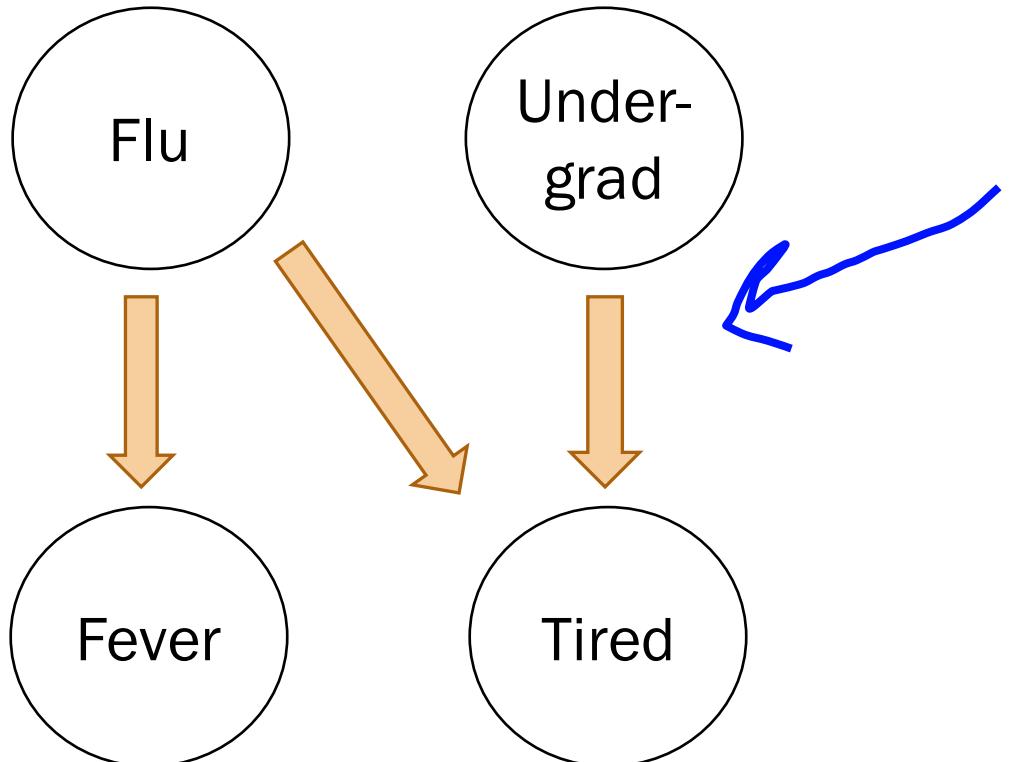
1. How do people actually define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

What haven't we talked about?

Machine Learning (last section of CS109)

$$P(F_{lu} = 1) = 0.1$$

$$P(U = 1) = 0.8$$



$$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$$
$$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$$

$$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$$
$$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$$
$$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$$
$$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$$

1. Learn this from data

2. Learn this from data