

General Inference

Chris Piech
CS109, Stanford University

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



SEARCH FILTER CLINIC

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

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?

At this point you know inference with
two random variables

Today: Five New Real + Exciting Problems

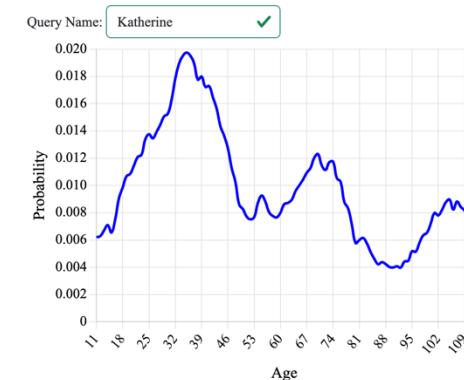
Age from C14



Updated Delivery Prob



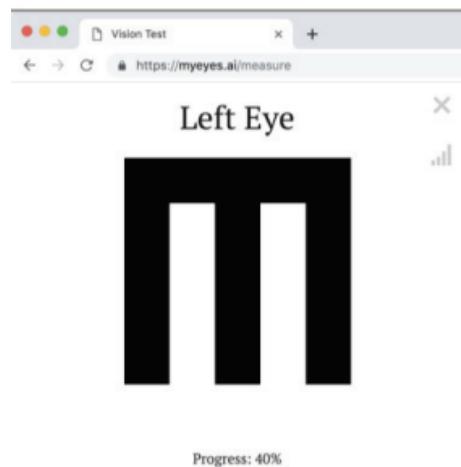
Age from Name



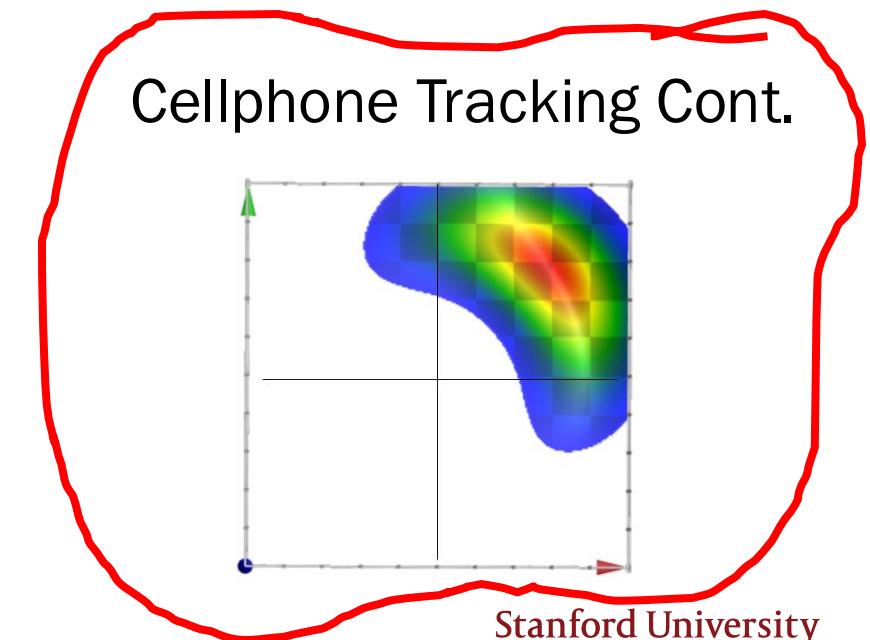
Hidden Chambers



Stanford Eye Test



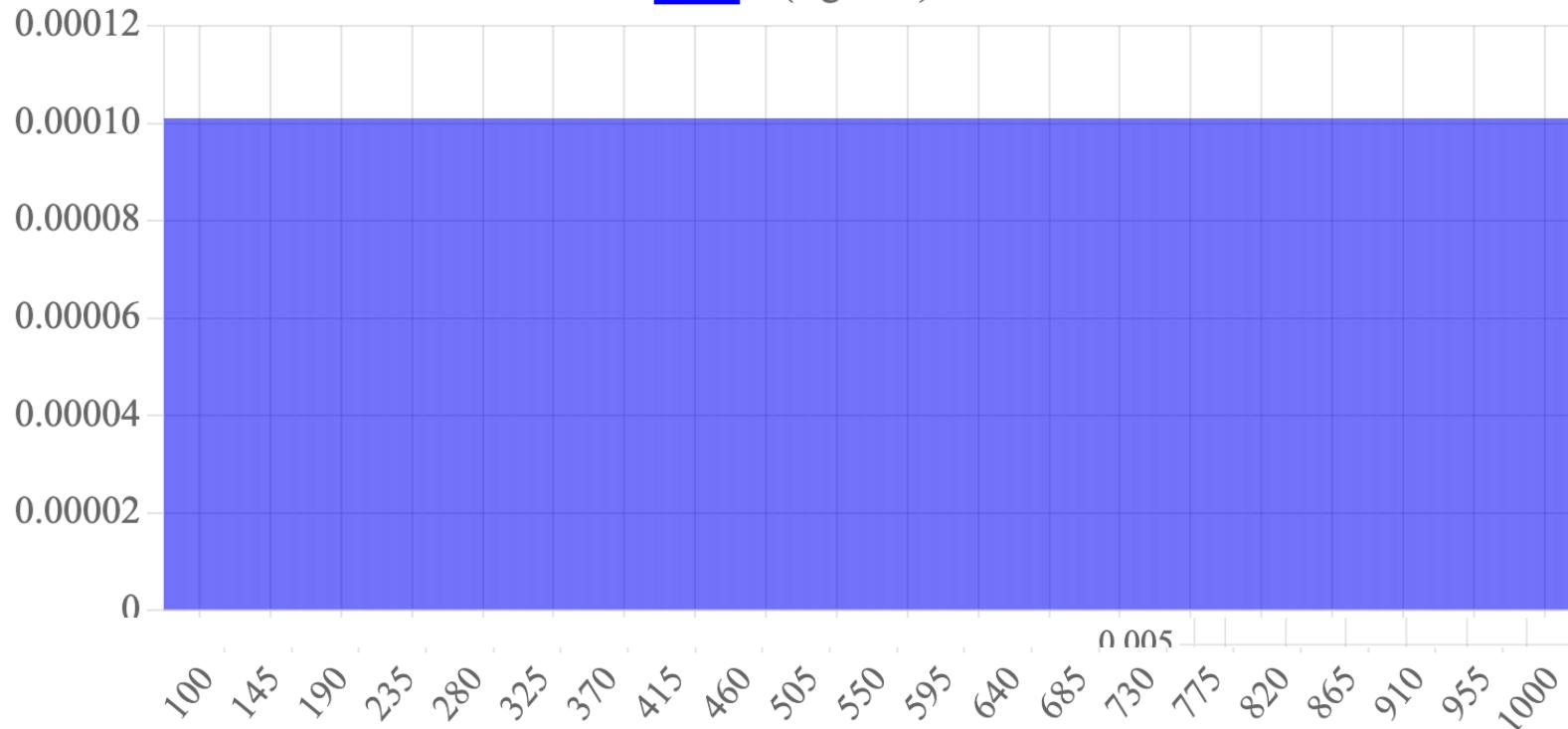
Cellphone Tracking Cont.



Update Belief PMF

Before observation

P(Age = x)



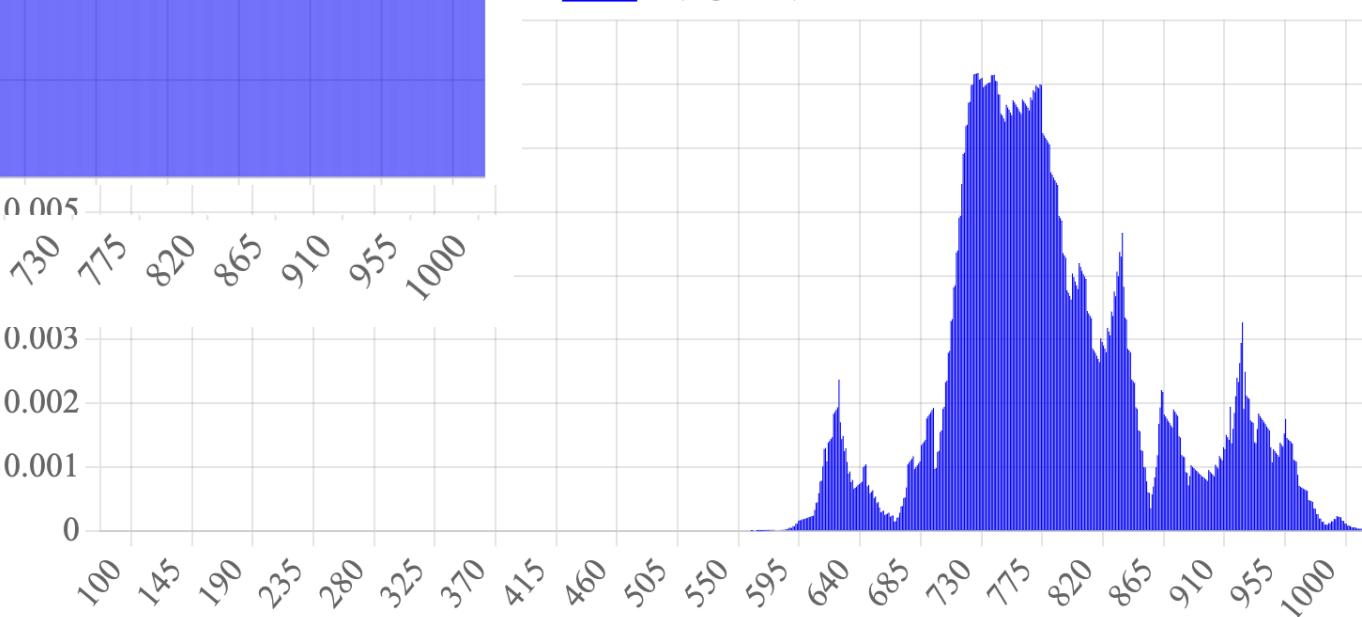
Observation

Remaining C14:

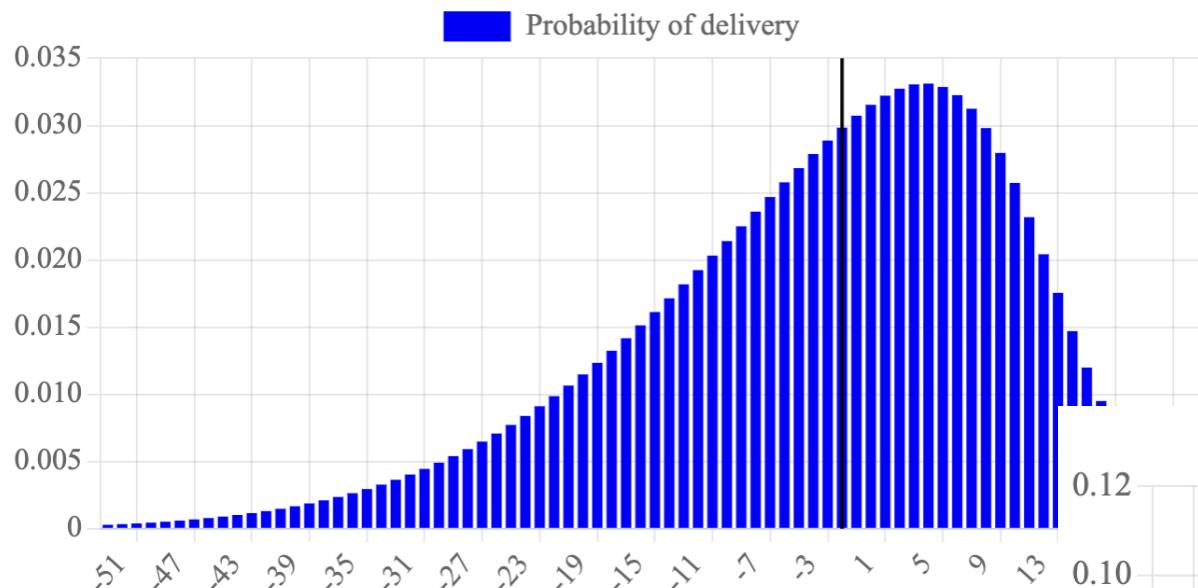
900

P(Age = x)

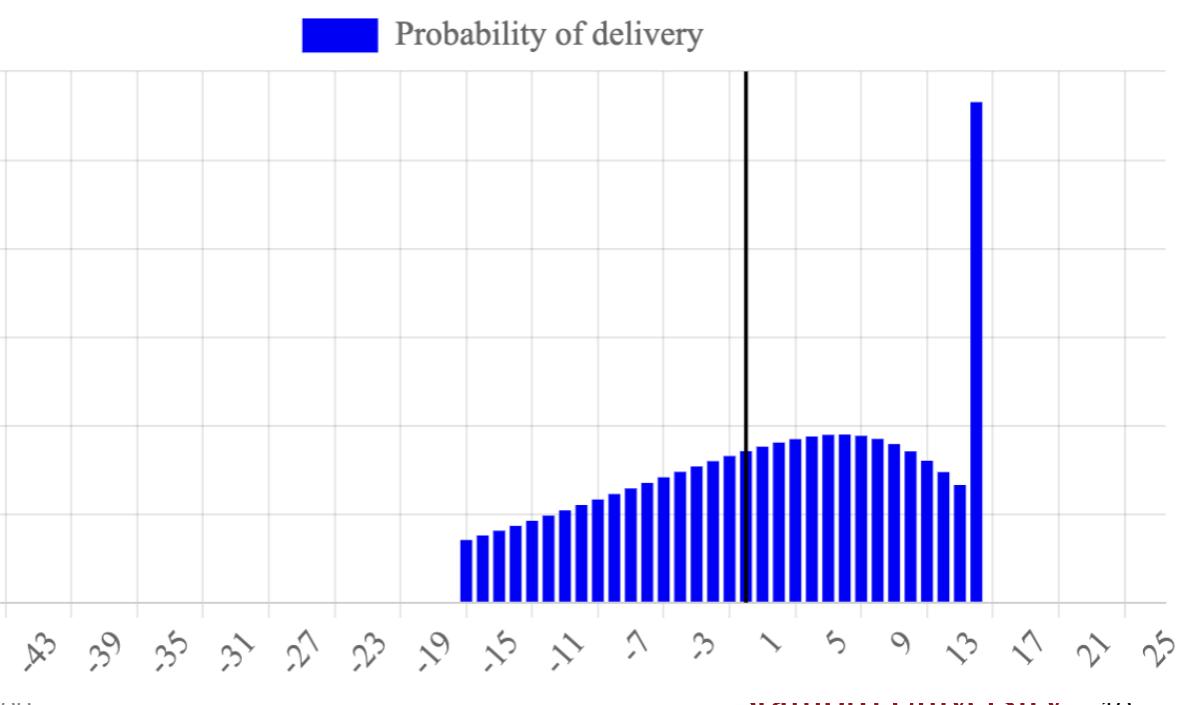
After



Baby delivery



Its 19 days until the due date and no baby



For each value d :

$$P(D = d | \text{no child so far})$$

$$= \frac{P(\text{no child so far} | D = d) P(D = d)}{P(\text{no child so far})}$$

```
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_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?

Normalize does so much for you!!!

Normalize: scale each value so that they would sum to 1

```
normalize({  
    'cat' : 5,  
    'dog' : 10,  
    'axolotyl' : 5  
})
```



```
{  
    'cat' : 0.2,  
    'dog' : 0.4,  
    'axolotyl' : 0.2  
}
```

Hidden Pyramid Chambers with Poisson + Bayes



Hidden Pyramid Chambers with Poisson + Bayes

Number of Muons

$$f(X = x|M = 12) = \frac{P(M = 12|X = x)f(X = x)}{P(M = 12)}$$



Amount of limestone

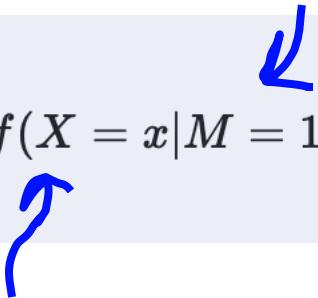
Denominator option #1: Law of total probability

$$P(M = 12) = \int_0^{100} P(M = 12|X = x)f(X = x)dx$$

Hidden Pyramid Chambers with Poisson + Bayes

Number of Muons

$$f(X = x|M = 12) = \frac{P(M = 12|X = x)f(X = x)}{P(M = 12)}$$



Amount of limestone

Denominator option #2: Solve for K

$$\begin{aligned} f(X = x|M = 12) &= \frac{P(M = 12|X = x)f(X = x)}{P(M = 12)} \\ &= \frac{\left(\frac{100 \cdot e^{-x/40}}{12!}\right)^{12} e^{-\left(\frac{100 \cdot e^{-x/40}}{40}\right)}}{P(M = 12)} \cdot \frac{1}{100} \propto e^{-\frac{12x}{40} - 100 \cdot e^{-x/40}} \end{aligned}$$

FunTip: Integration via Sampling

Teachers hate him!

See how you (approximate) the
integral of any function with
one simple trick.

ProTip: Integration via Sampling

6 Goodbye integral, my old friend [14 points]

Fall 2017

In this problem we are going to compute probabilities for a random variable X that can takes on values in the range $0 \leq x \leq 1$, and has probability density function:

$$f_X(x) = \frac{1}{K} \cdot g(x)$$

Where $g(x)$ is some terribly nasty and non-integrable function. For your sanity I won't even write out g . It is that bad. In such a situation, we can turn to the power of computers to help us (you may assume that while g is impossible to integrate, you were able to code g as a function in a programming language). The key idea that we are going to use is called Monte Carlo Integration:

Generate N values (X_1, X_2, \dots, X_N) uniformly sampled over a range (a, b) . We can approximate the integral of a function h over (a, b) as:

$$\int_a^b h(x) dx \approx \frac{(b-a)}{N} \sum_{i=1}^N h(X_i)$$

Pretty amazing! Why did we bother with integrals at all? This question requires you to write pseudo code. Such code does not have to compile, but it should be specific enough that a knowledgeable programmer could implement what you have described. You may use a function `random(a, b)` which returns a sample from $X \sim \text{Uni}(a, b)$. You may also use a function $g(x)$ which is the hard-to-integrate term in $f_X(x)$.

Today: Five New Real + Exciting Problems

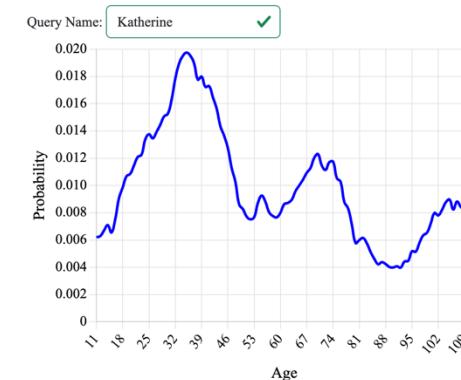
Age from C14



Updated Delivery Prob



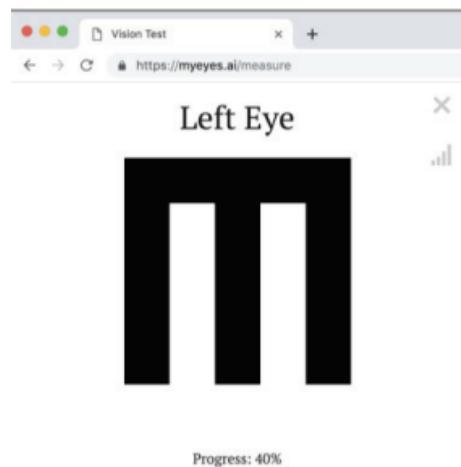
Age from Name



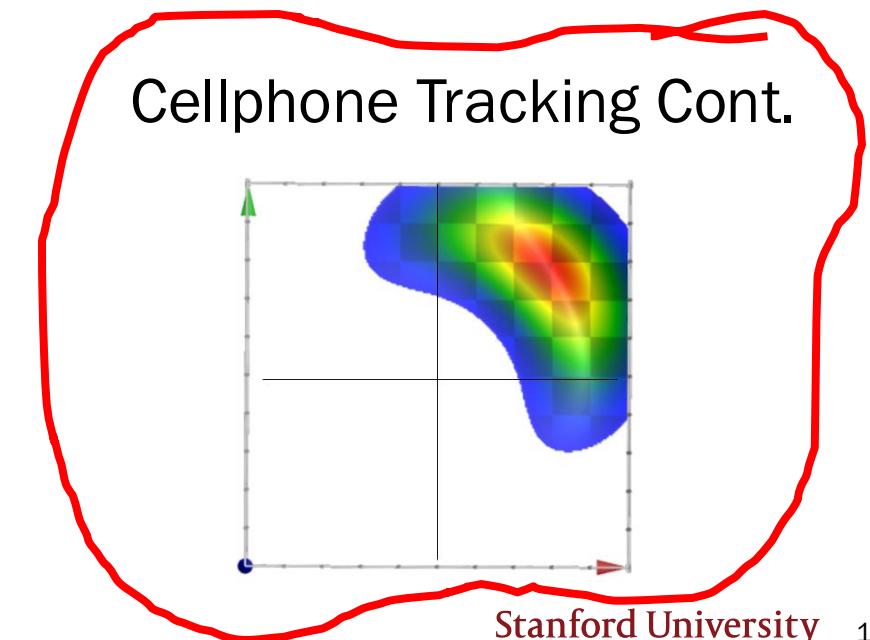
Hidden Chambers



Stanford Eye Test



Cellphone Tracking Cont.



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



SEARCH FILTER CANCEL

The screenshot shows the WebMD Symptom Checker interface. At the top, the title "WebMD Symptom Checker WITH BODY MAP" is displayed. Below the title is a navigation bar with tabs: INFO, SYMPTOMS (which is highlighted with a teal underline), QUESTIONS, CONDITIONS, DETAILS, and TREATMENT. A large, light gray 3D model of a human torso with arms extended is centered below the tabs. To the left of the torso, there is a text input field with the placeholder "Type your main symptom here". To the right of the torso, there are three small rectangular buttons labeled "SEARCH", "FILTER", and "CANCEL".

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**

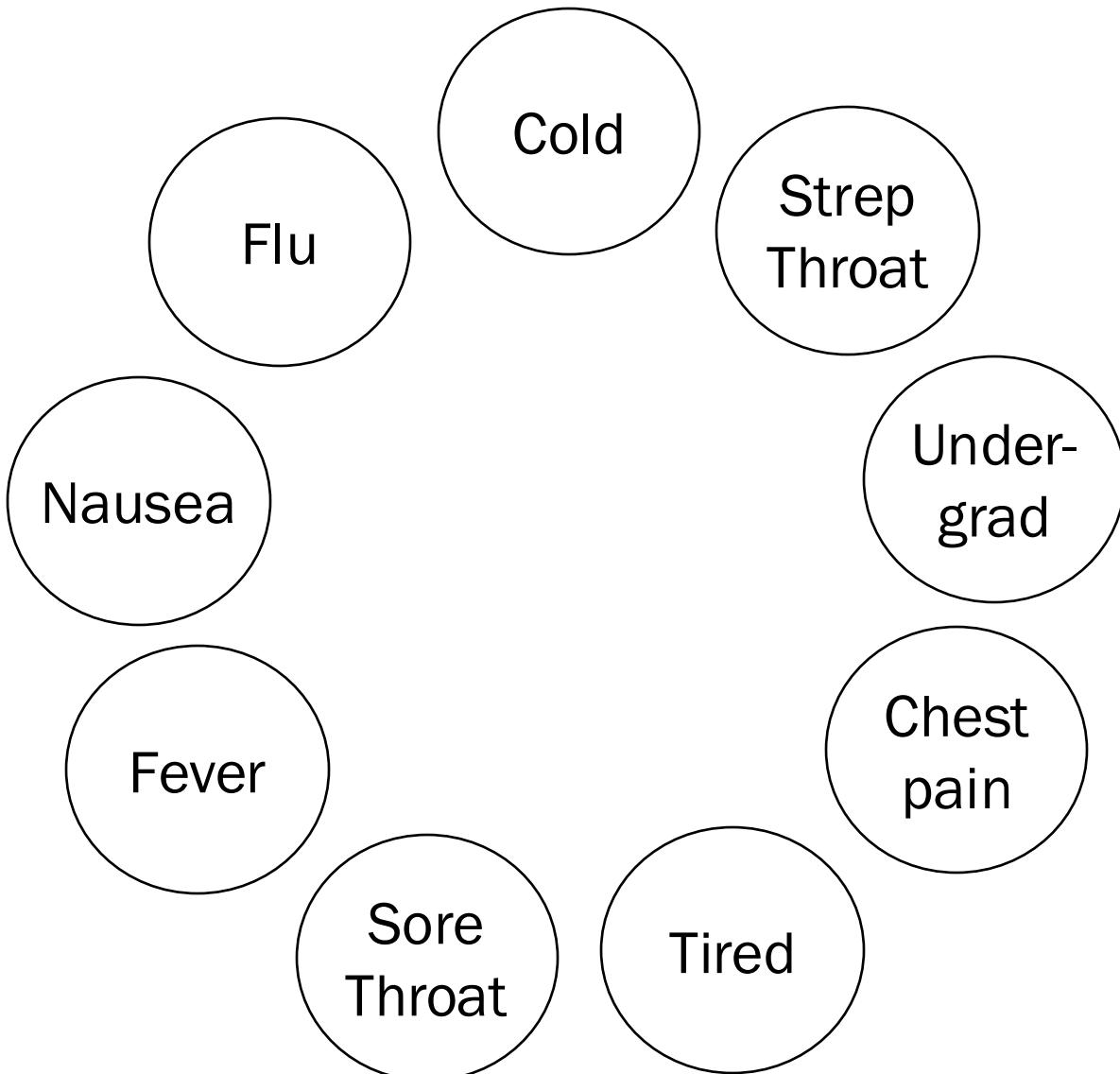
[Edit](#)

My Symptoms

[Edit](#)

dizziness, one sided headache

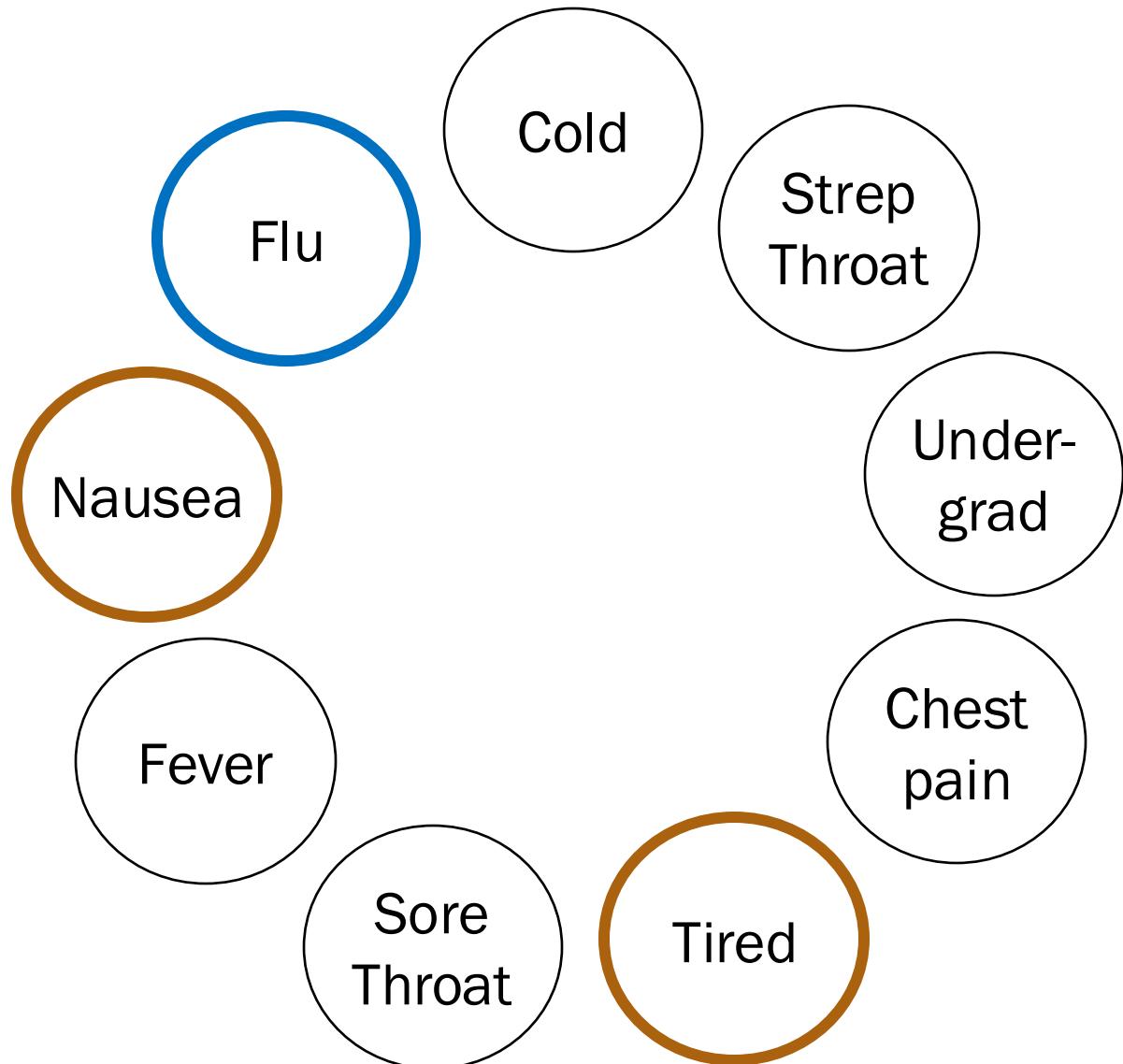
Challenge #1: Many Inference Questions



Inference question:

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

Challenge #1: Many Inference Questions

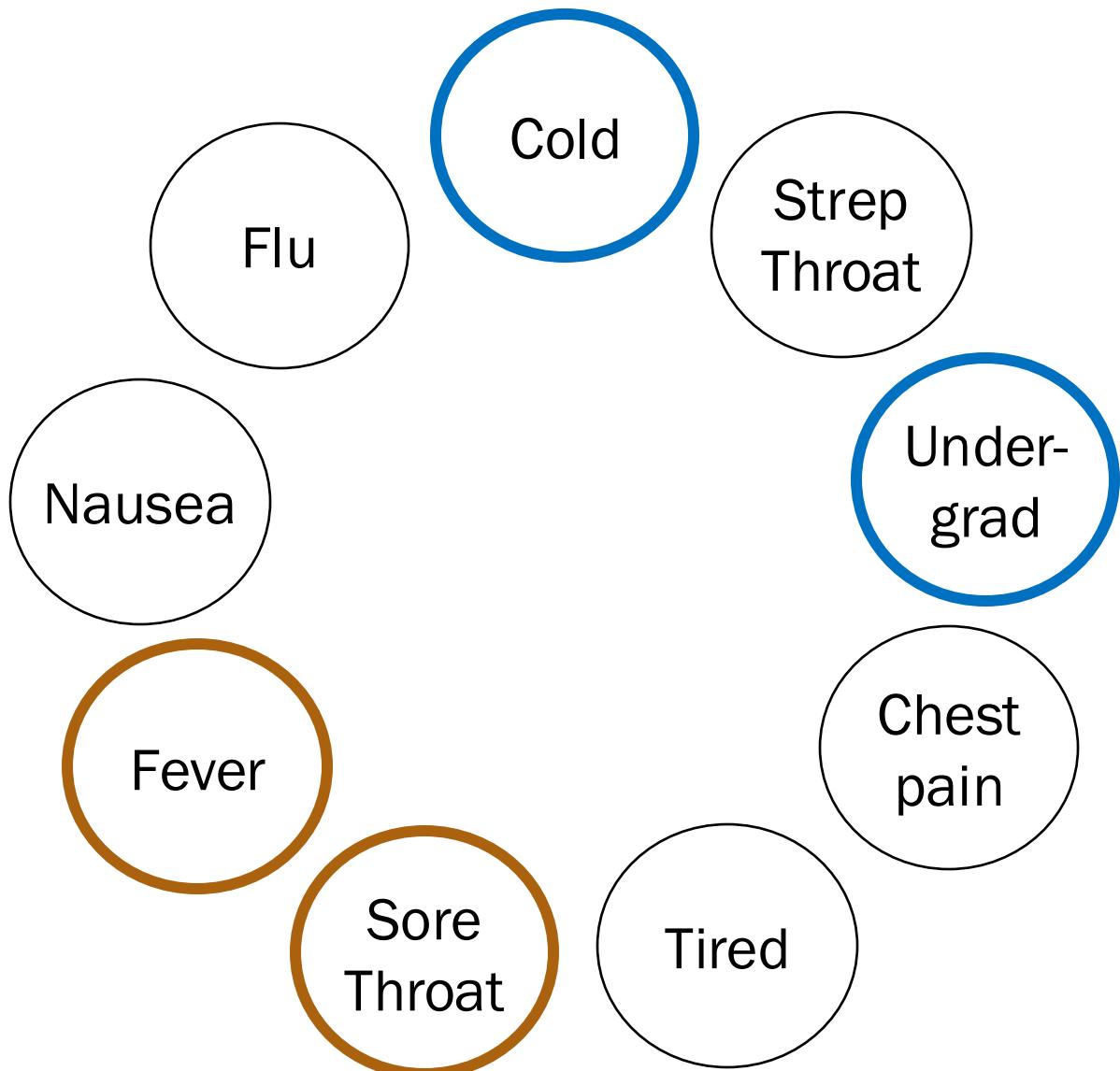


One inference question:

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

$$= \frac{P(F = 1, N = 1, T = 1)}{P(N = 1, T = 1)}$$

Challenge #1: Many Inference Questions

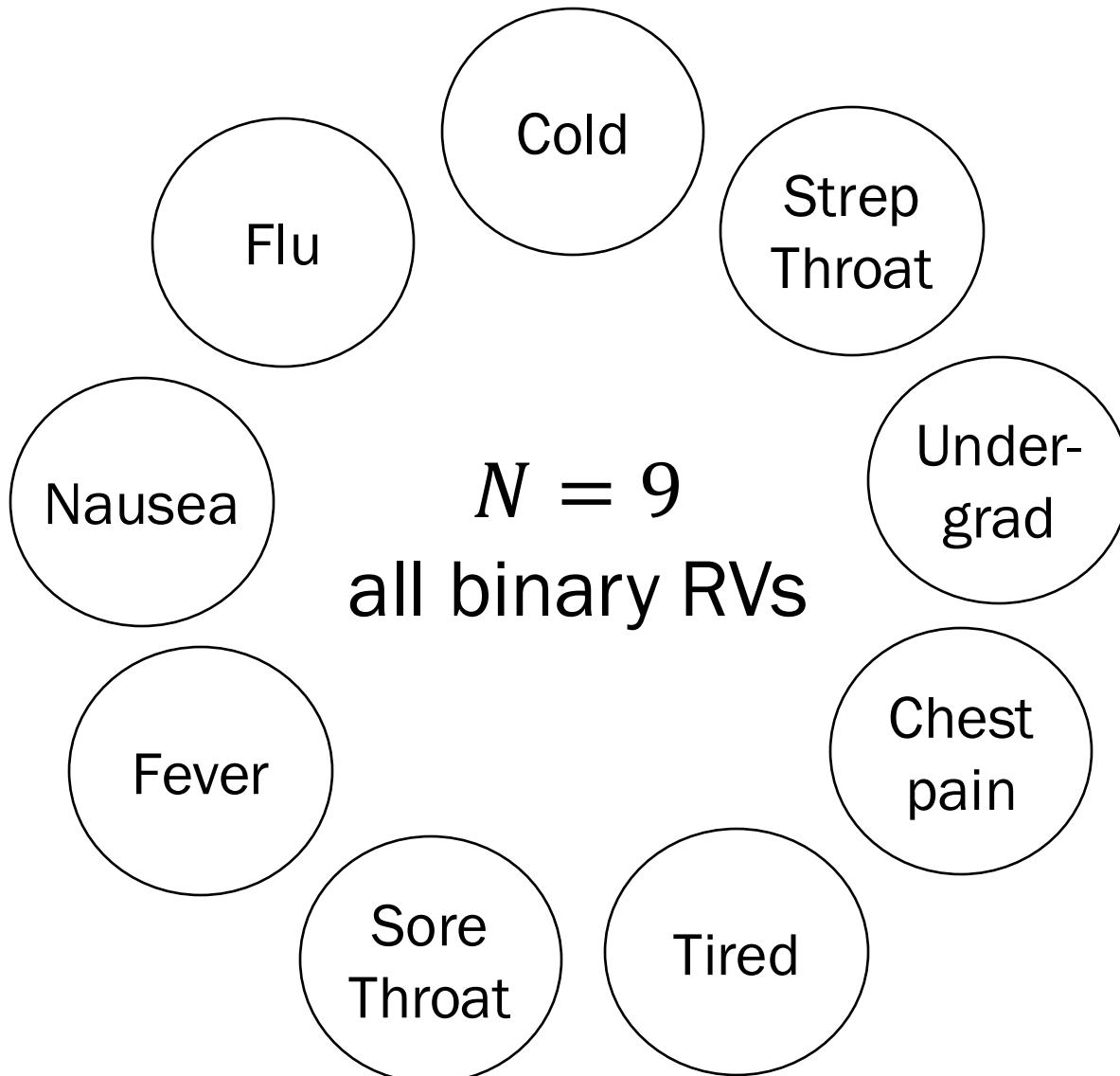


Another inference question:

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

$$= \frac{P(C_o = 1, U = 1, S = 0, F_e = 0)}{P(S = 0, F_e = 0)}$$

Challenge #2: Joint is Large



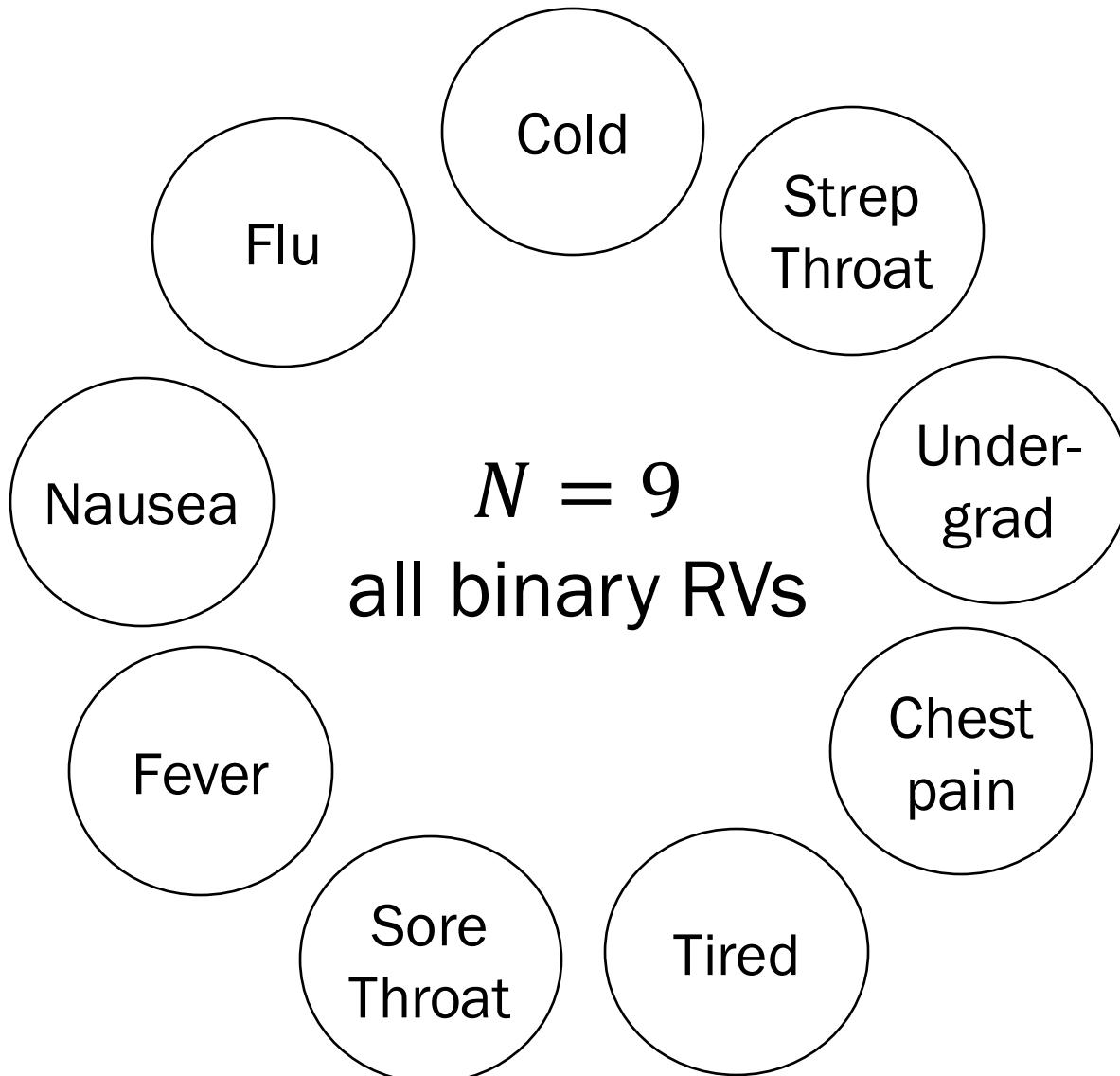
If we knew the **joint distribution**, we can answer all probabilistic inference questions.

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



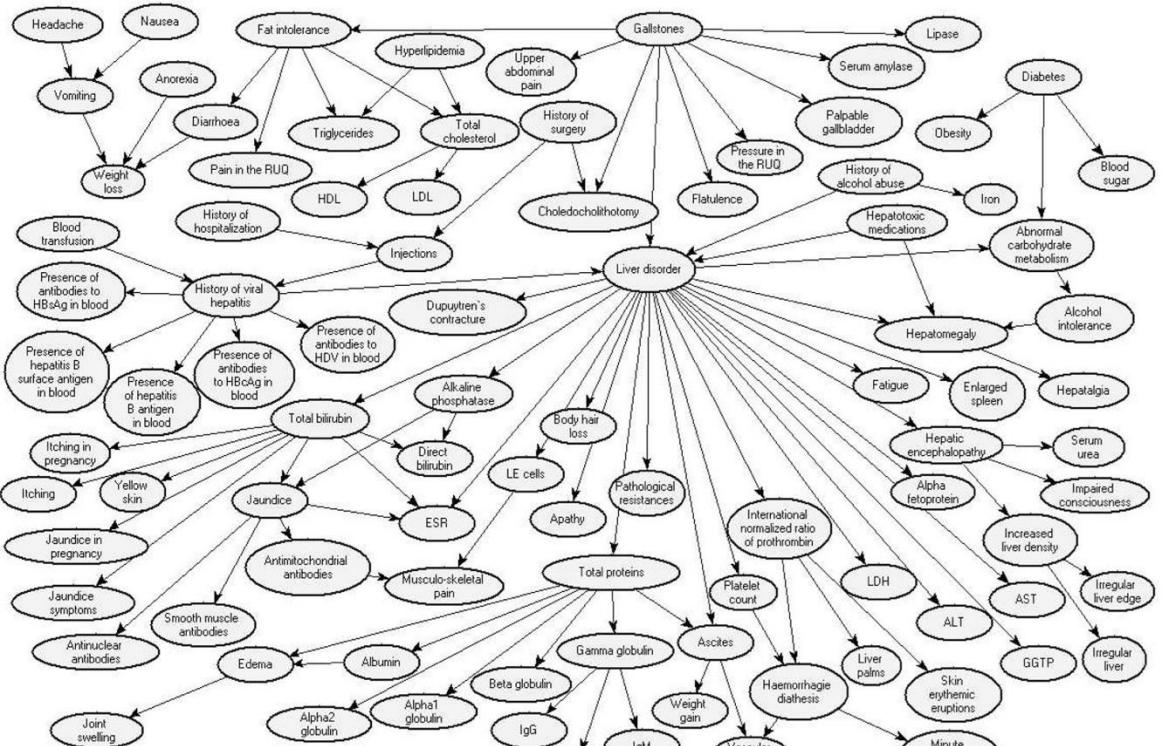
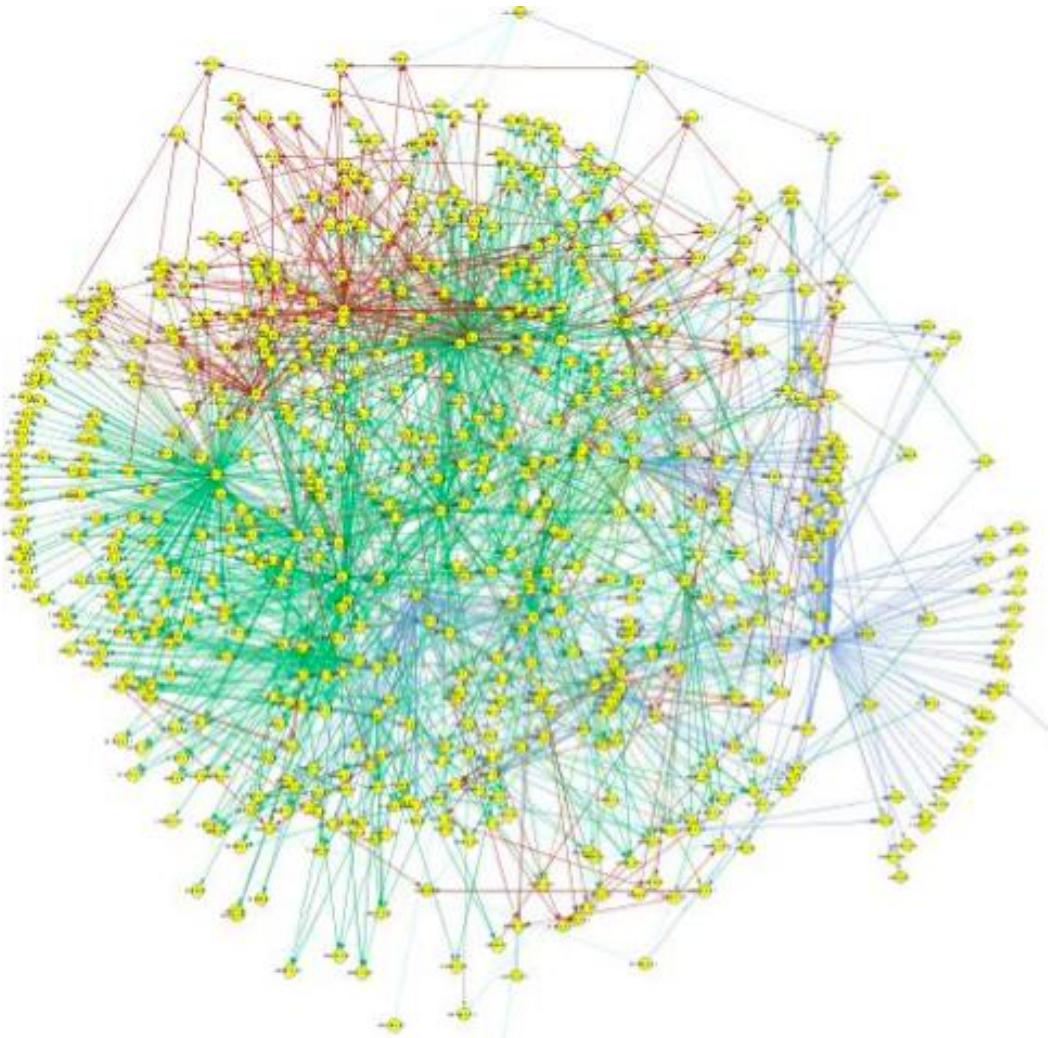
If we knew the **joint distribution**, we can answer all probabilistic inference questions.

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

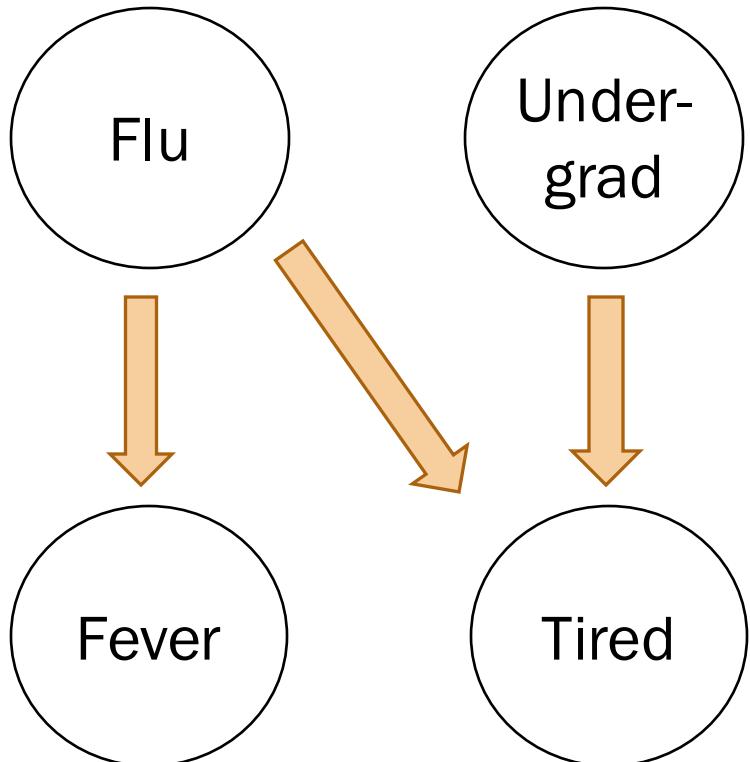
Tired

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

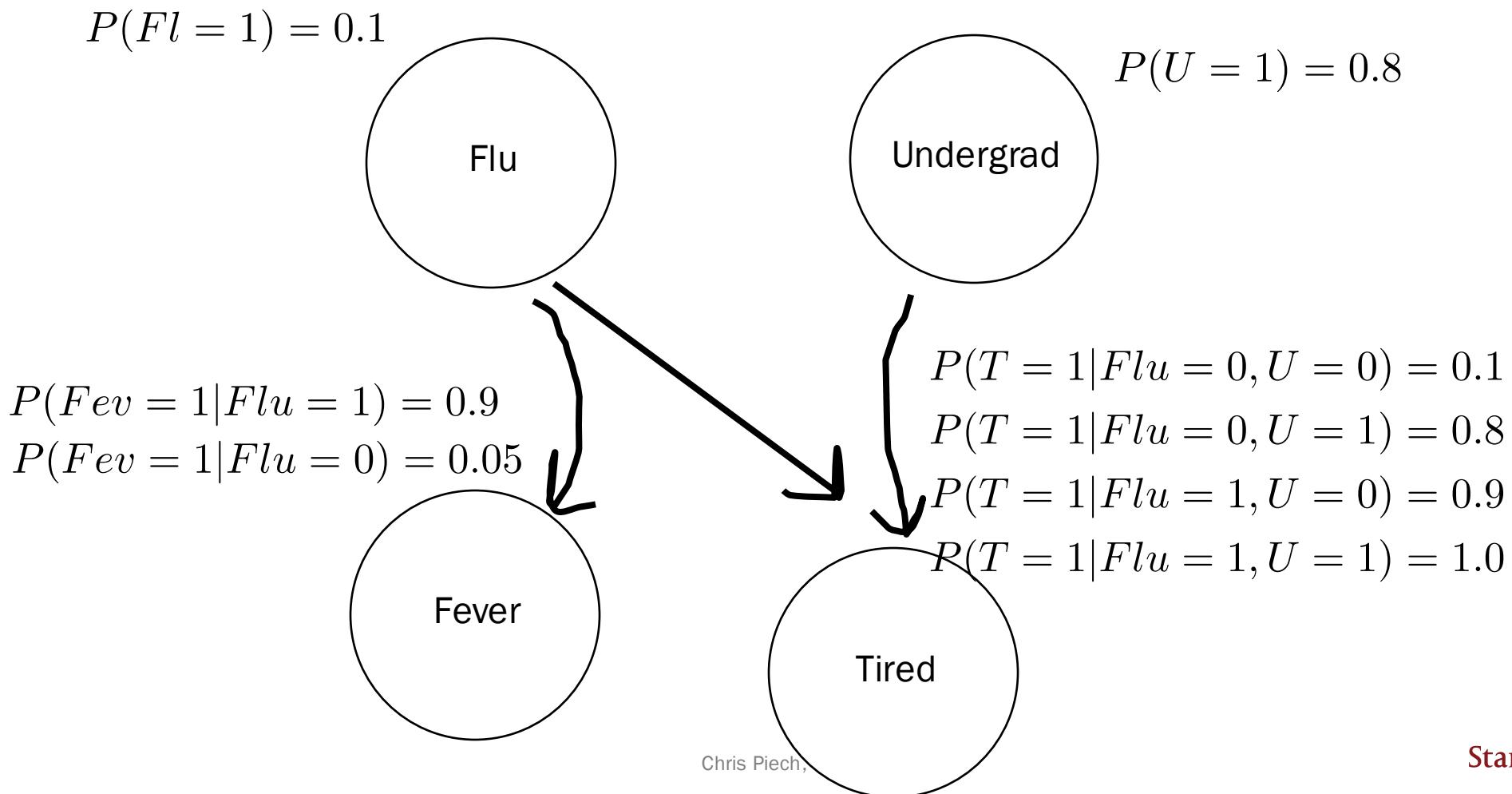


What would a Stanford flu expert do?

- 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



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

To the Code



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

```
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 (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 flue
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 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    }
```

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



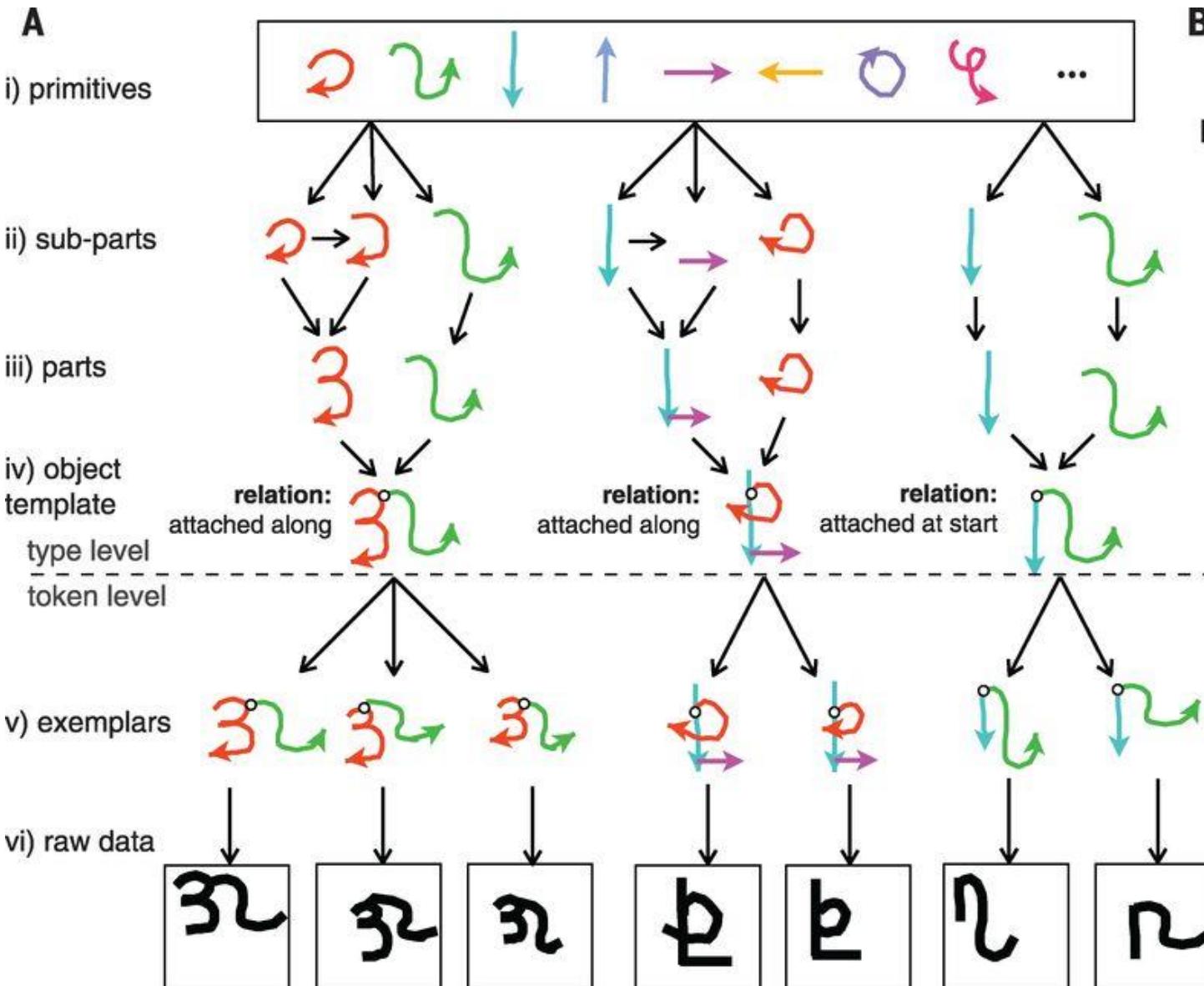
A good probabilistic model is **generative**. It explains the process through which the joint is **created**.

Generative Model of Binomial Questions

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

- Left Window:** A file browser showing a directory structure under `__pycacie__`. The `questionExp.py` file is selected.
- Middle Window:** A code editor titled `questionExp.py — generateBinomial` showing Python code for generating binomial questions. It defines two classes: `DeclareExpTask` and `DeclareSubtleExpTask`. The `DeclareExpTask` class has a `renderCode` method that returns either an explicit or subtle template based on a choice. The `TEMPLATES` dictionary contains three entries: `'standard'`, `'v2'`, and `'v3'`, each with a template string and weight. The `DeclareSubtleExpTask` class has a `registerChoices` method and a `renderCode` method that uses a template from the `TEMPLATES` dictionary.
- Right Window:** A terminal window titled `generateBinomial — zsh — 85x43` showing generated probability questions and their answers. It includes three examples involving coin flips, bitcoin mining, and election votes, each with a mathematical formula for the probability.

Generative Model of Hand Written Letters



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

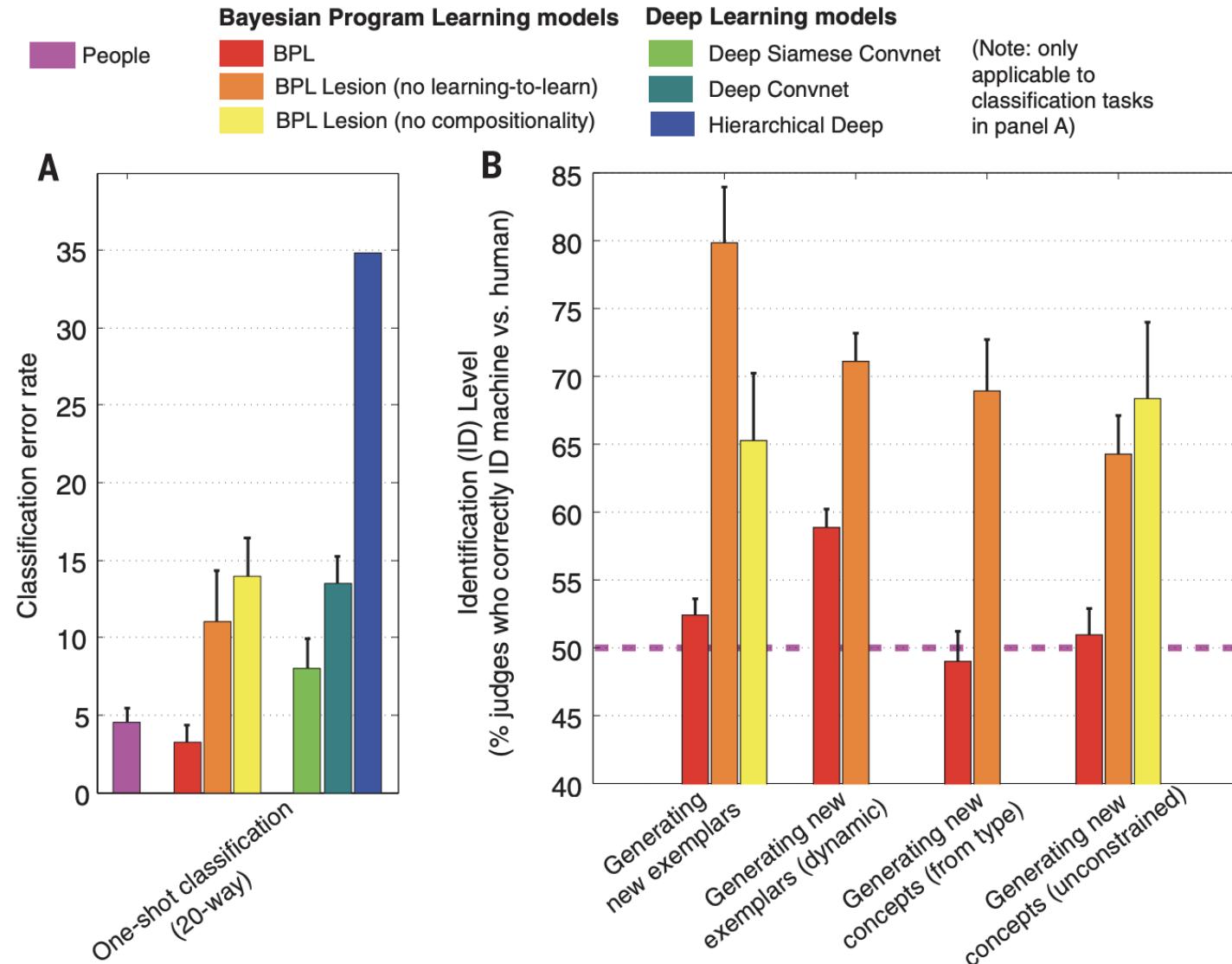
```

```

procedure GENERATETOKEN( $\psi$ )
    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  $I^{(m)}$ 

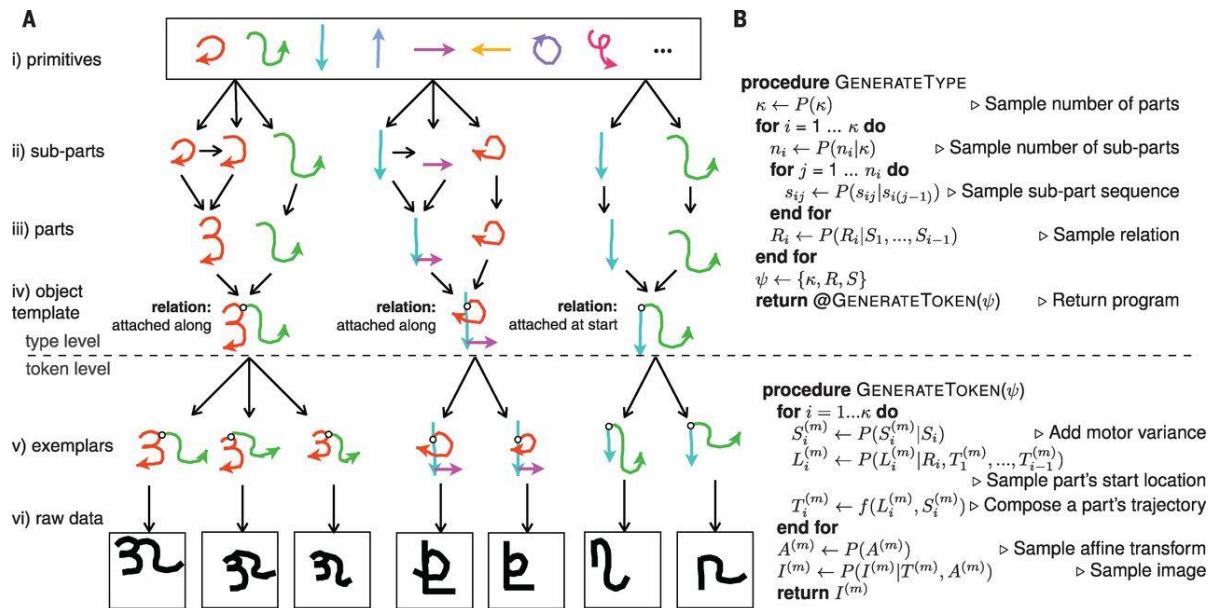
```

Human Level. And More!

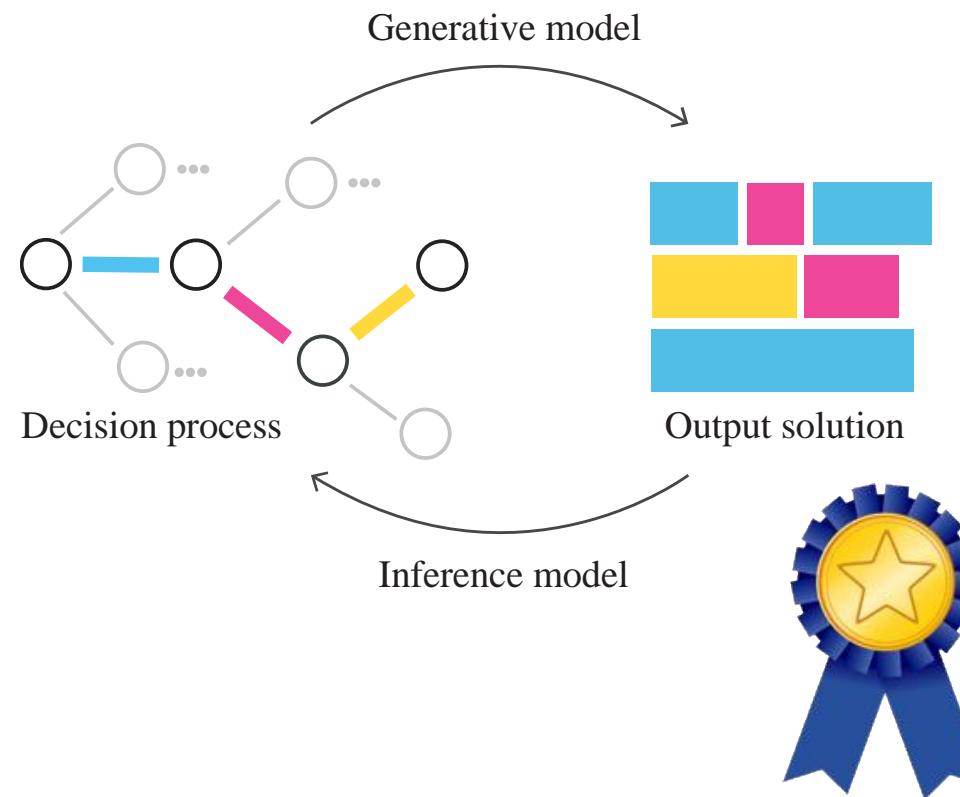


Generative Model of Grading

Lake et al, 2015

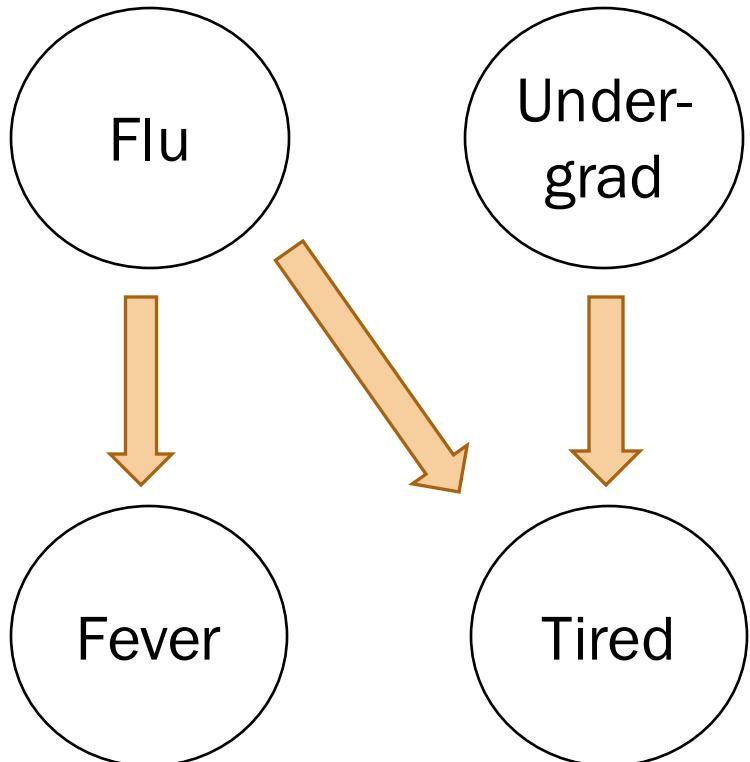


Muke Wu, Ali Malik, Noah Goodman, Chris Piech, 2019



*Outstanding
Paper Award, AAAI 2019*

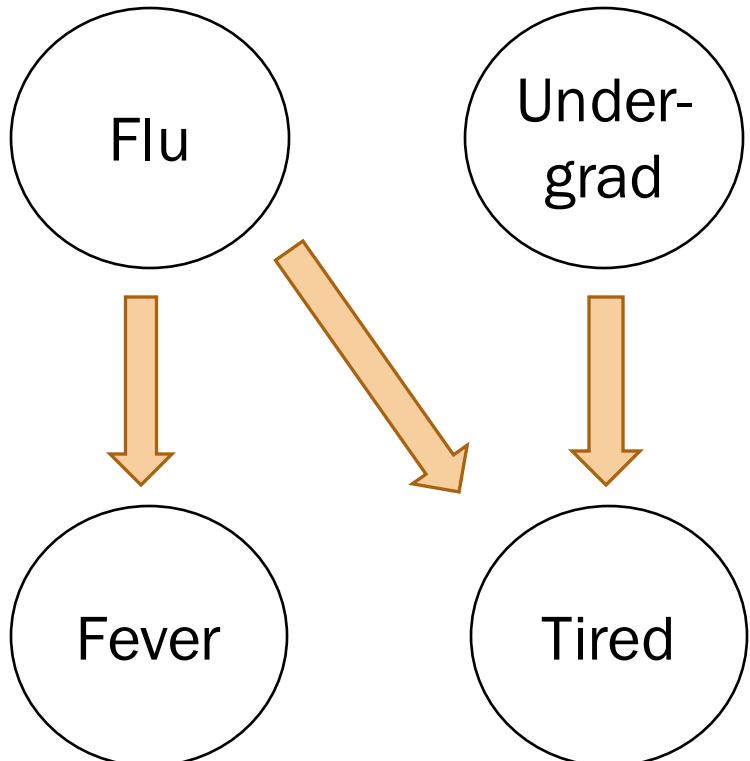
Generative Models make Independence Assumptions



What would a Stanford flu expert do?

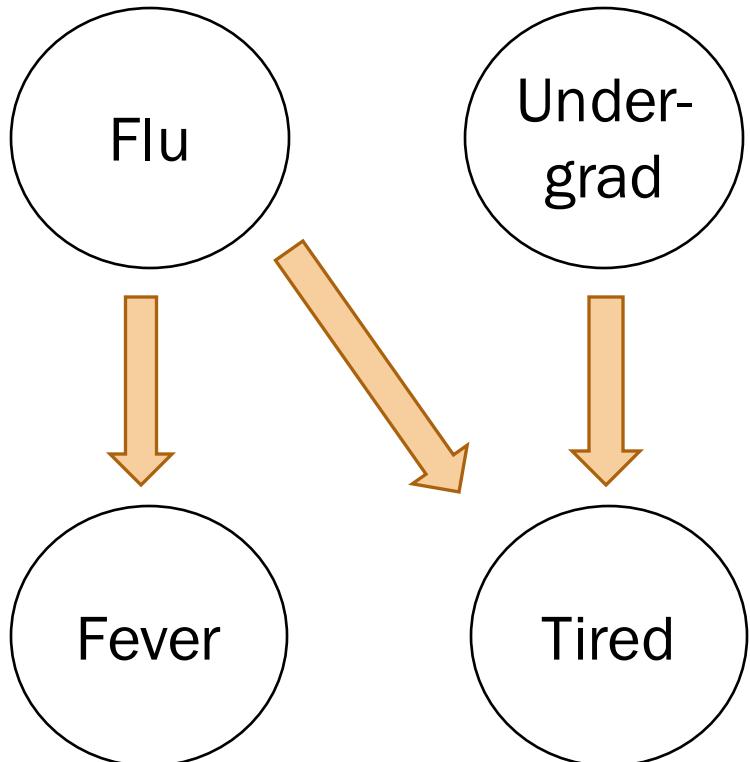
- 1. Describe the causality.
- 2. Provide $P(\text{values}|\text{causal parents})$ for each random variable
- 3. Implicitly assumes independences.

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

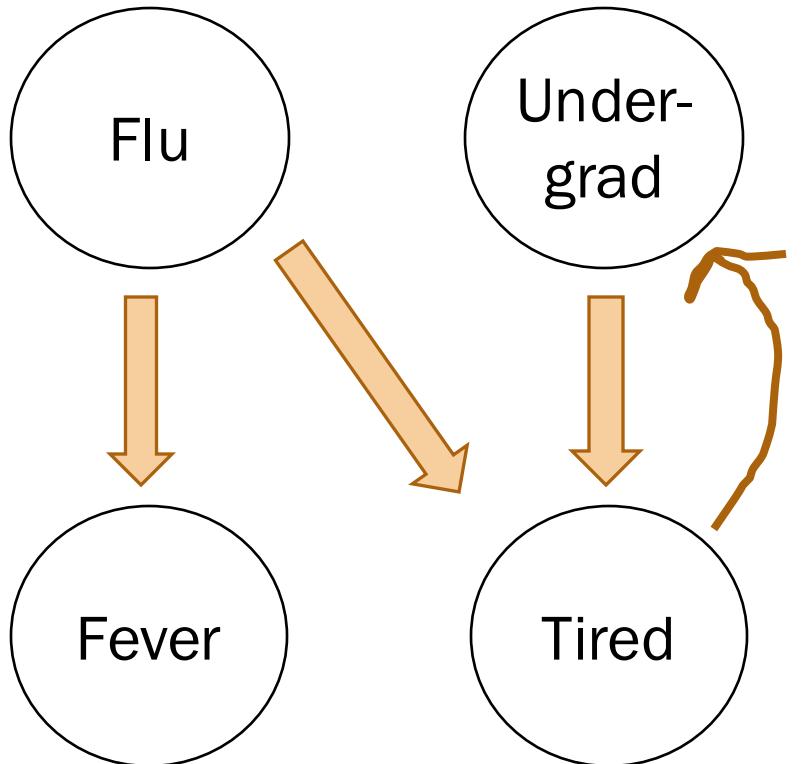


This model assumes that Flu and being an Undergraduate are independent.

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

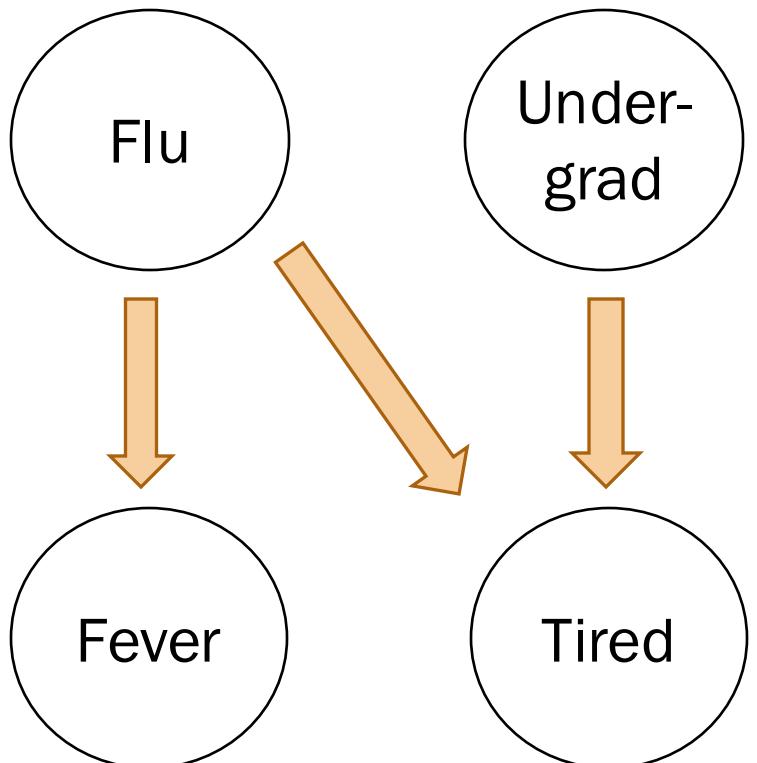
1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

First, the hard way

Algorithm #1: Inference via math

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

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



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

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

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

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

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

$$P(T = 1|F_{lu} = 1, U = 1) = 1.0$$

1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$

$$= P(F_{lu} = 0)$$

$$\cdot P(U = 1|F_{lu} = 0)$$

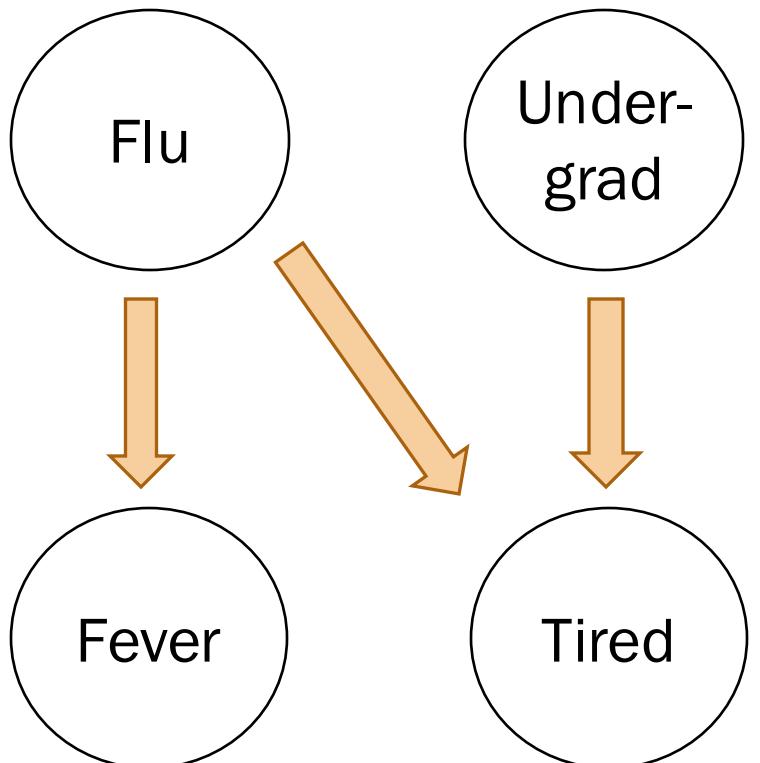
$$\cdot P(F_{ev} = 0|F_{lu} = 0, U = 1)$$

$$\cdot P(T = 1|F_{ev} = 0, F_{lu} = 0, U = 1)$$

Algorithm #1: Inference via math

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

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



$$\begin{aligned} P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\ P(F_{ev} = 1 | F_{lu} = 0) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\ P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\ P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\ P(T = 1 | F_{lu} = 1, U = 1) &= 1.0 \end{aligned}$$

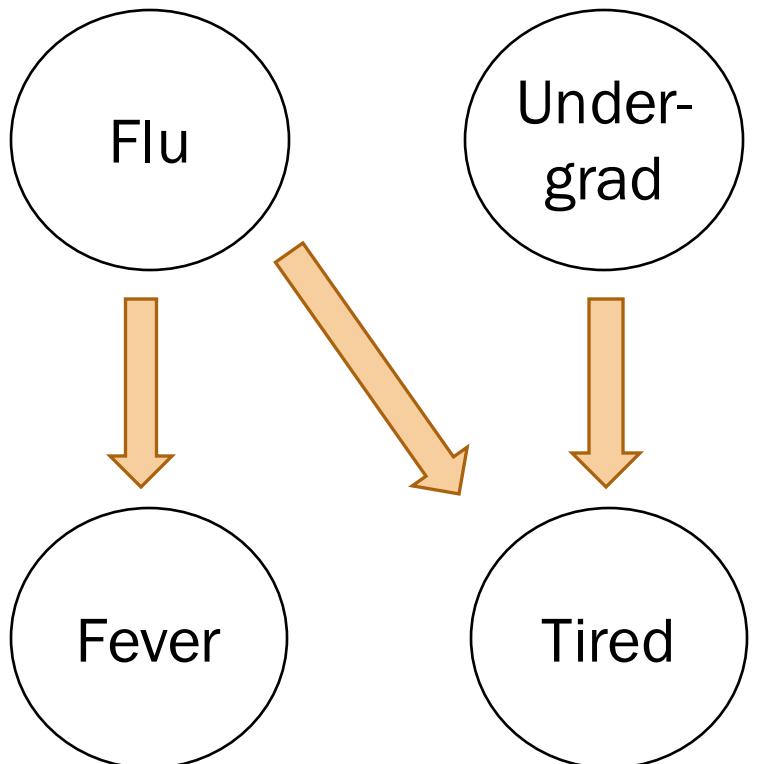
1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$

$$\begin{aligned} &= P(F_{lu} = 0) \\ &\cdot P(U = 1) \\ &\cdot P(F_{ev} = 0 | F_{lu} = 0, U = 1) \\ &\cdot P(T = 1 | F_{ev} = 0, F_{lu} = 0, U = 1) \end{aligned}$$

Algorithm #1: Inference via math

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

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



$$\begin{aligned} P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\ P(F_{ev} = 1 | F_{lu} = 0) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\ P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\ P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\ P(T = 1 | F_{lu} = 1, U = 1) &= 1.0 \end{aligned}$$

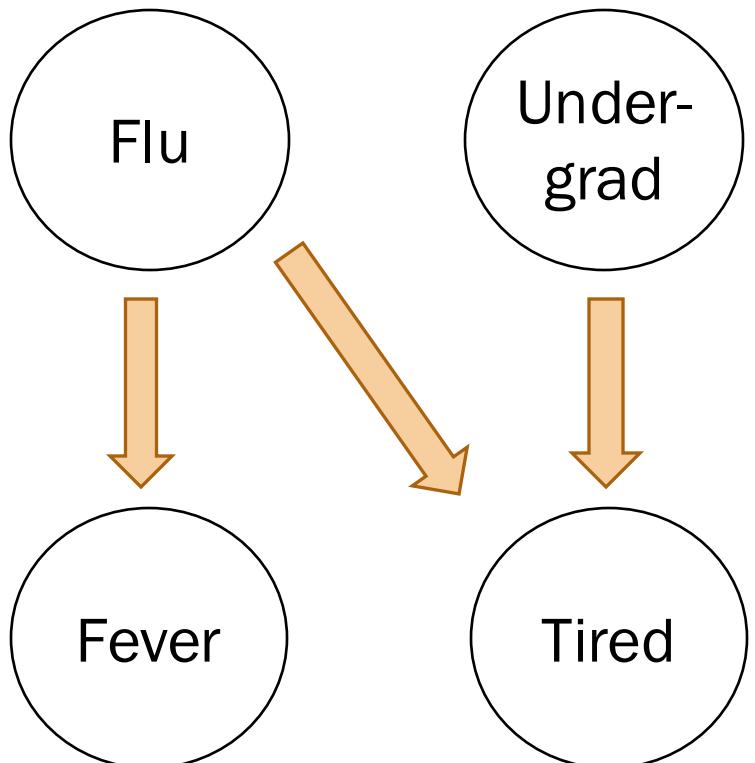
1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$

$$\begin{aligned} &= P(F_{lu} = 0) \\ &\cdot P(U = 1) \\ &\cdot P(F_{ev} = 0 | F_{lu} = 0) \\ &\cdot P(T = 1 | F_{ev} = 0, F_{lu} = 0, U = 1) \end{aligned}$$

Algorithm #1: Inference via math

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

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



$$\begin{aligned} P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\ P(F_{ev} = 1 | F_{lu} = 0) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\ P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\ P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\ P(T = 1 | F_{lu} = 1, U = 1) &= 1.0 \end{aligned}$$

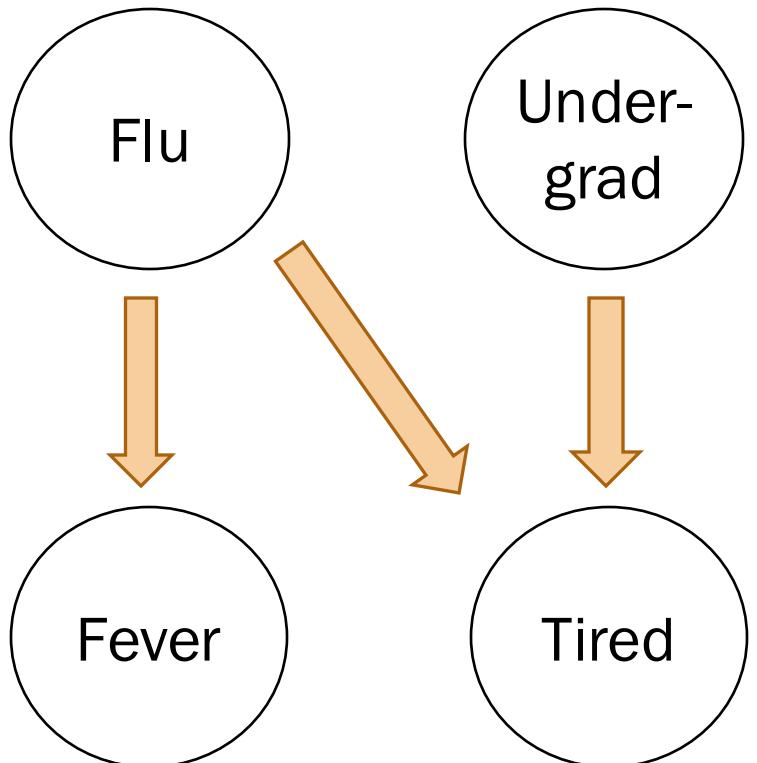
1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$

$$\begin{aligned} &= P(F_{lu} = 0) \\ &\cdot P(U = 1) \\ &\cdot P(F_{ev} = 0 | F_{lu} = 0) \\ &\cdot P(T = 1 | F_{lu} = 0, U = 1) \end{aligned}$$

Algorithm #1: Inference via math

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

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



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

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

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

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

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

$$P(T = 1|F_{lu} = 1, U = 1) = 1.0$$

1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$

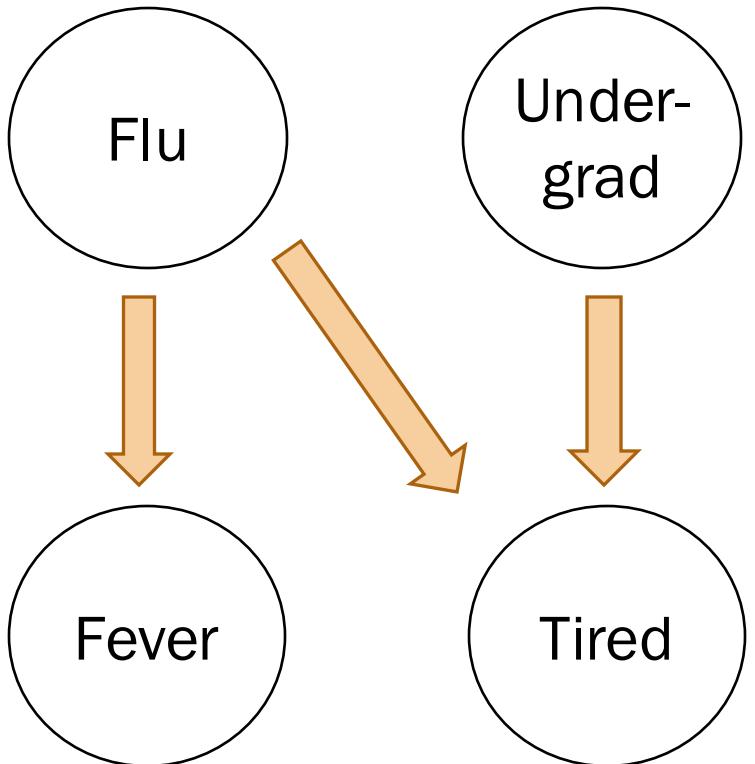
$$= P(F_{lu} = 0) \quad 0.9$$

$$\cdot P(U = 1) \quad 0.2$$

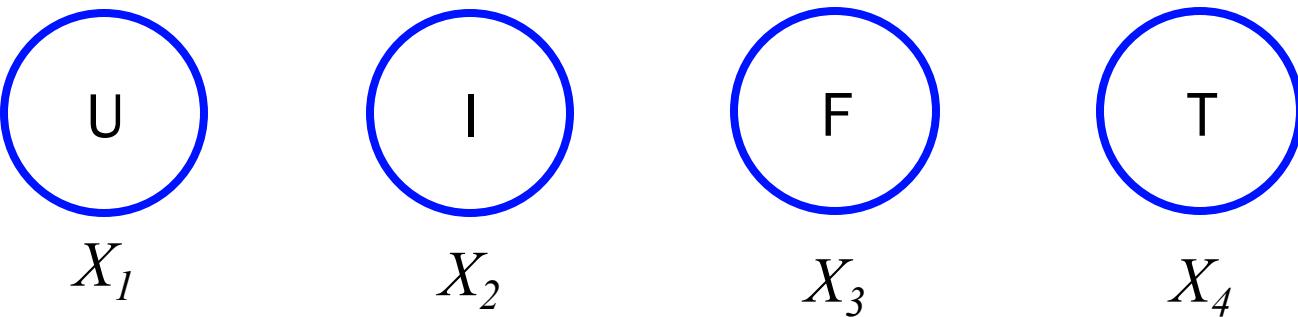
$$\cdot P(F_{ev} = 0|F_{lu} = 0) \quad 0.05$$

$$\cdot P(T = 1|F_{lu} = 0, U = 1) \quad 0.8$$

Algorithm #1: Inference via math



Order nodes by ancestry



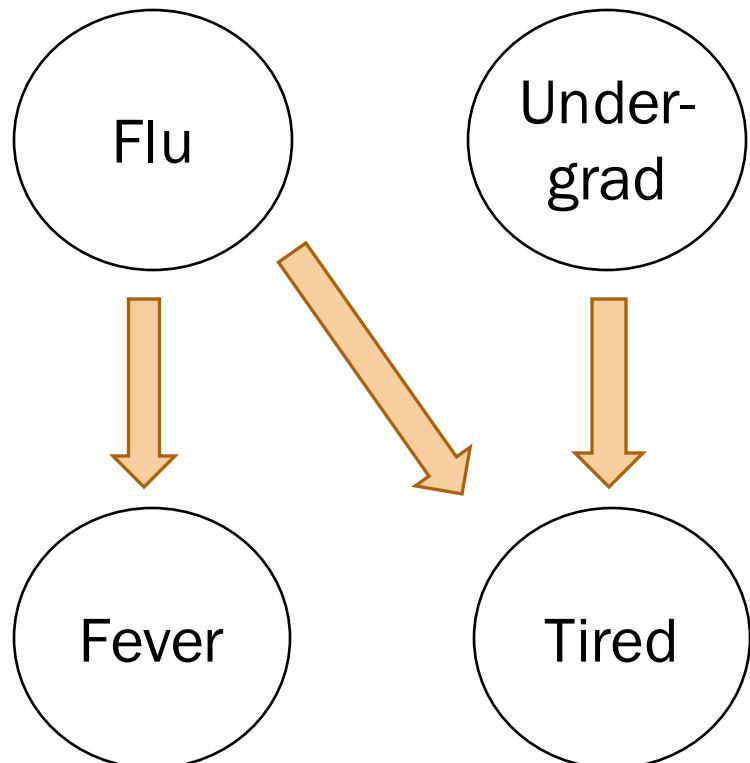
$$\begin{aligned} P(\text{Joint}) &= \prod_i P(x_i | x_{i-1}, \dots, x_1) \\ &= \prod_i P(x_i | \text{Values of parents of } X_i) \end{aligned}$$

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

Inference via math

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

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



$$\begin{aligned} P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\ P(F_{ev} = 1 | F_{lu} = 0) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\ P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\ P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\ P(T = 1 | F_{lu} = 1, U = 1) &= 1.0 \end{aligned}$$

$$P(\text{Joint}) = \prod_i P(x_i | \text{Values of parents of } X_i)$$

2. $P(F_{lu} = 1 | F_{ev} = 0, U = 0, T = 1)?$

1. Compute joint probabilities

$$P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)$$

$$P(F_{lu} = 0, F_{ev} = 0, U = 0, T = 1)$$

2. Definition of conditional probability

$$\frac{P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)}{\sum_x P(F_{lu} = x, F_{ev} = 0, U = 0, T = 1)}$$

$$= 0.095$$

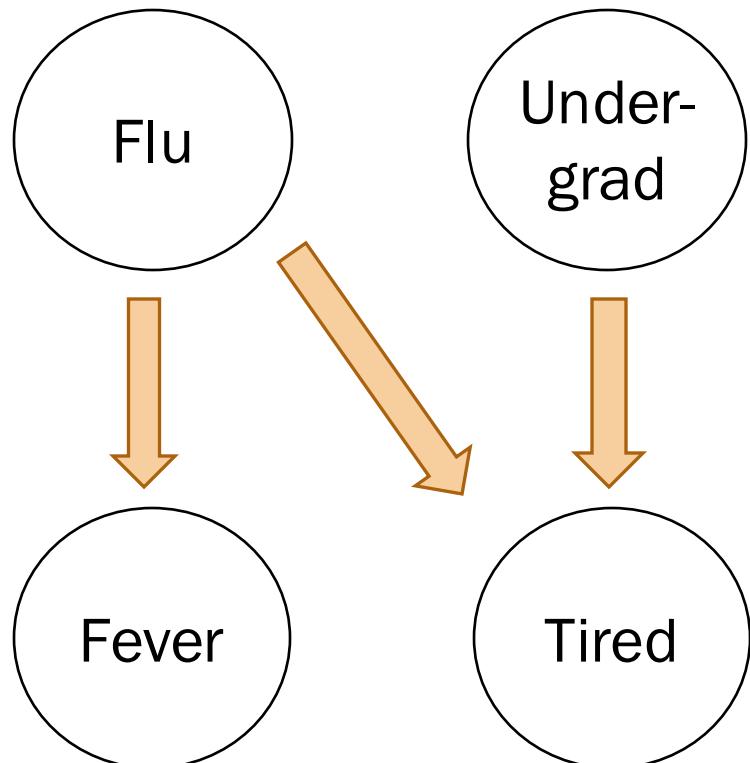
Want more practice?

Inference via math

$$P(\text{Joint}) = \prod_i P(x_i | \text{Values of parents of } X_i)$$

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

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



$$\begin{aligned}P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\P(F_{ev} = 1 | F_{lu} = 0) &= 0.05\end{aligned}$$

$$\begin{aligned}P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\P(T = 1 | F_{lu} = 1, U = 1) &= 1.0\end{aligned}$$

3. $P(F_{lu} = 1 | U = 1, T = 1)?$

1. Compute joint probabilities

$$P(F_{lu} = 1, \textcolor{blue}{U} = 1, F_{ev} = 1, \textcolor{blue}{T} = 1)$$

...

$$P(F_{lu} = 0, \textcolor{blue}{U} = 1, F_{ev} = 0, \textcolor{blue}{T} = 1)?$$

2. Definition of conditional probability

$$\frac{\sum_y P(F_{lu} = 1, \textcolor{blue}{U} = 1, F_{ev} = y, \textcolor{blue}{T} = 1)}{\sum_x \sum_y P(F_{lu} = x, \textcolor{blue}{U} = 1, F_{ev} = y, \textcolor{blue}{T} = 1)}$$

$$= 0.122$$

There must be a better way...

[suspense]

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          set 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': 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': 0, 'fever': 0, 'tired': 1}
{'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:

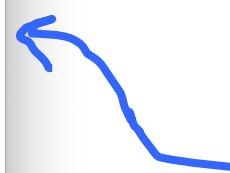
$$P(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n} \quad n = \# \text{ of total trials} \quad n(E) = \# \text{ trials where } E \text{ occurs}$$



```
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': 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': 0, '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 %
```



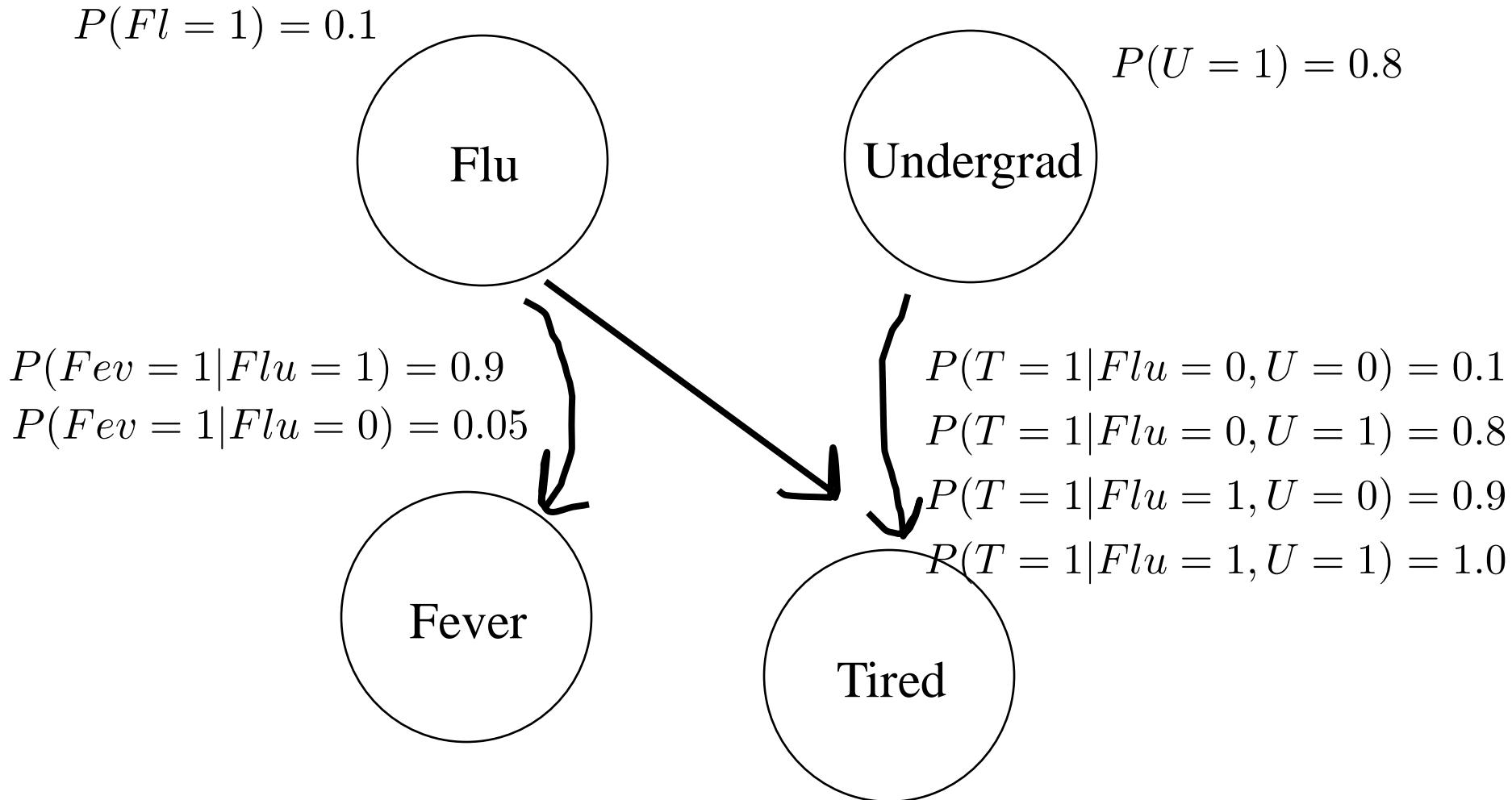
If you can sample enough
from the joint distribution,
you can answer any
probability question



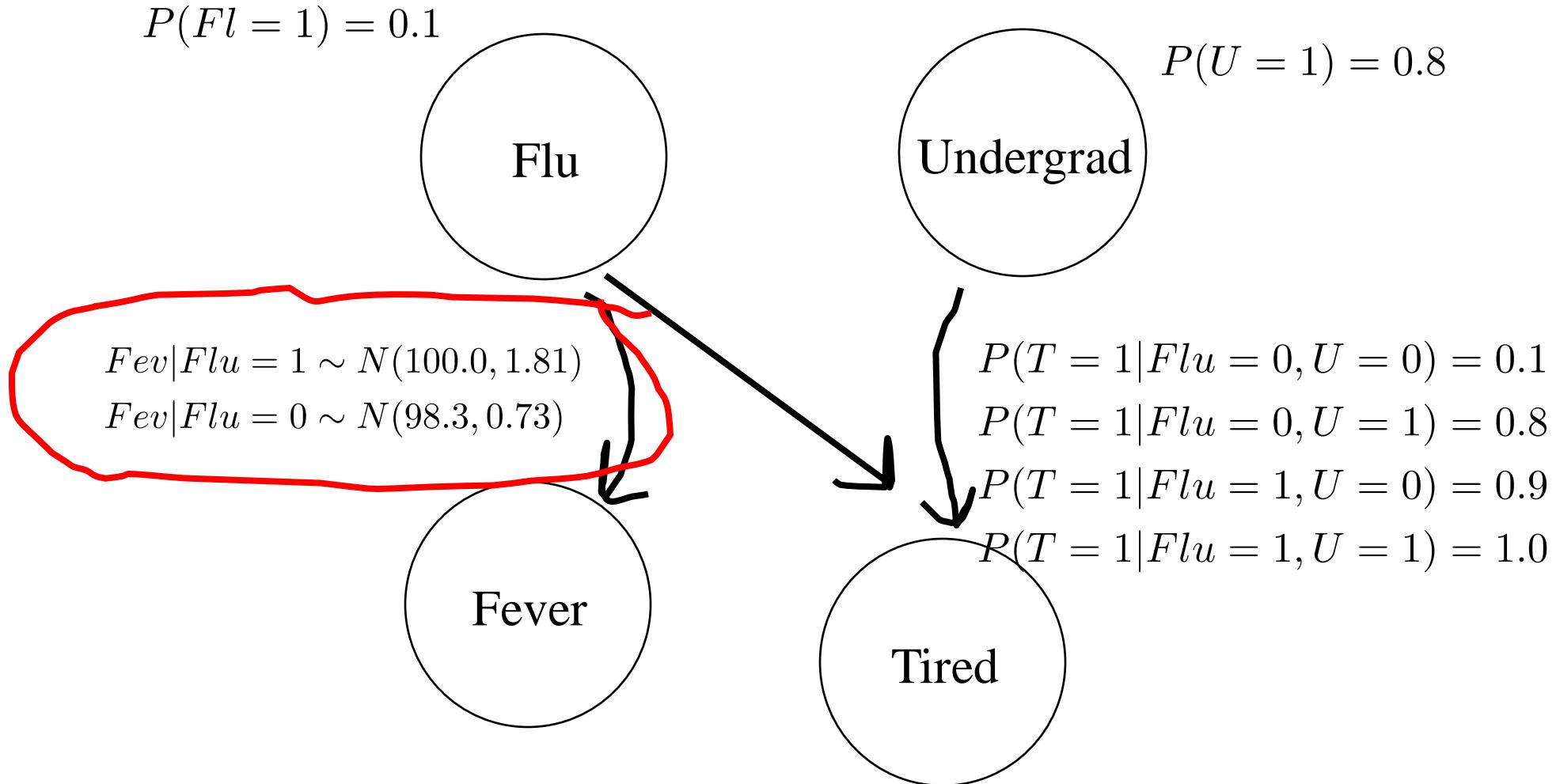
Each one of these is one joint
sample

What's the matter with
rejection sampling?

Probabilistic Model



Probabilistic Model



The Magic School Bus™



University

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?

AutoSave OFF > Search Sheet

Home Insert Page Layout Formulas Data Share Cells Editing

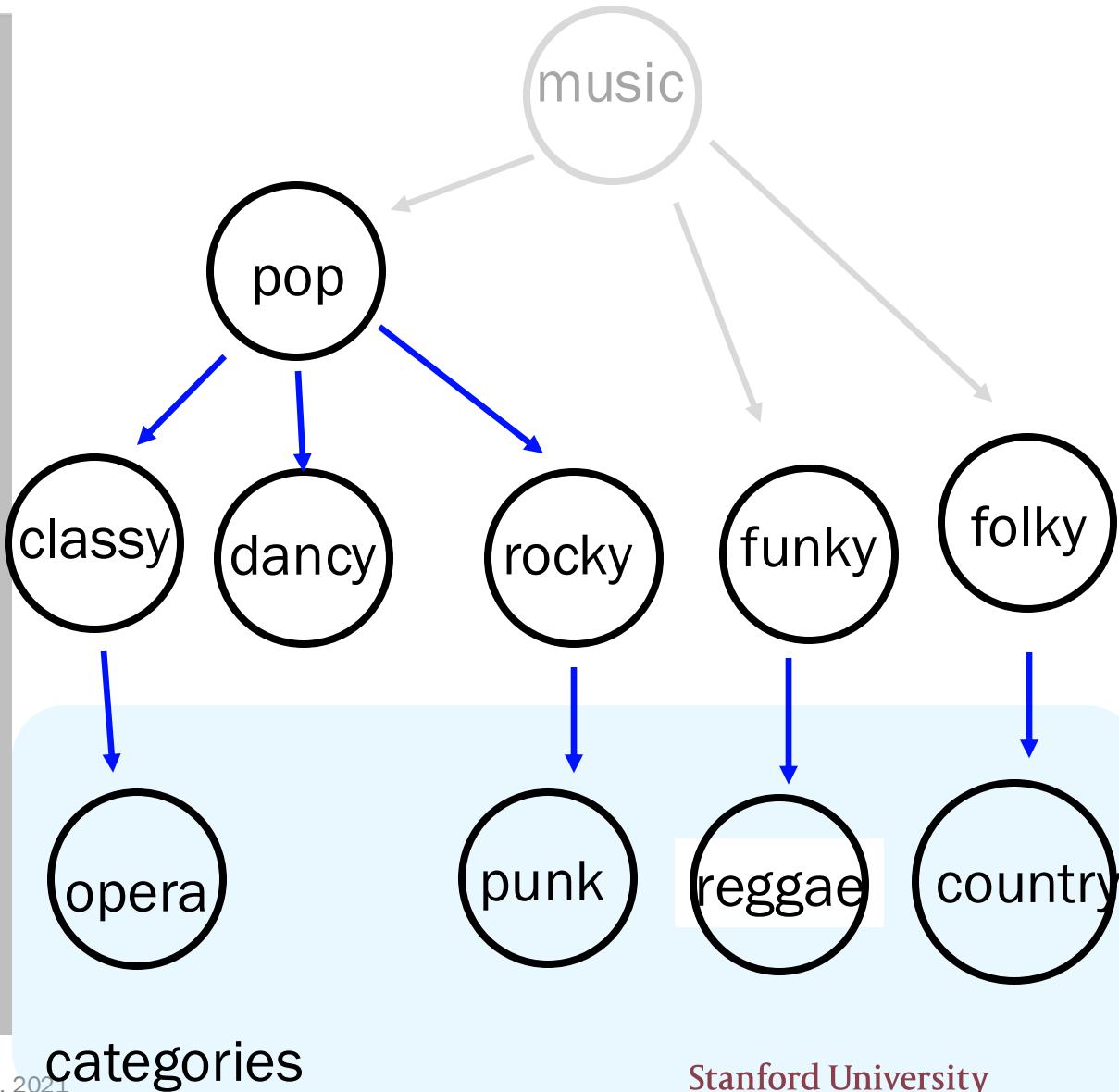
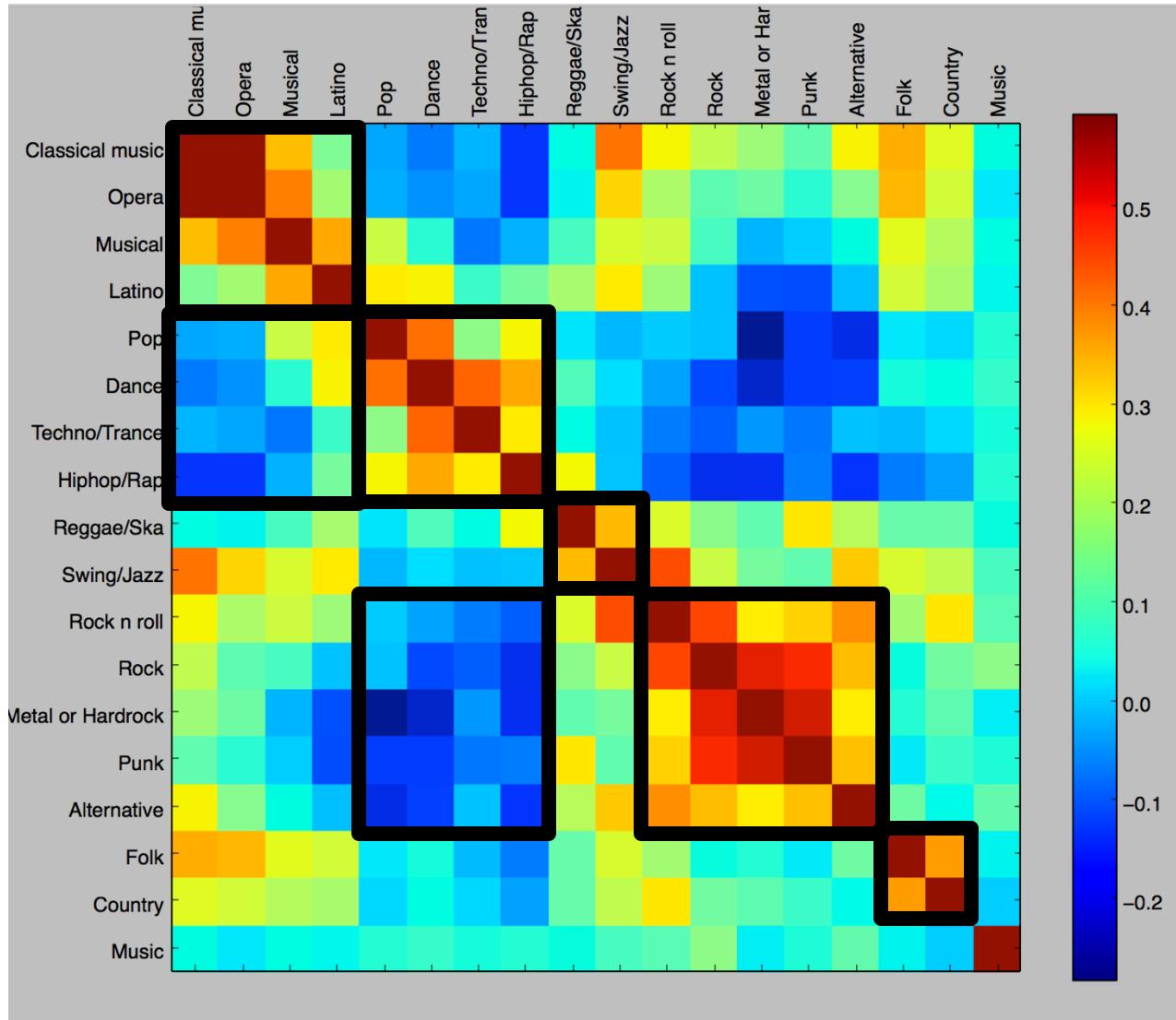
C15 A B C D E F G H M

	Music	Dance	Folk	Country	Classical music	Musical	Pop	Rock	M
1	5	2	1	2	2	1	1	5	5
2	4	2	1	1	1	1	2	3	5
3	5	2	2	3	4	5	3	3	5
4	5	2	1	1	1	1	1	2	2
5	5	4	3	2	4	3	5	3	3
6	5	2	3	2	3	3	2	2	5
7	5	5	3	1	2	2	2	3	5
8	5	3	2	1	1	2	2	2	5
9	5	1	1	1	4	1	2	4	5
10	5	3	1	1	2	4	3	3	5
11	5	2	5	2	2	2	5	3	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	3	5
15	5	5	3	2	1	5	5	2	2
16	5	2	1	1	2	3	4	3	5
17	1	2	2	3	4	3	3	3	5
18	5	3	1	1	1	2	2	4	4
19	5	3	3	3	3	2	2	4	4
20	5	5	4	3	4	5	5	5	4
21	5	3	3	2	2	2	2	2	5
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	4	4	4	5
26	5	3	1	1	4	3	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	3	2	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?

Home Insert Page Layout Formulas Data Share Search Sheet

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	3	2	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	3	5
11	5	2	5	2	2	5	3	3	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	3	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	2	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	2	5
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	5
25	5	4	2	2	2	4	4	4	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	3	4
35	5	2	2	3	4	5	4	4	3

music +

Ready

AutoSave OFF > Q Search Sheet

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

X ✓ fx 3

A... Alignment %

Cells Editing

Home Insert Page Layout Formulas Data Share Share ^

100%

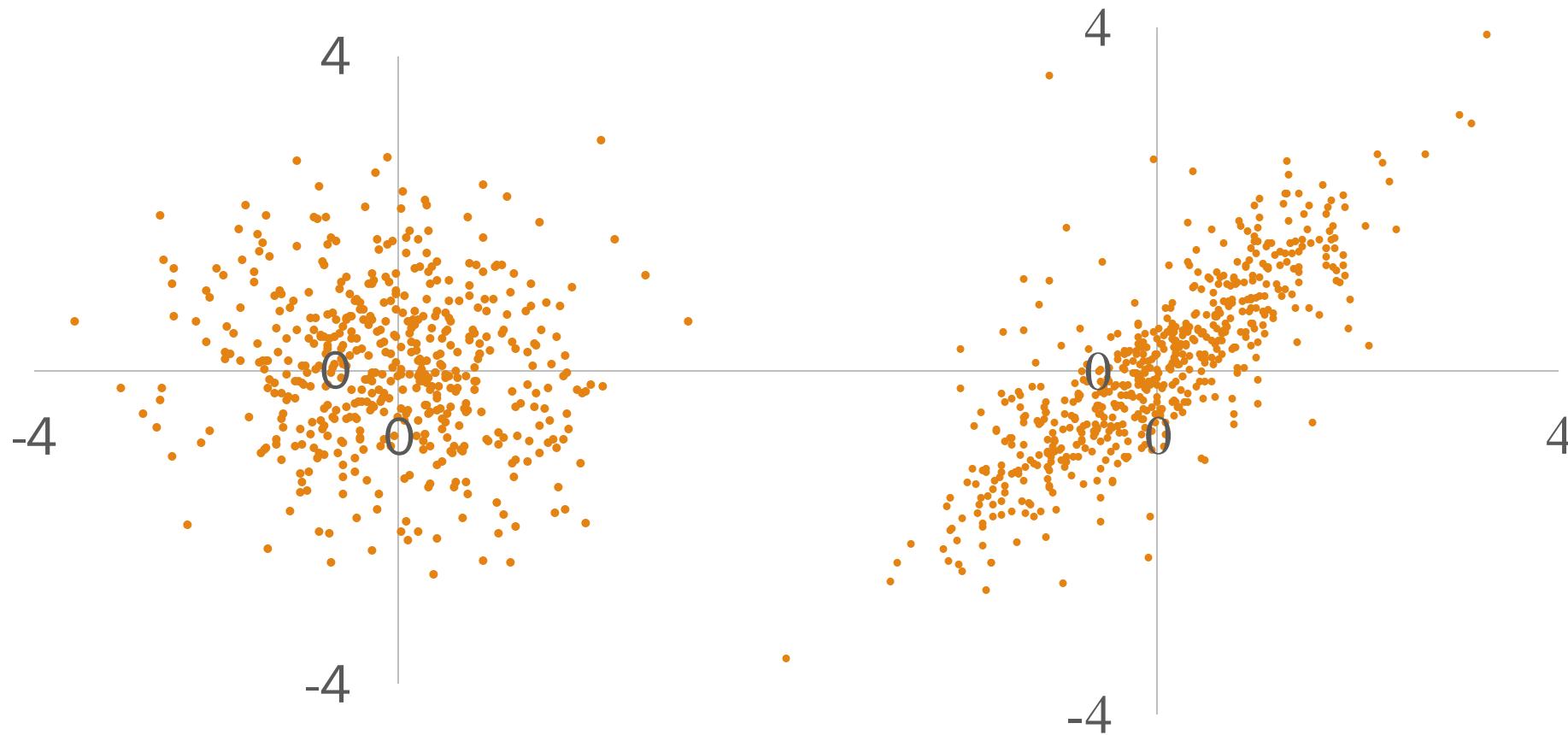
Bat Data

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

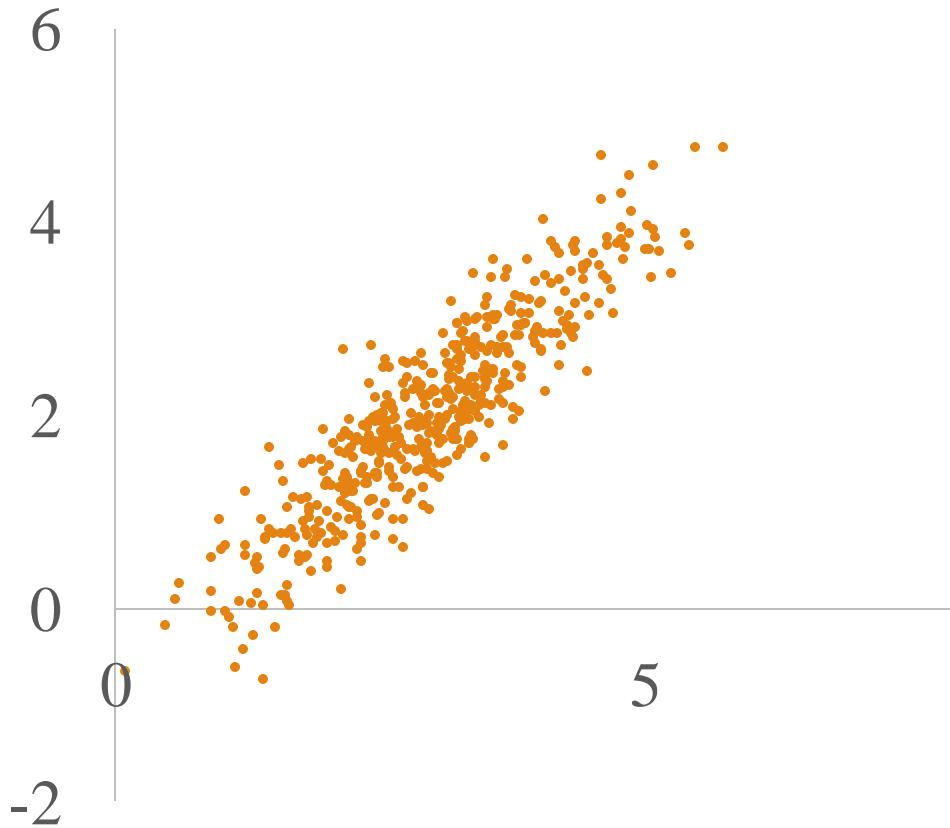
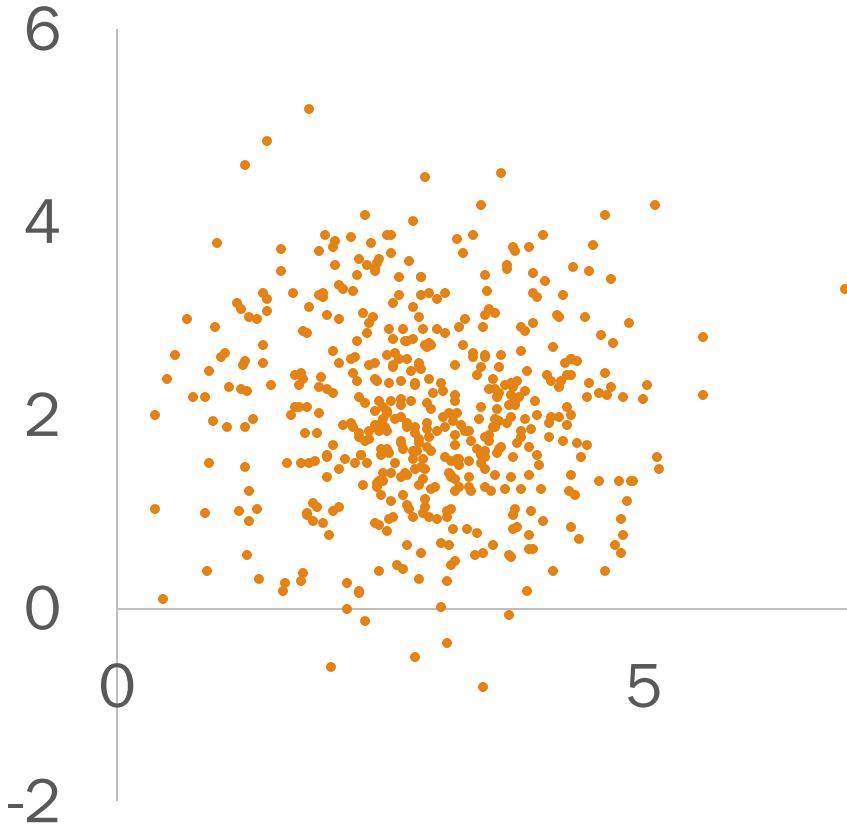
Expression Amount

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

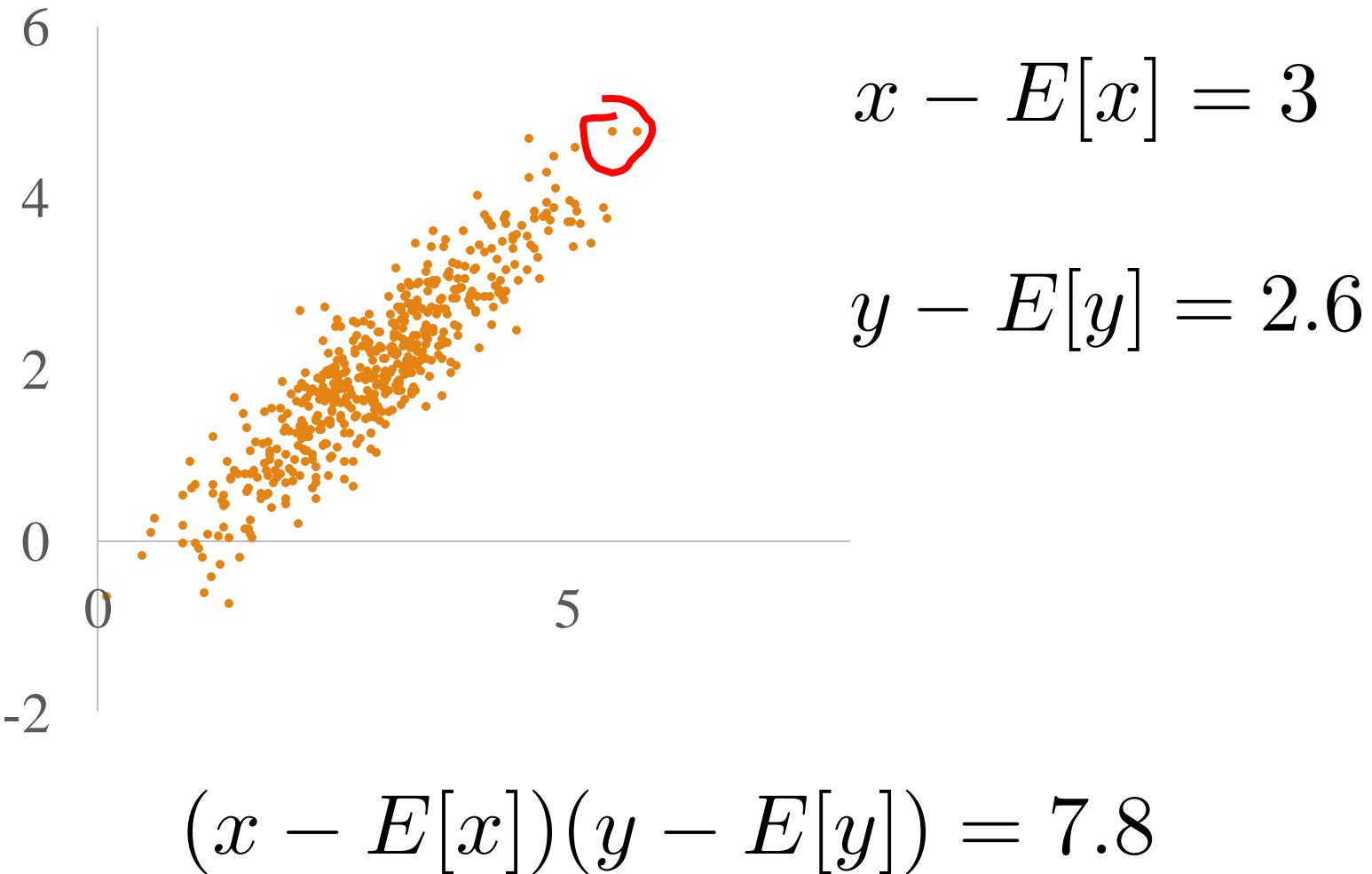
Spot The Difference



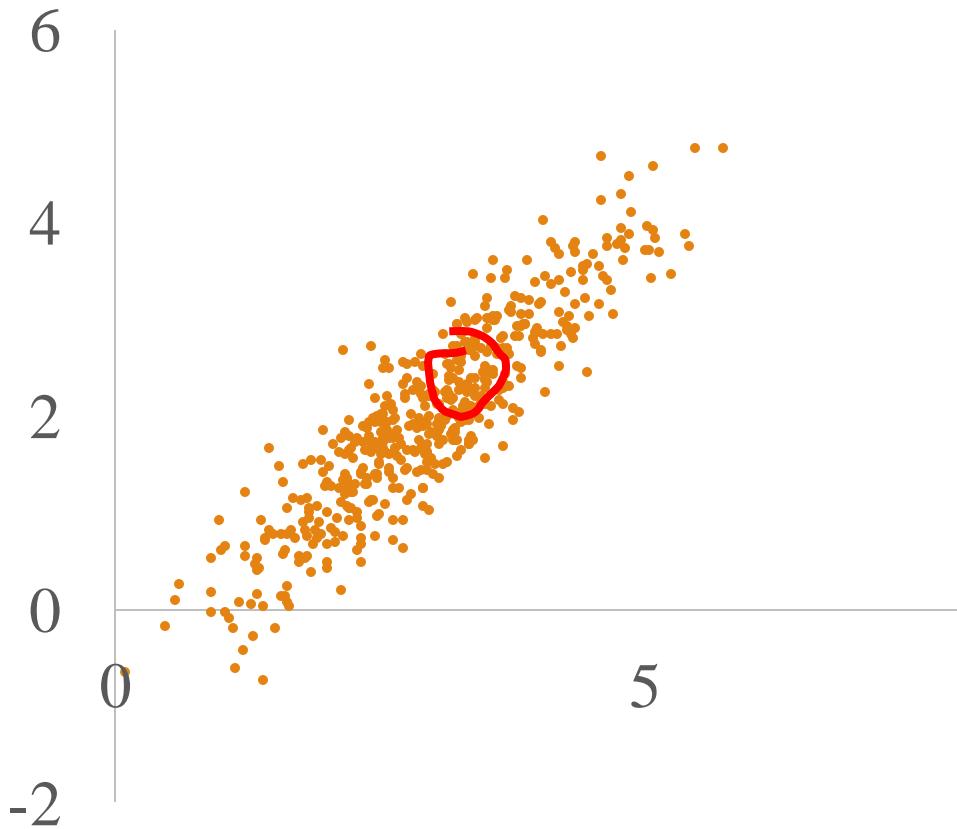
Spot The Difference



Vary Together



Vary Together

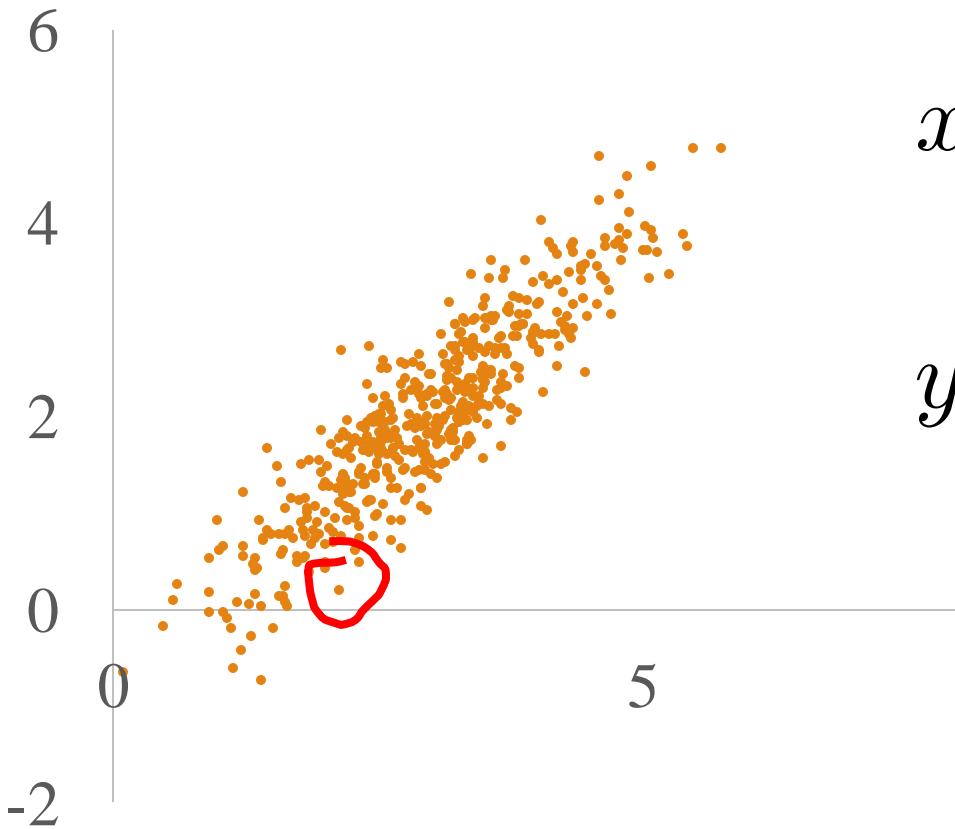


$$x - E[x] \approx 0$$

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

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

Vary Together

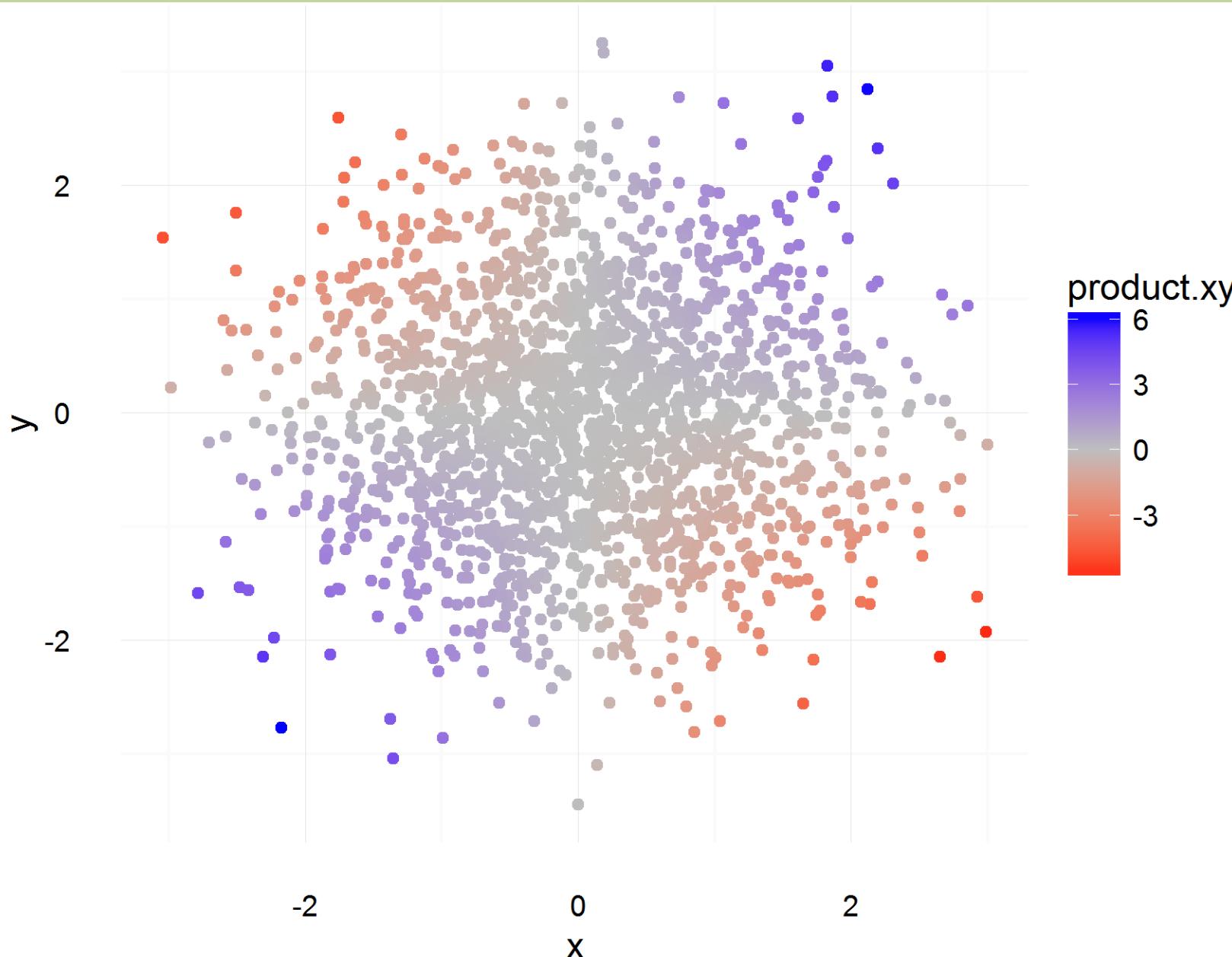


$$x - E[x] = -1.1$$

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

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

Understanding Covariance



The Dance of the Covariance

Say X and Y are arbitrary random variables

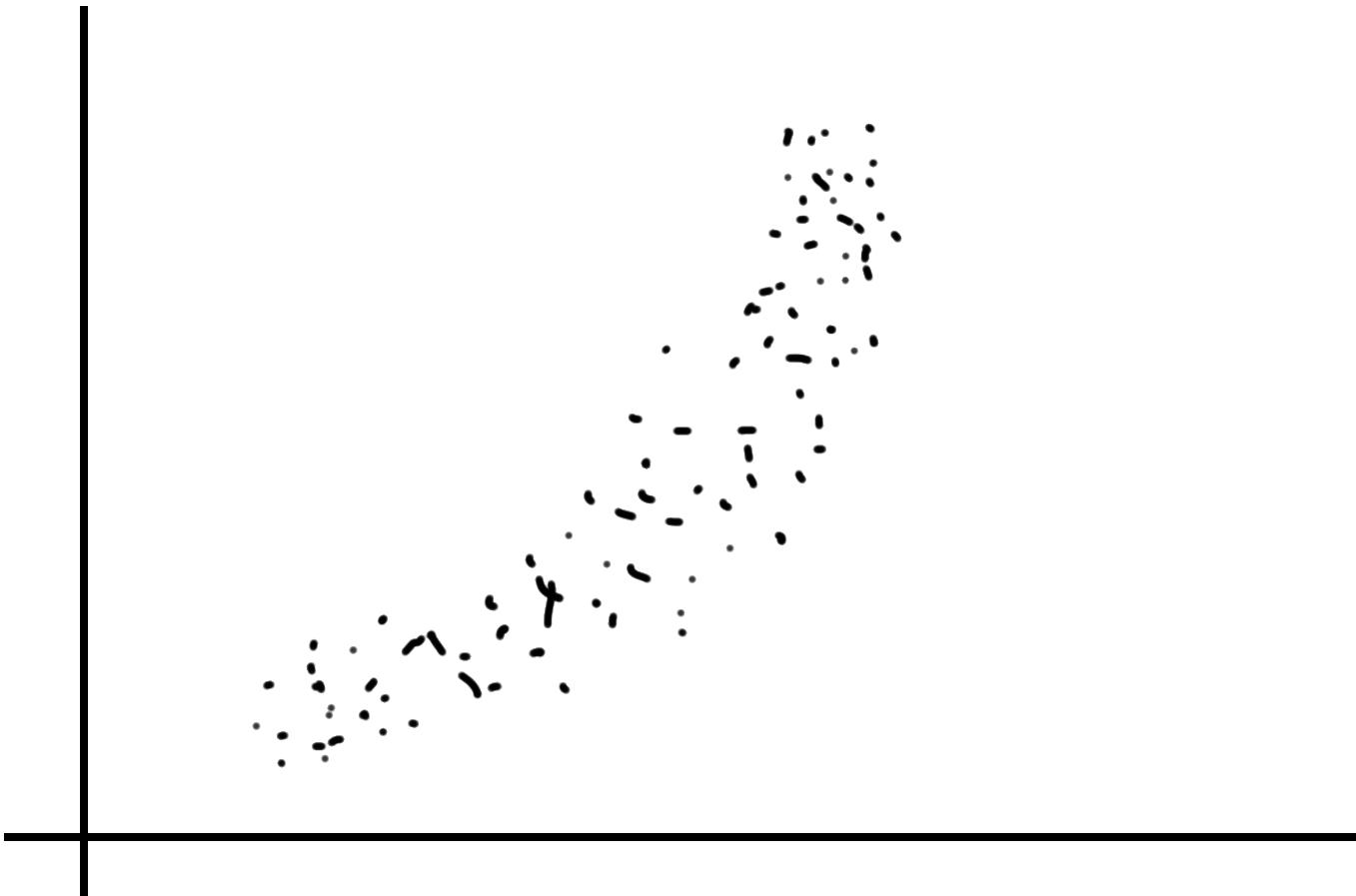
Covariance of X and Y :

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

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

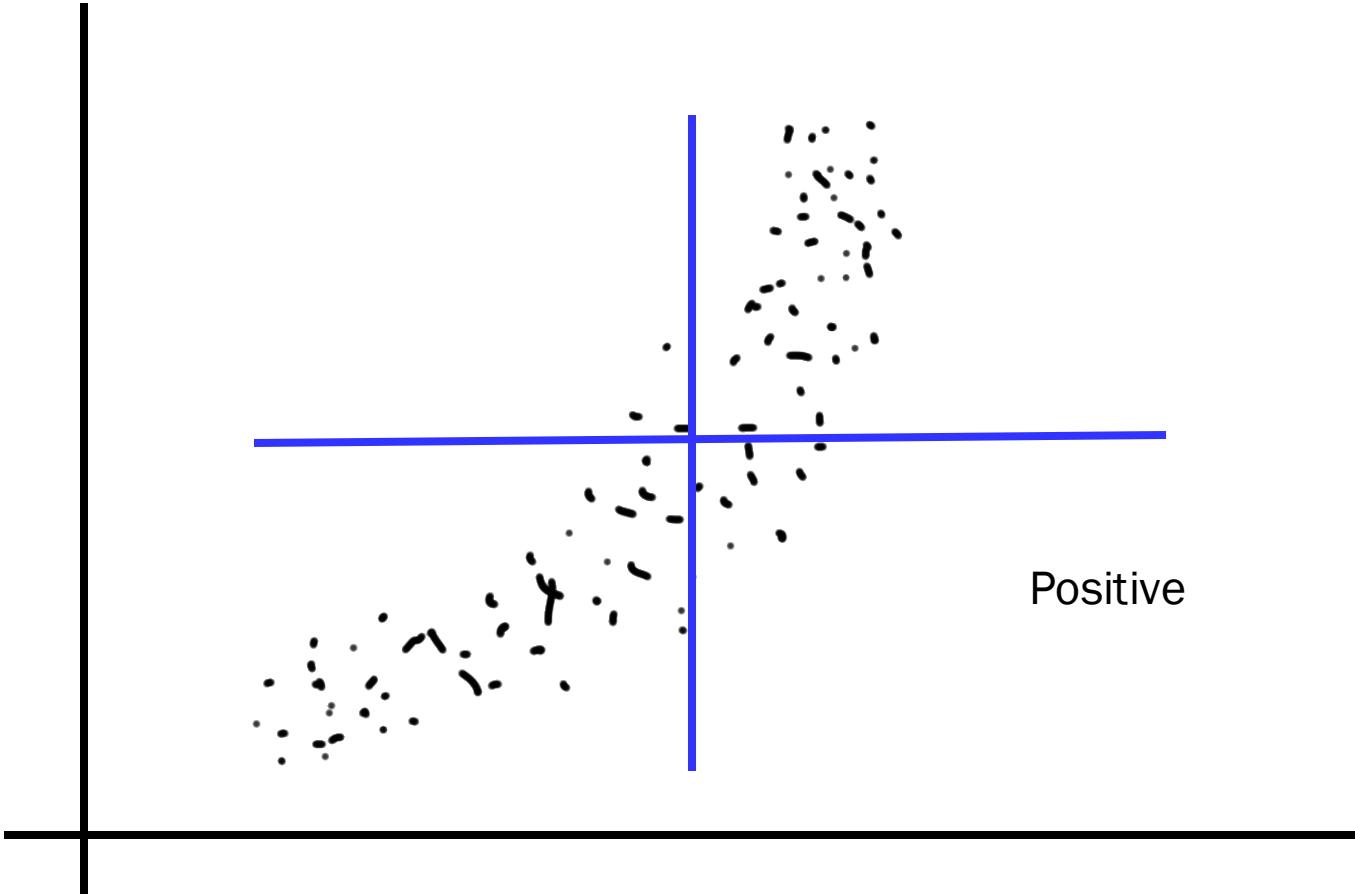
Covariance

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



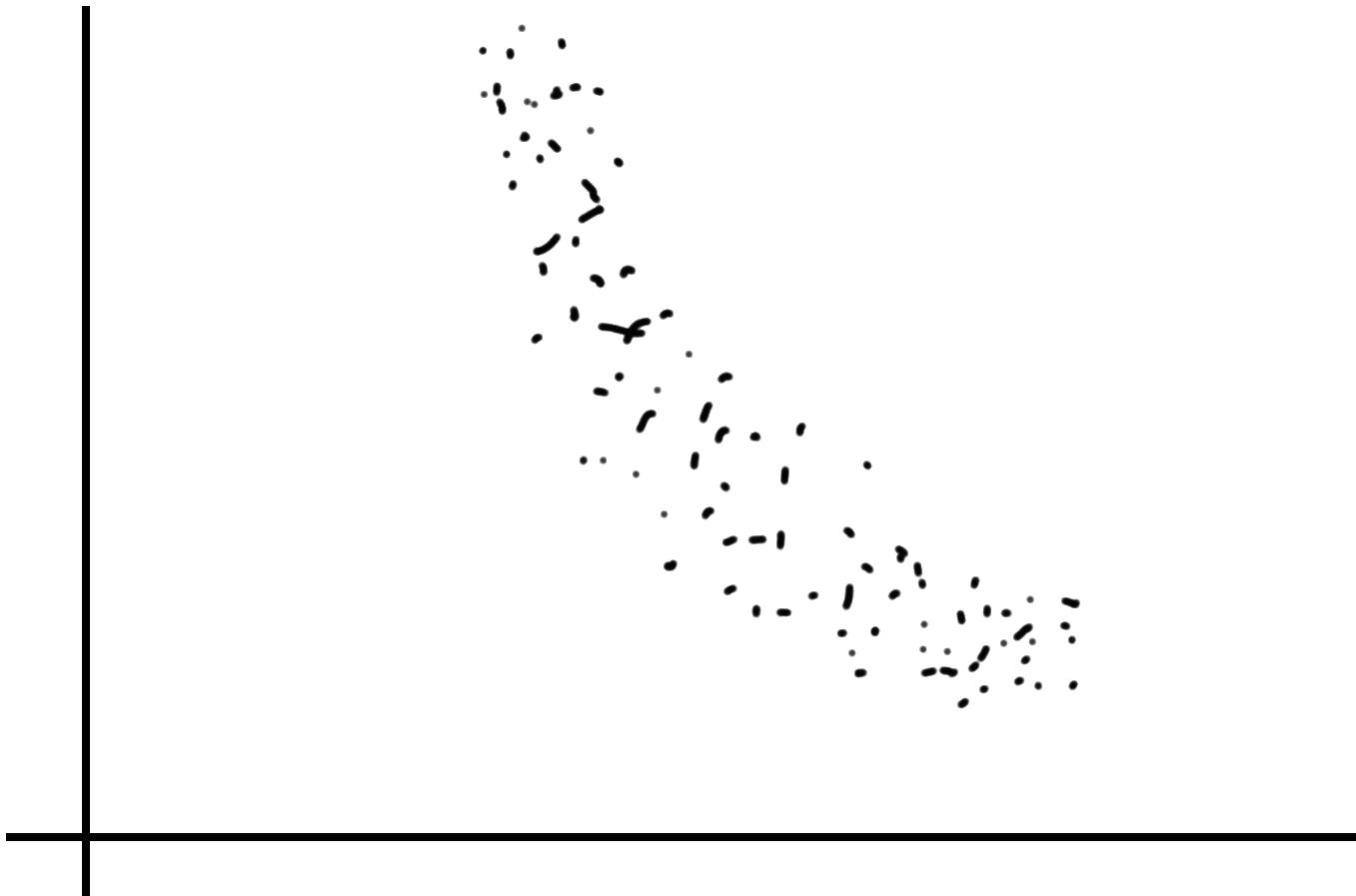
Covariance

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



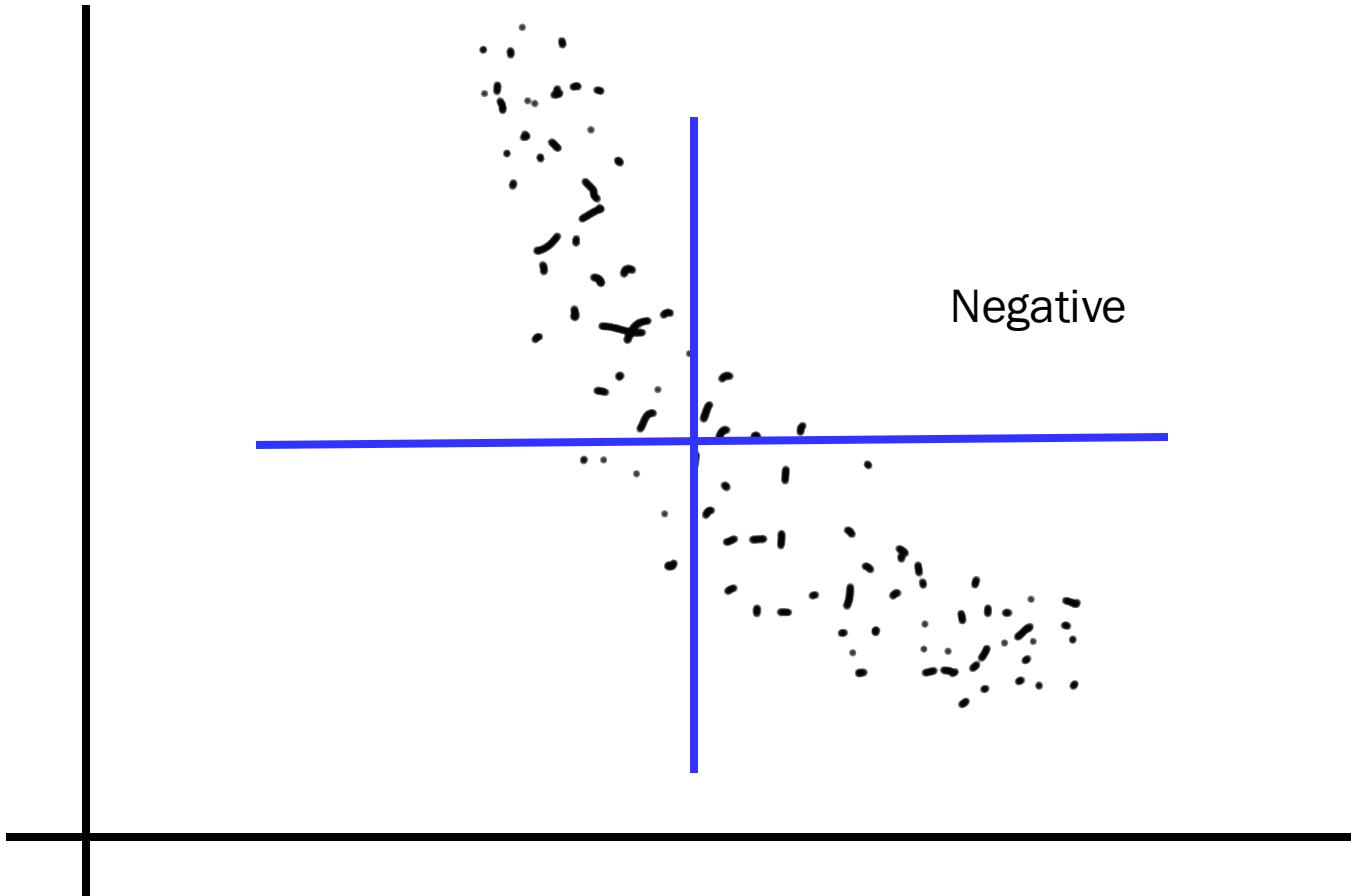
Covariance

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



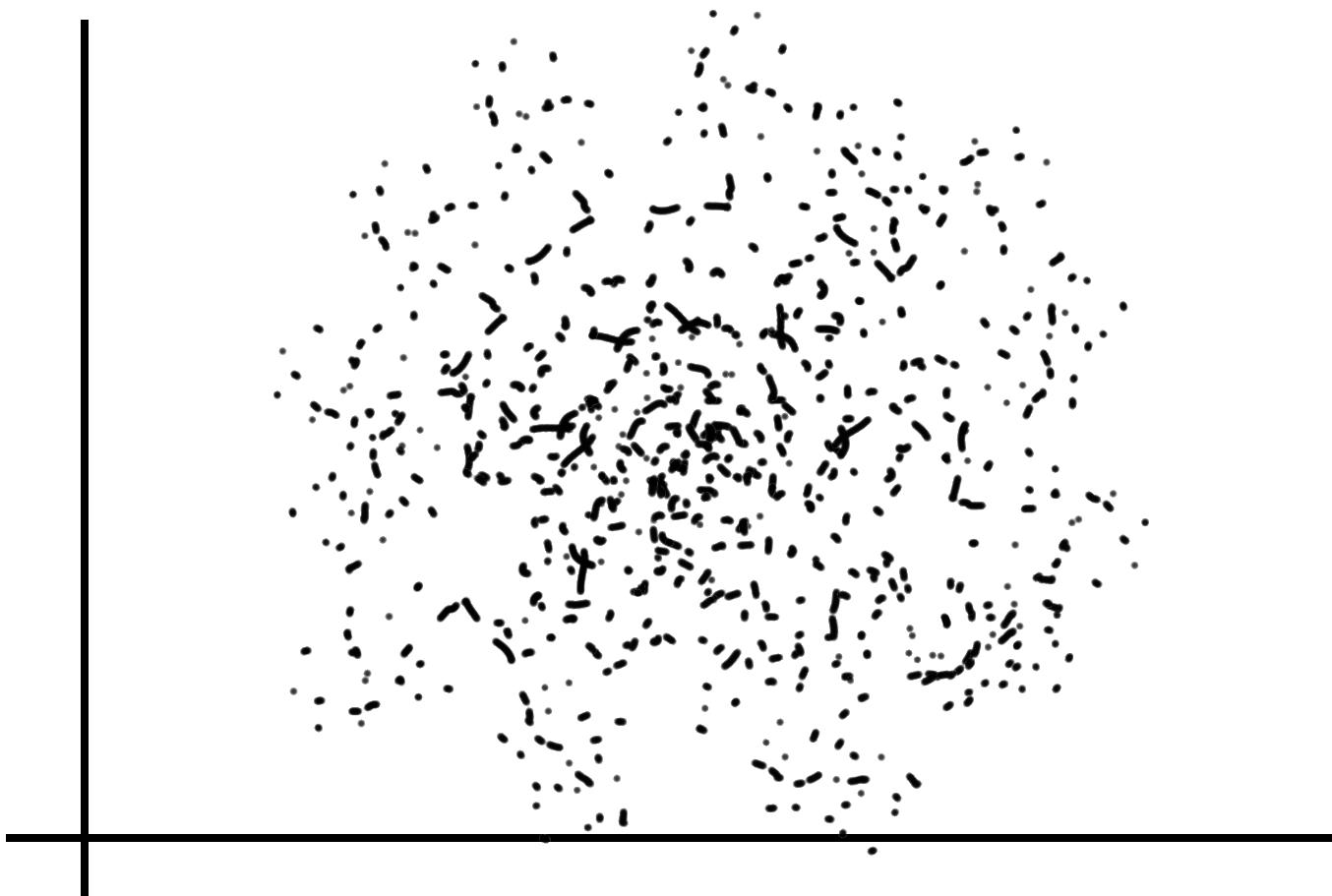
Covariance

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



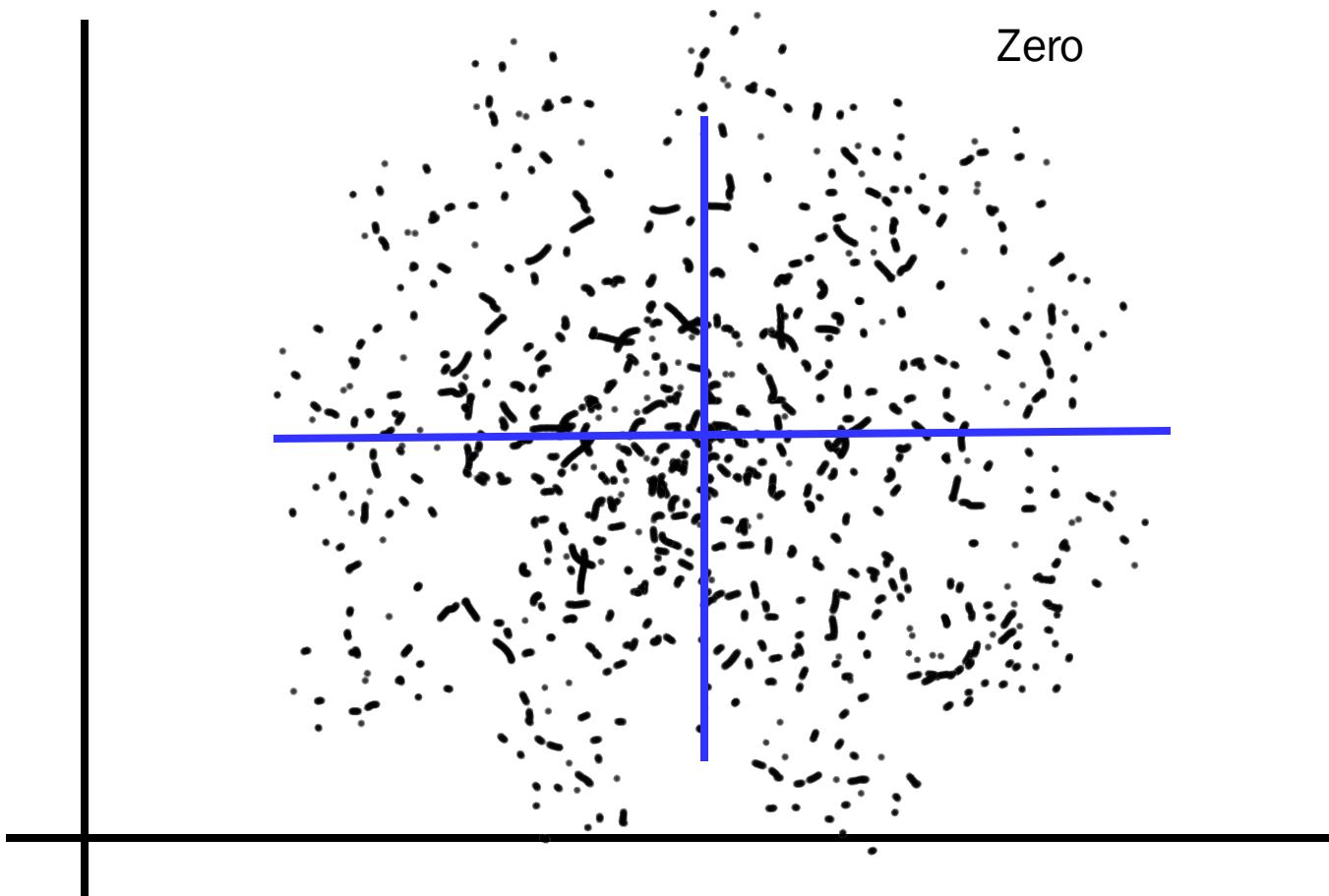
Covariance

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
----	----	------

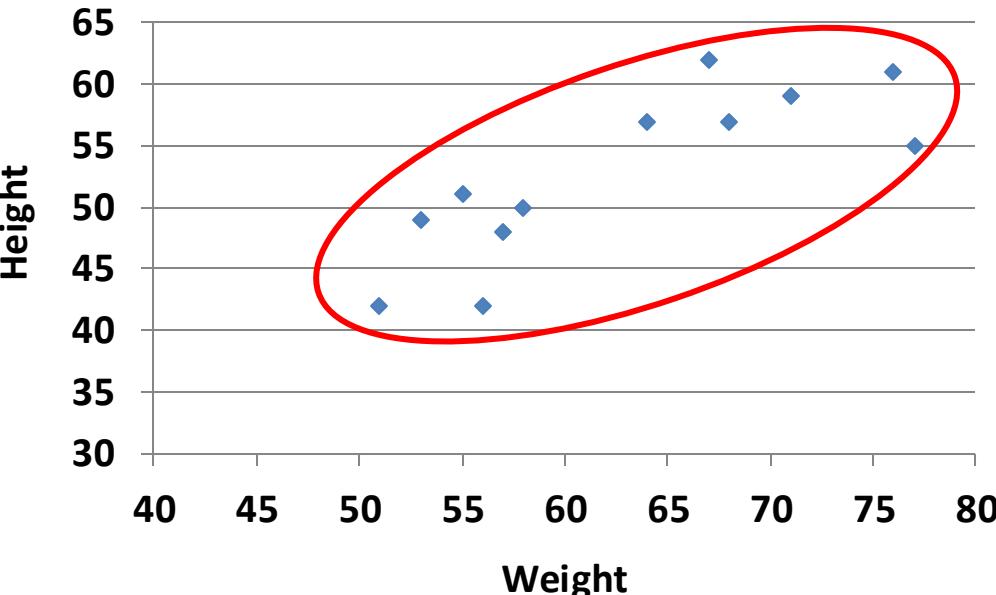
51	42	2142
----	----	------

76	61	4636
----	----	------

68	57	3876
----	----	------

$$\begin{aligned}E[W] &= 62.75 \\E[H] &= 52.75\end{aligned}$$

$$\begin{aligned}E[W^*H] &= 3355.83\end{aligned}$$



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

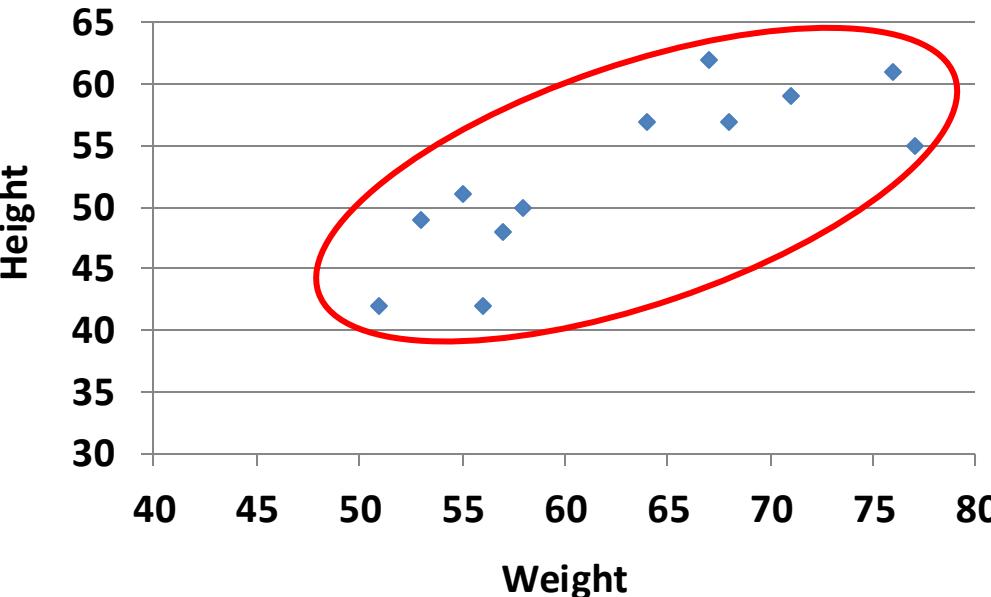
Correlation

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

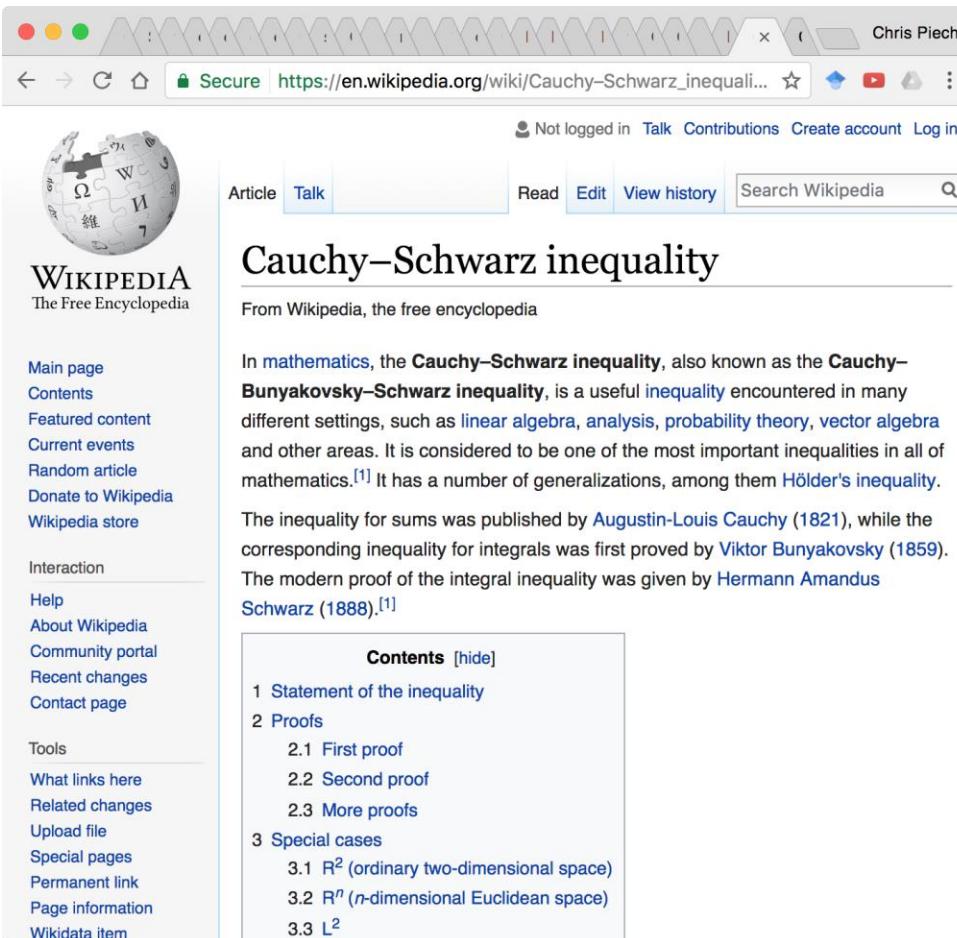
$E[W]$	$E[H]$	$E[W^*H]$
= 62.75	= 52.75	= 3355.83



$$\text{Cov}(W, H) = E[W^*H] - E[W]E[H]$$

$$\begin{aligned} &= 3355.83 - (62.75)(52.75) \\ &= 45.77 \end{aligned}$$

Cauchy Schwarz, a great way to normalize!



The screenshot shows a web browser window with the URL https://en.wikipedia.org/wiki/Cauchy-Schwarz_inequality. The page title is "Cauchy–Schwarz inequality". The left sidebar contains links to the Main page, Contents, Featured content, Current events, Random article, Donate to Wikipedia, Wikipedia store, Interaction, Help, About Wikipedia, Community portal, Recent changes, Contact page, Tools, What links here, Related changes, Upload file, Special pages, Permanent link, Page information, and Wikidata item. The main content area starts with a brief introduction about the inequality's history and applications in mathematics, linear algebra, analysis, probability theory, and vector algebra. It mentions generalizations like Hölder's inequality. Below this is a "Contents" section with links to the Statement of the inequality, Proofs (First proof, Second proof, More proofs), and Special cases (R², Rⁿ, L²).

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

Viva La Correlatióñ

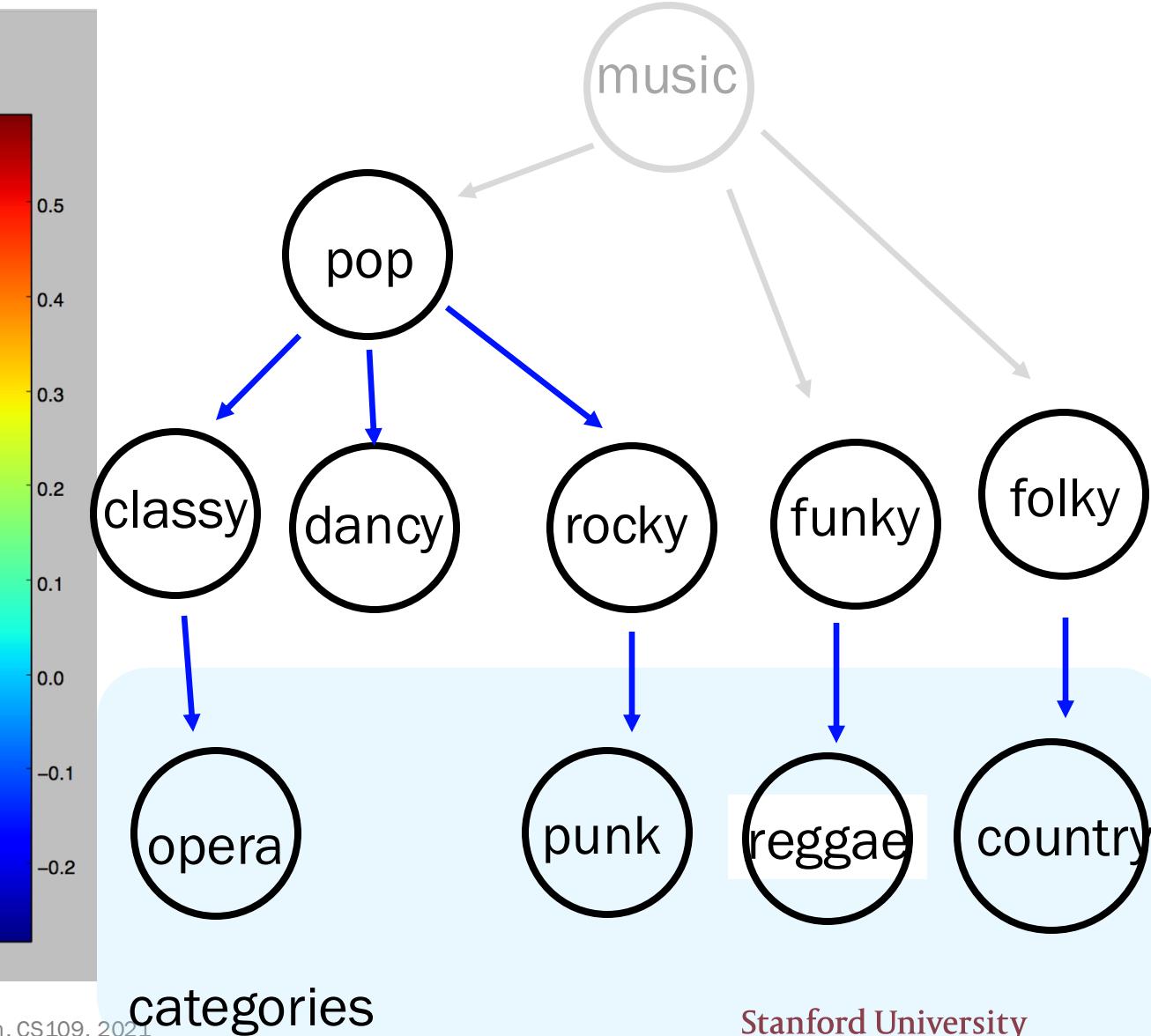
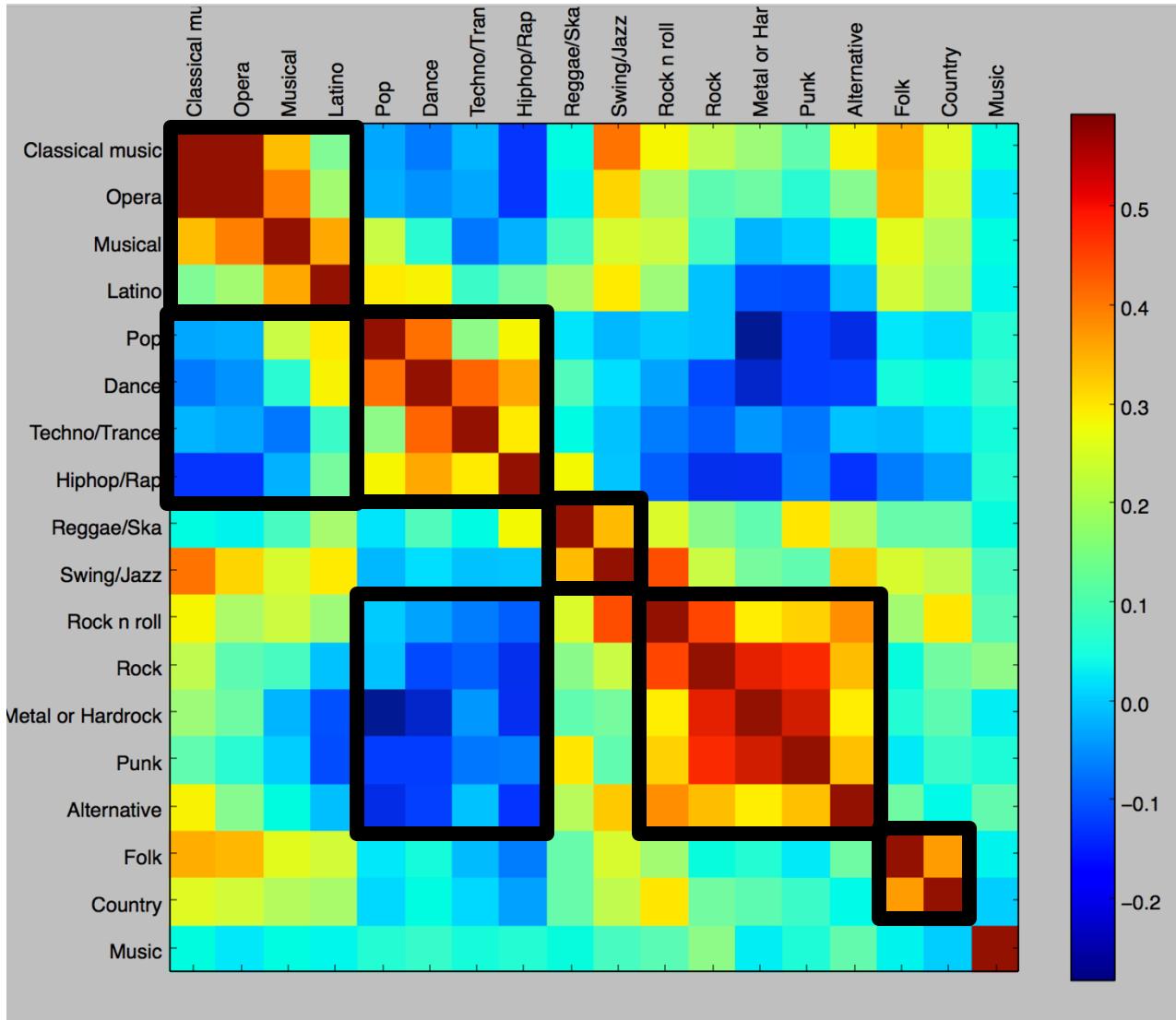
Say X and Y are arbitrary random variables

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

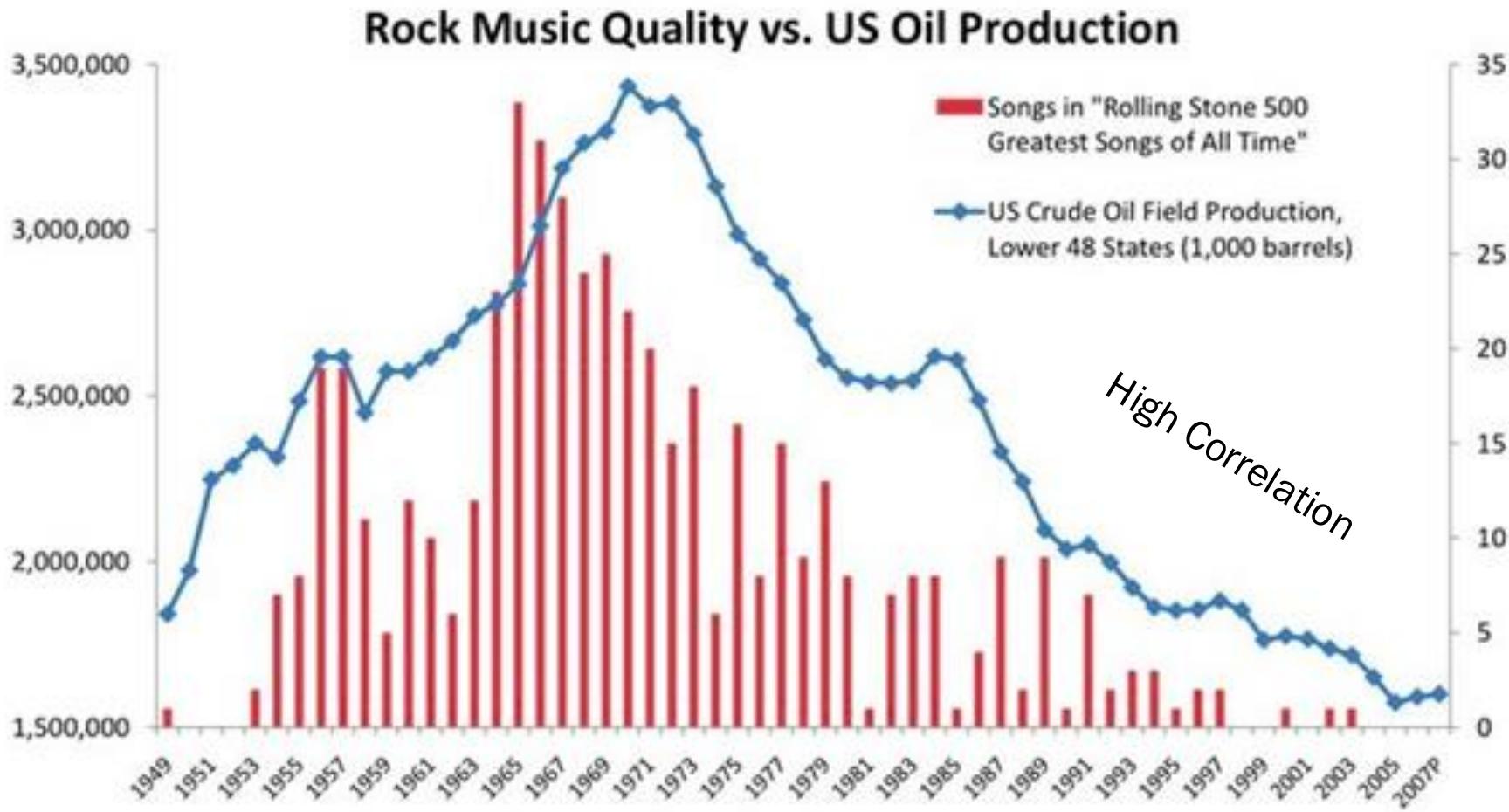
$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

- 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

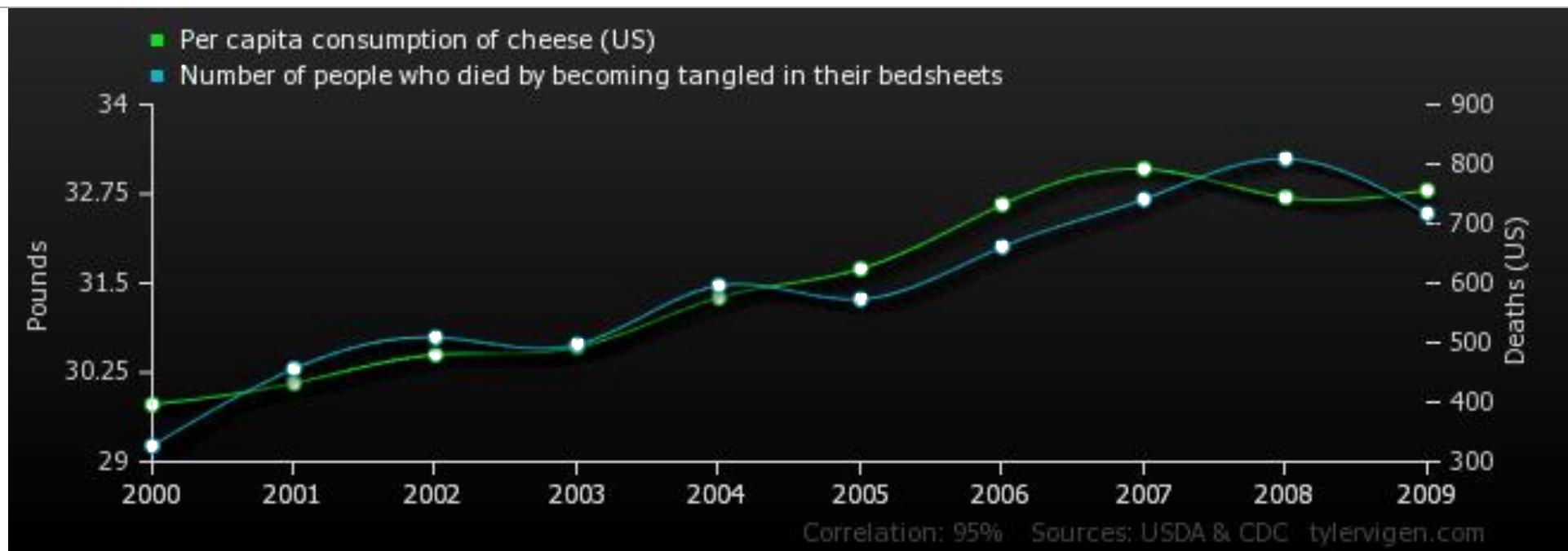


Rock Music Vs Oil?



Hubbert Peak Theory

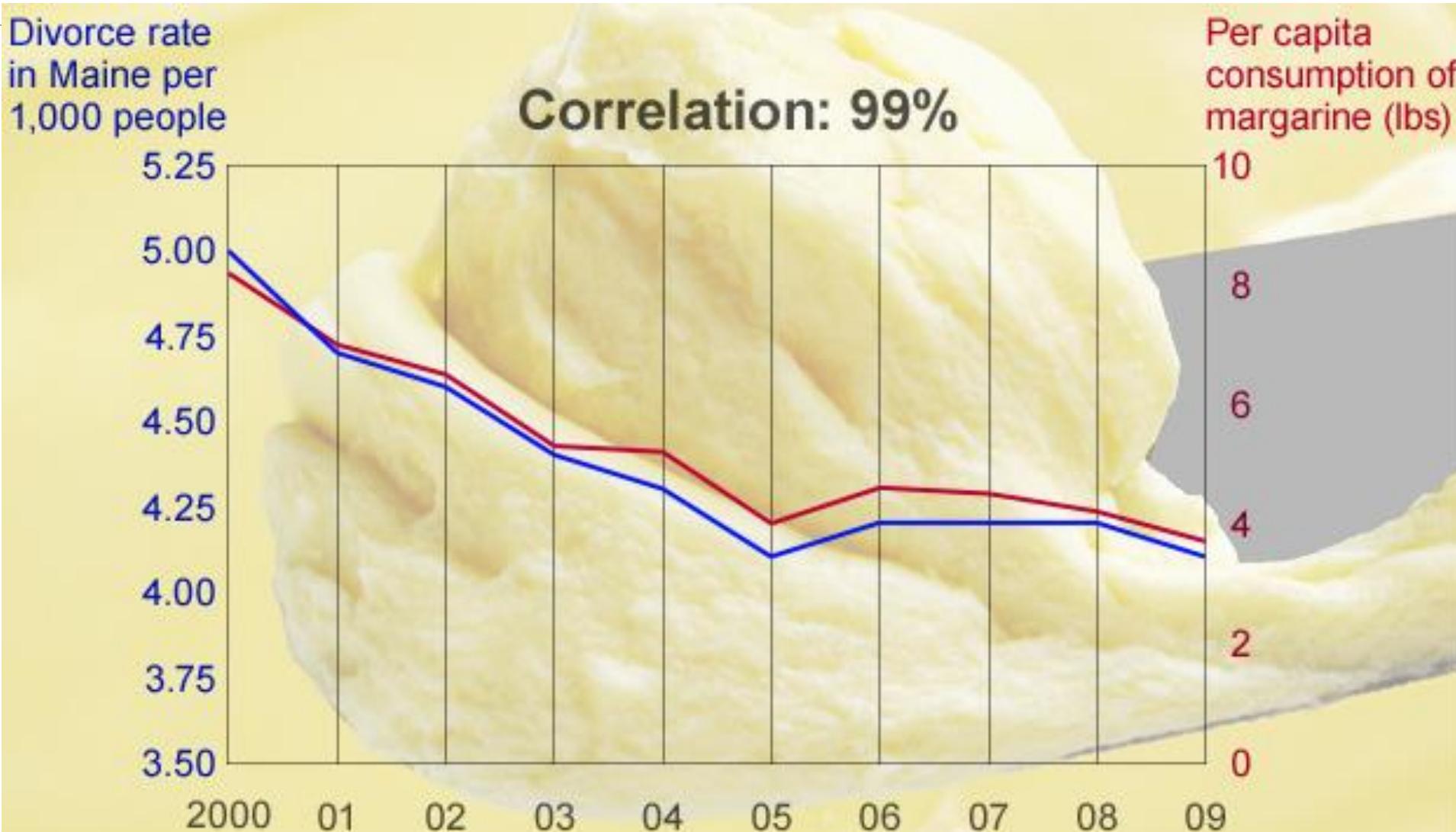
Tell your friends!



	<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) Pounds (USDA)</i>	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 Deaths (US) (CDC)</i>	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

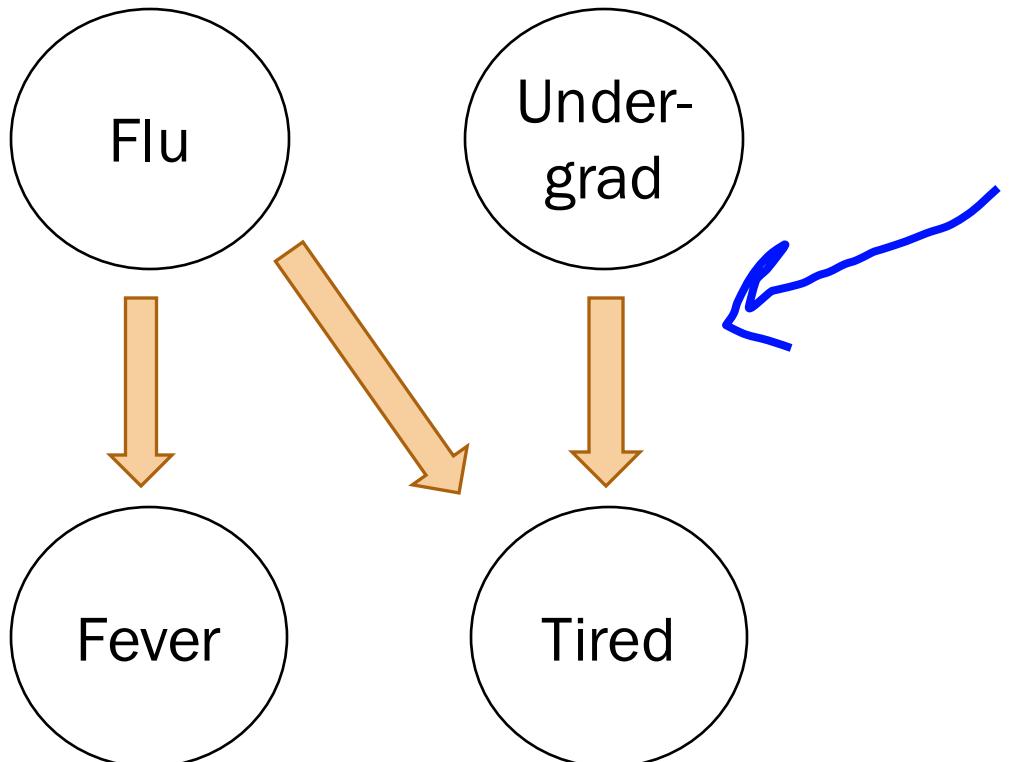
1. How do people actually define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

What haven't we talked about?

Machine Learning (last section of CS109)

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

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



1. Learn this from data

2. Learn this from data

$$\begin{aligned} P(F_{ev} = 1 | F_{lu} = 1) &= 0.9 \\ P(F_{ev} = 1 | F_{lu} = 0) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(T = 1 | F_{lu} = 0, U = 0) &= 0.1 \\ P(T = 1 | F_{lu} = 0, U = 1) &= 0.8 \\ P(T = 1 | F_{lu} = 1, U = 0) &= 0.9 \\ P(T = 1 | F_{lu} = 1, U = 1) &= 1.0 \end{aligned}$$