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.

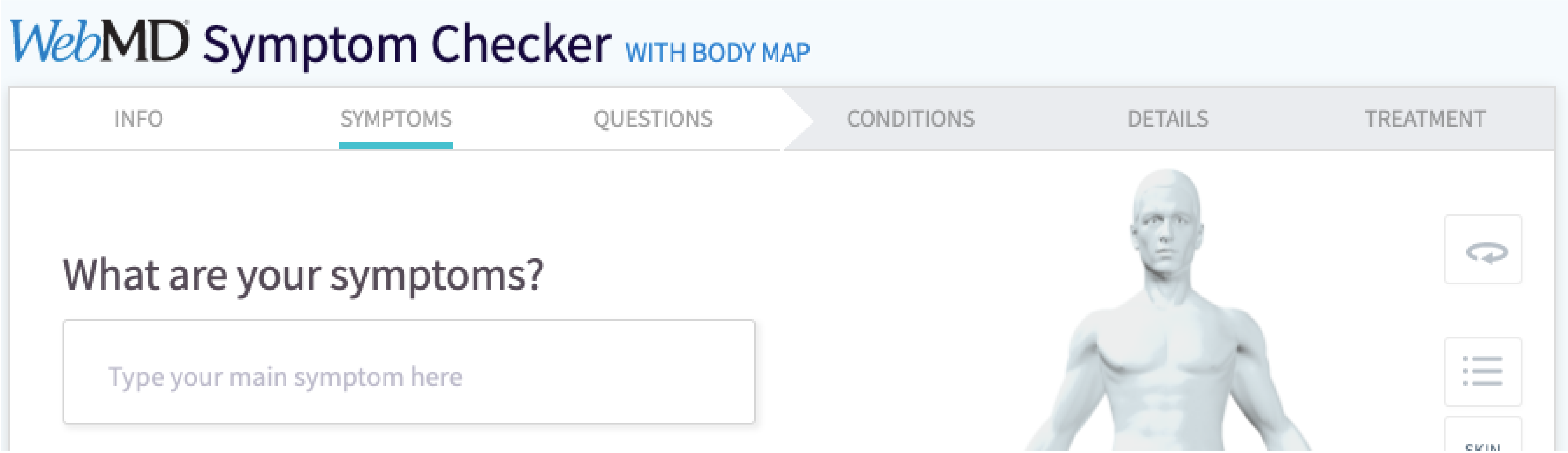
(5) Leave phones at home if possible! If not – we will collect them before exam starts.

(6) Review session on Friday at 4:30pm (location TBD). No lecture on Monday 2/9

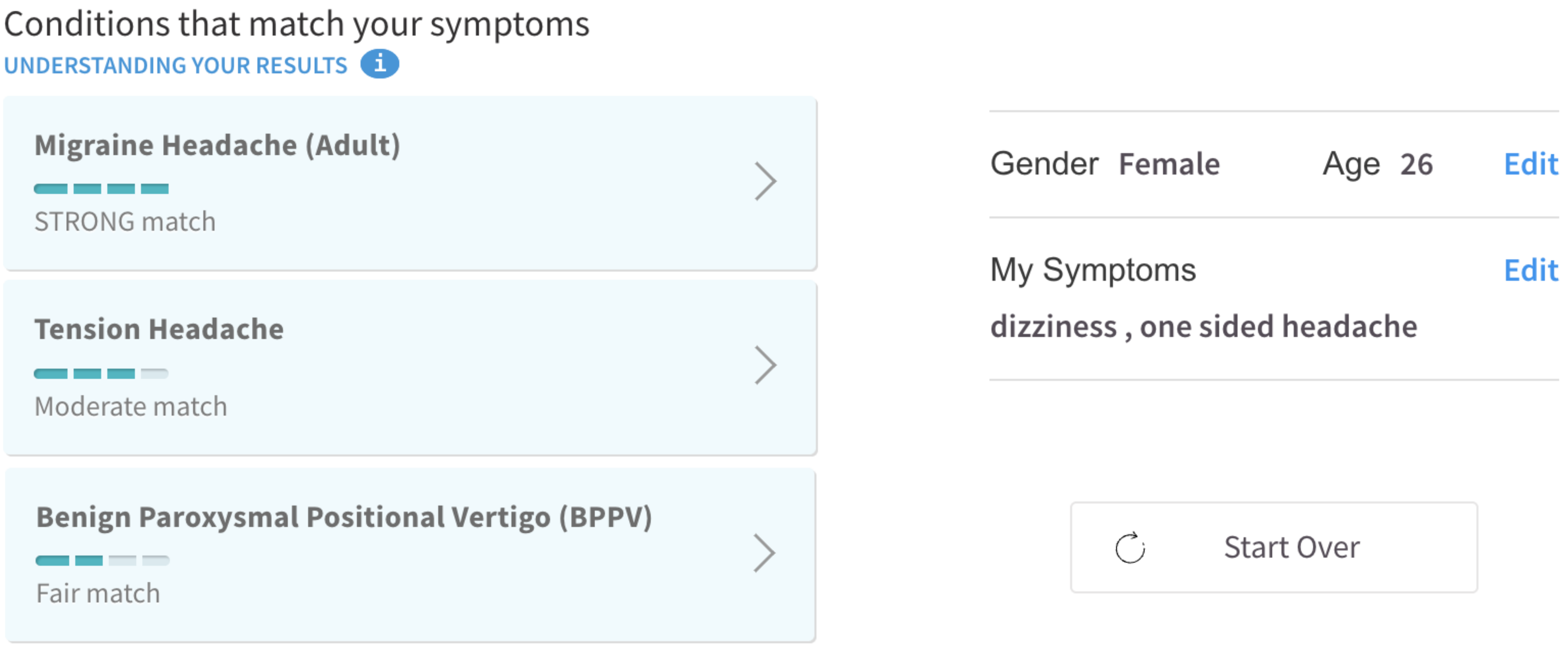
Why You Need a Model

WebMD®

Why You Need a Model



Multiple Random Variables. Start of Digital Revolution



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.

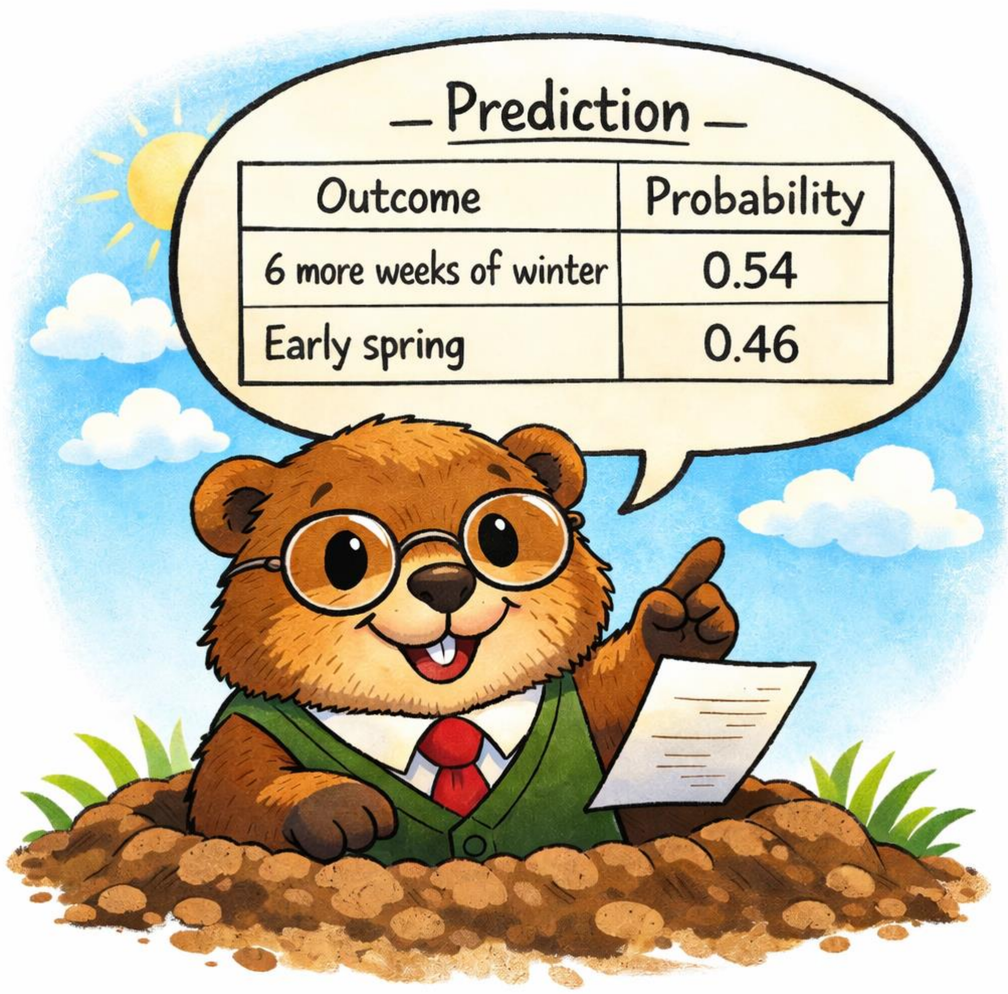
Based on historical data:

- 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.

$$P(W \mid S) = \frac{P(S \mid W)P(W)}{P(S \mid W)P(W) + P(S \mid W^c)P(W^c)}$$

$$= \frac{0.7 \cdot 0.5}{0.7 \cdot 0.5 + 0.6 \cdot 0.5} = \frac{0.35}{0.65} \approx 0.54$$



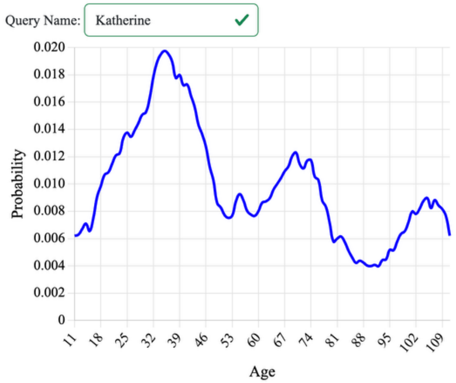
At this point you know inference with two random variables

Today: Five New Real + Exciting Problems

Age from C14

Updated Delivery Prob

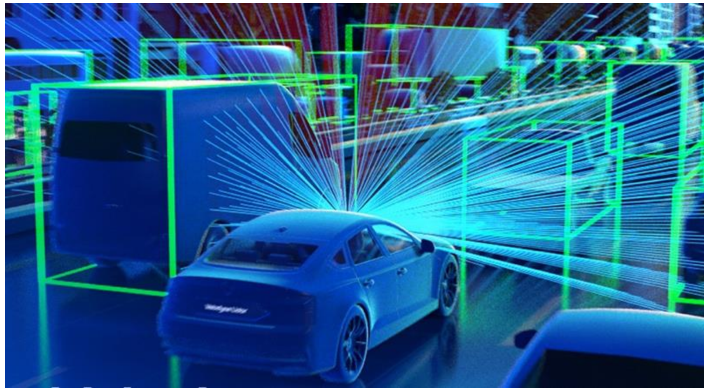
Age from Name



Hidden Chambers

Stanford Eye Test

Updating Lidar Belief



Today: Five New Real + Exciting Problems

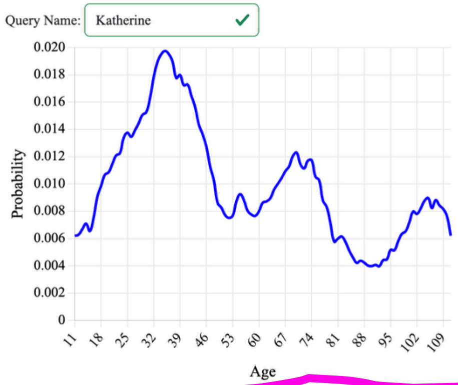
Simple Joint

Interesting Likelihood

Age from C14

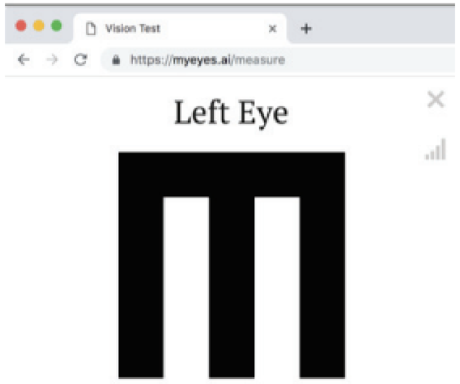
Updated Delivery Prob

Age from Name



Hidden Chambers

Stanford Eye Test

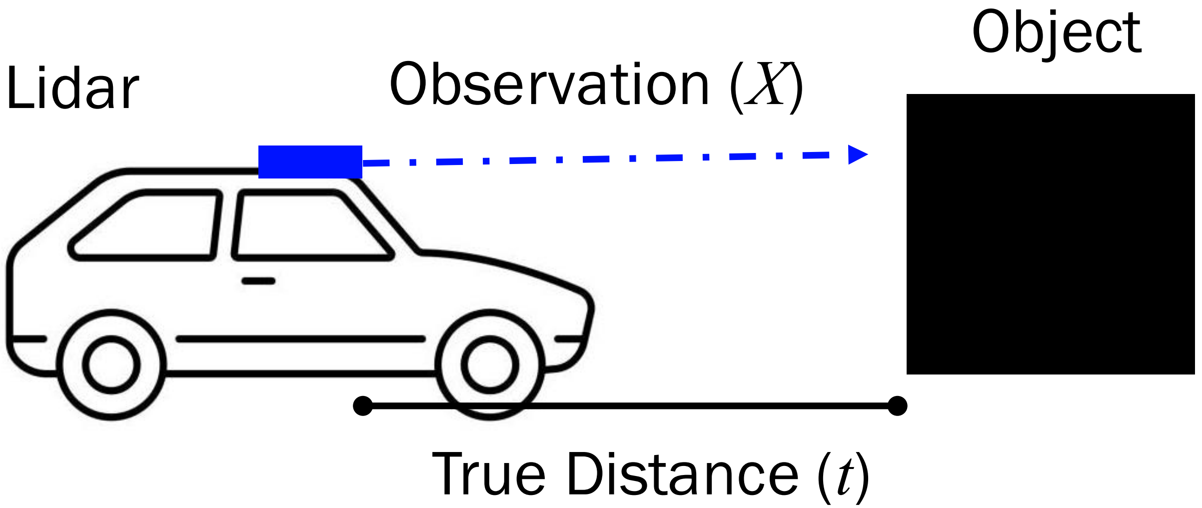


Repeat Observations

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: $X = 4$

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

Bayes theorem

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

Plug in normal PDF

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

Drop constants + Combine exponents

$$\propto \exp\left[-\frac{1}{2}(t^2 - 6t)\right]$$

After simplifying

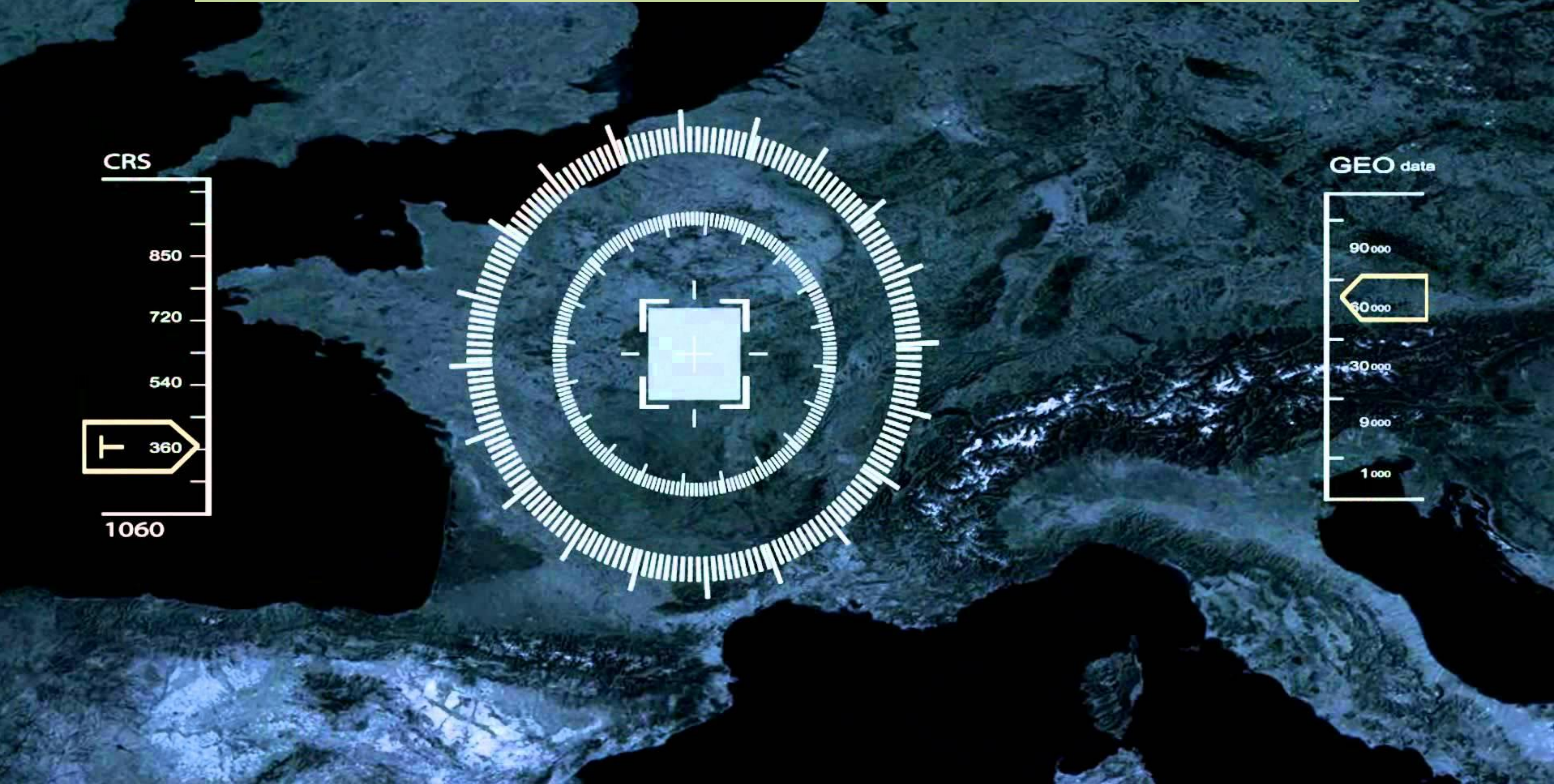
$$\propto \exp\left[-\frac{1}{2}(t-3)^2\right] \qquad \sim N(\mu = 3, \sigma^2 = 1)$$

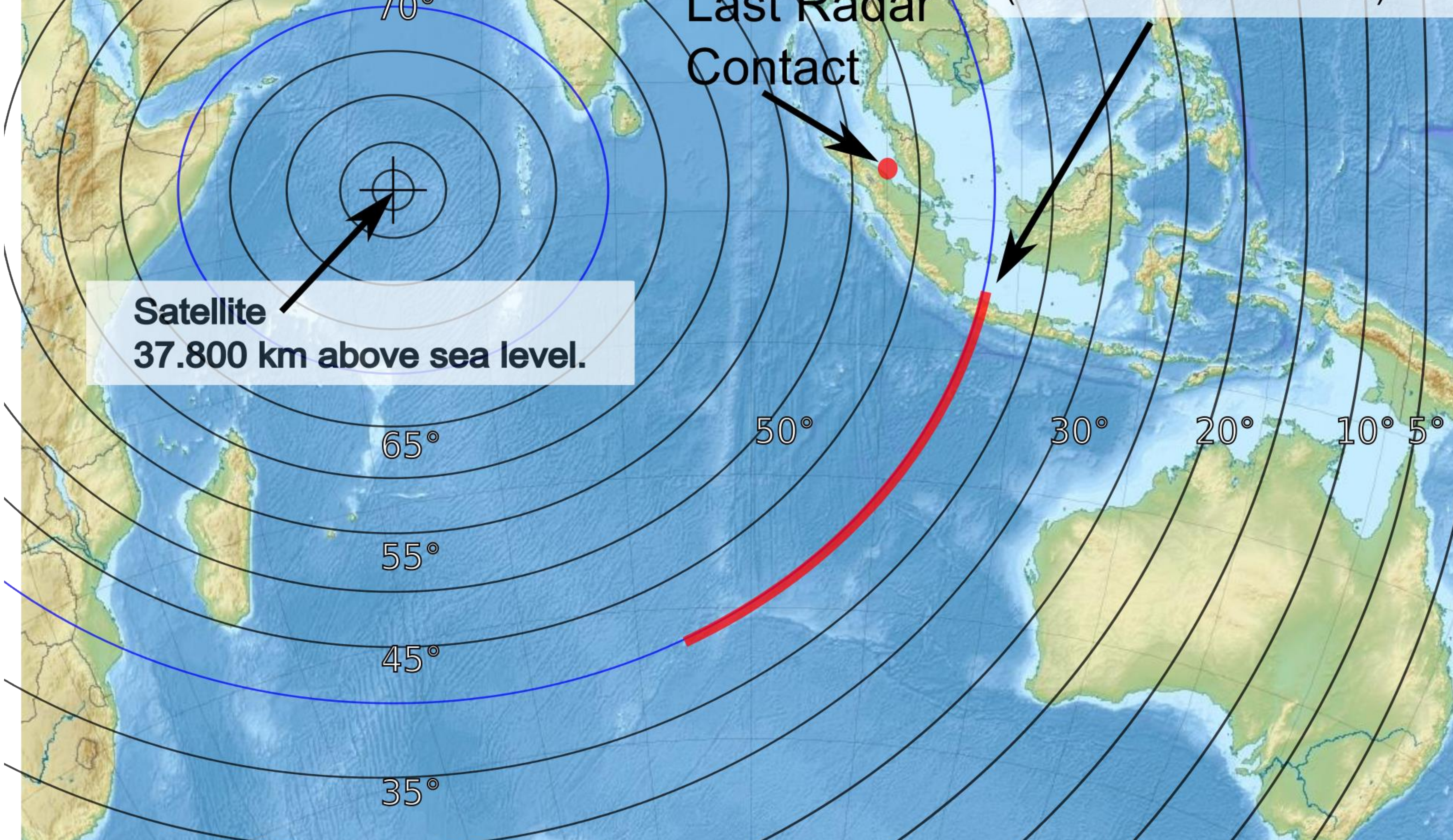
Complete the square

Optional but neat

Pro tip: keep the -1/2 factored out

Tracking in 2D Space?



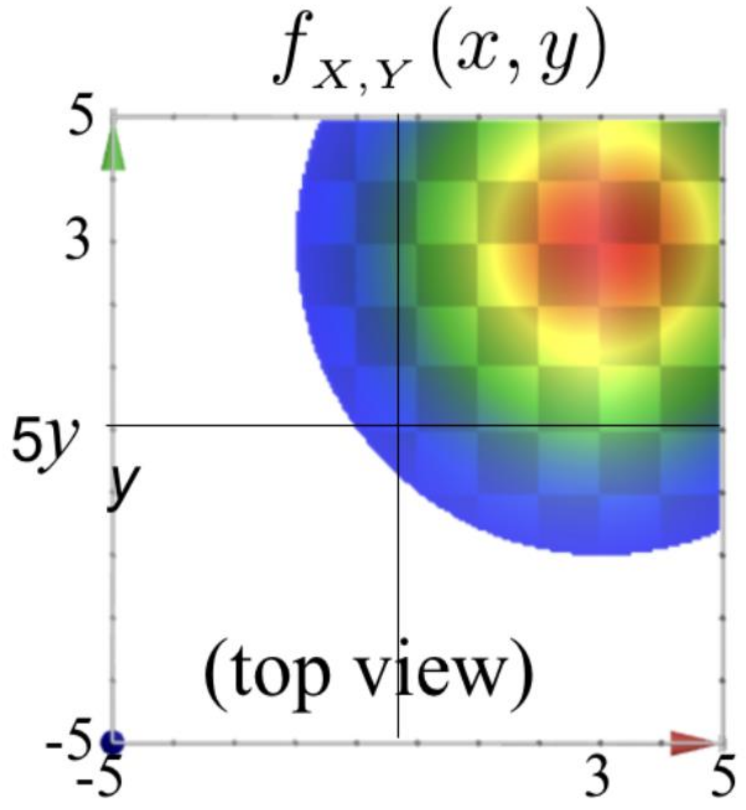


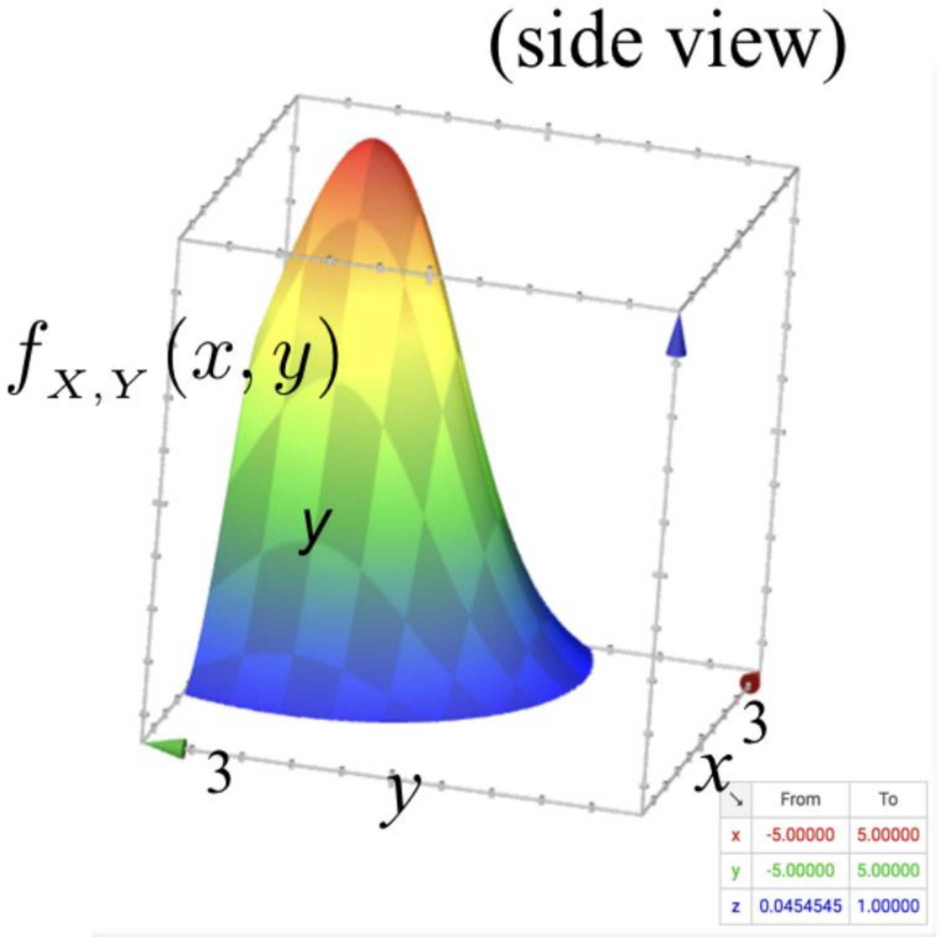
Last Radar
Contact
Satellite
37.800 km above sea level.
70°
65°
55°
45°
35°
50°
30°
20°
10°
5°


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

Prior

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

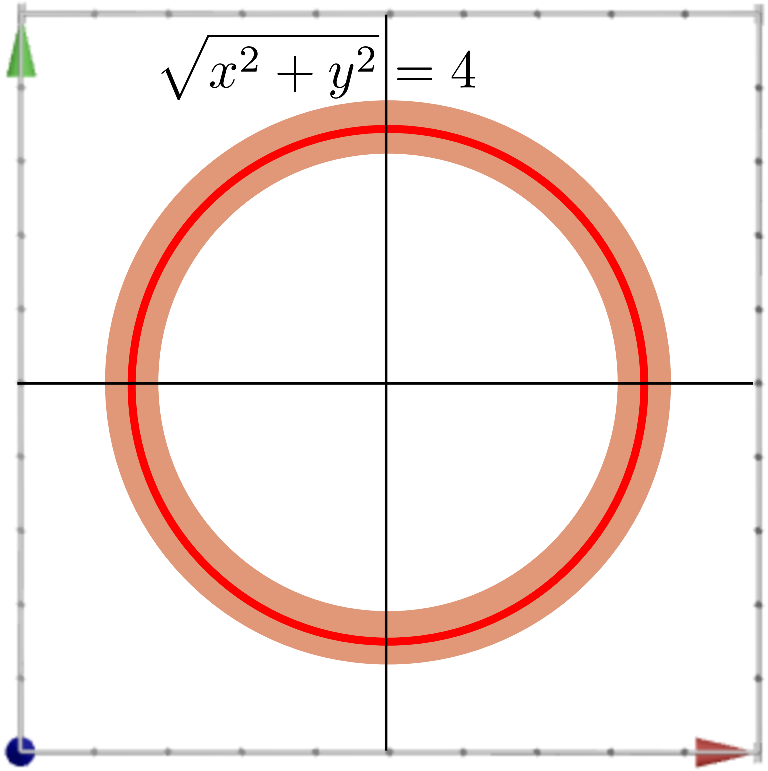


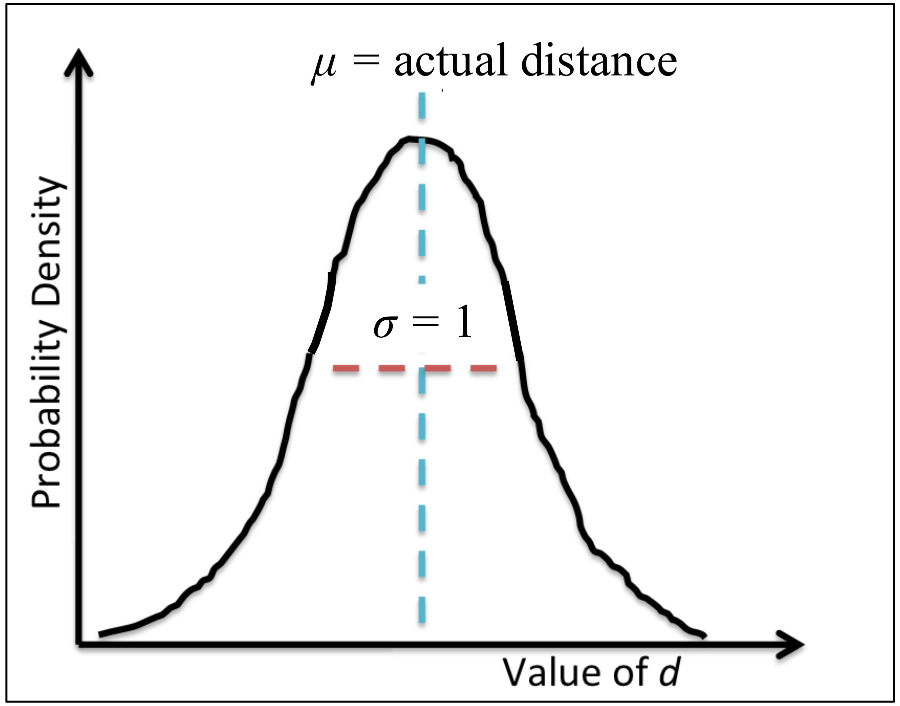


Likelihood

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

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

$$= \frac{f(D=4 \mid X=x,Y=y)\cdot f(X=x,Y=y)}{f(D=4)}$$ Bayes using densities

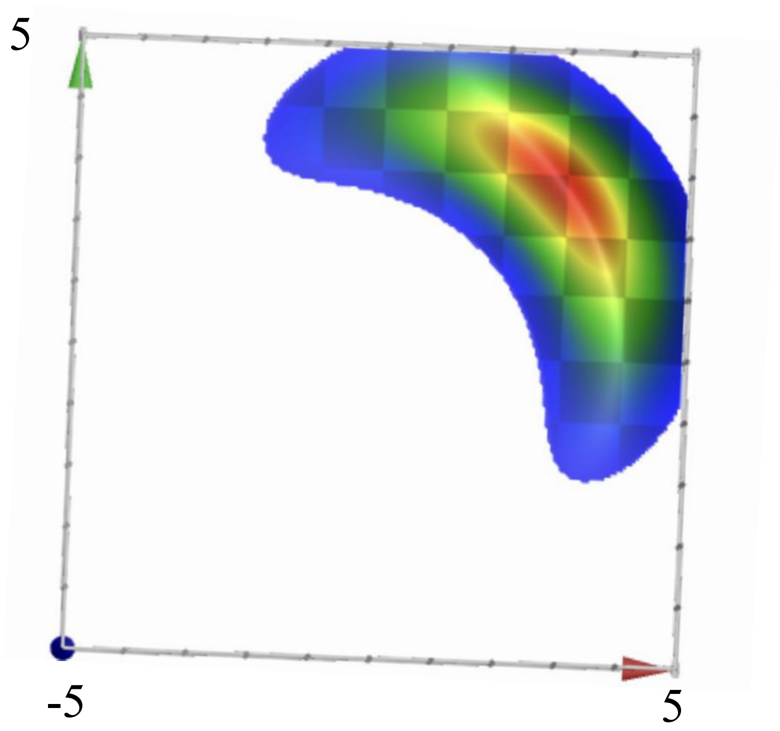
$$= \frac{K_1 \cdot e^{-\frac{(4-\sqrt{x^2+y^2})^2}{2}} \cdot K_2 \cdot e^{-\frac{(x-3)^2+(y-3)^2}{8}}}{f(D=4)}$$ Substitute

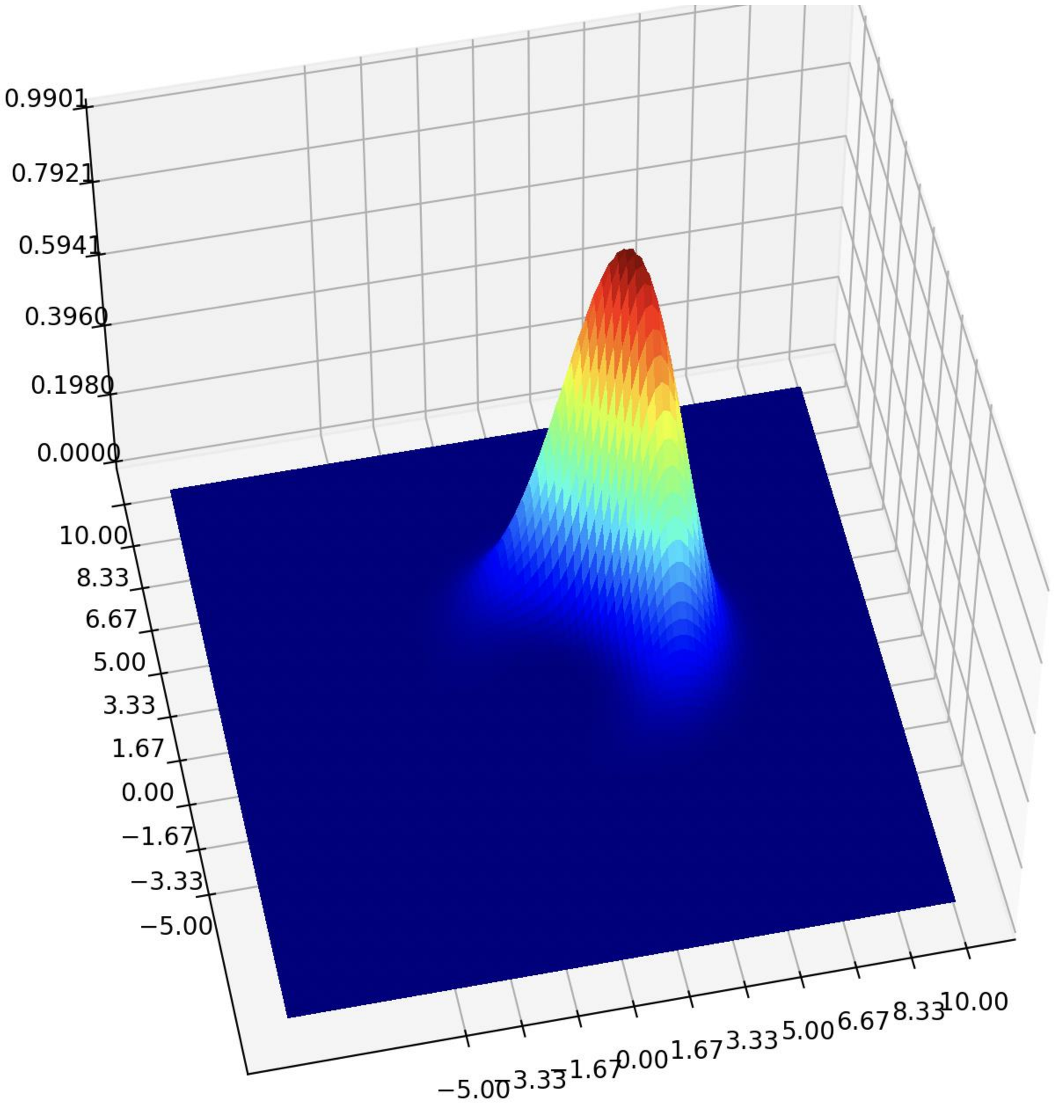
$$= \frac{K_1 \cdot K_2}{f(D=4)} \cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]}$$ $f(D=4)$ is a constant w.r.t. (x,y)

$$= K_3 \cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2} + \frac{(x-3)^2+(y-3)^2}{8}\right]}$$ K_3 is a new constant

Tracking Posterior

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



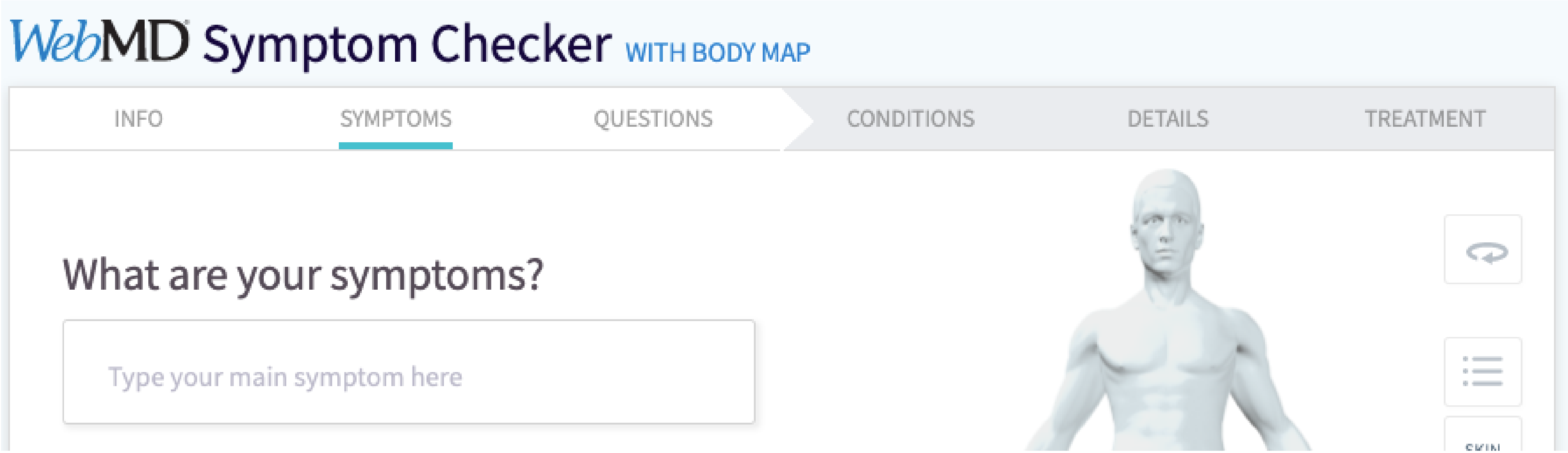


Many real world problems have way more than two random variables…

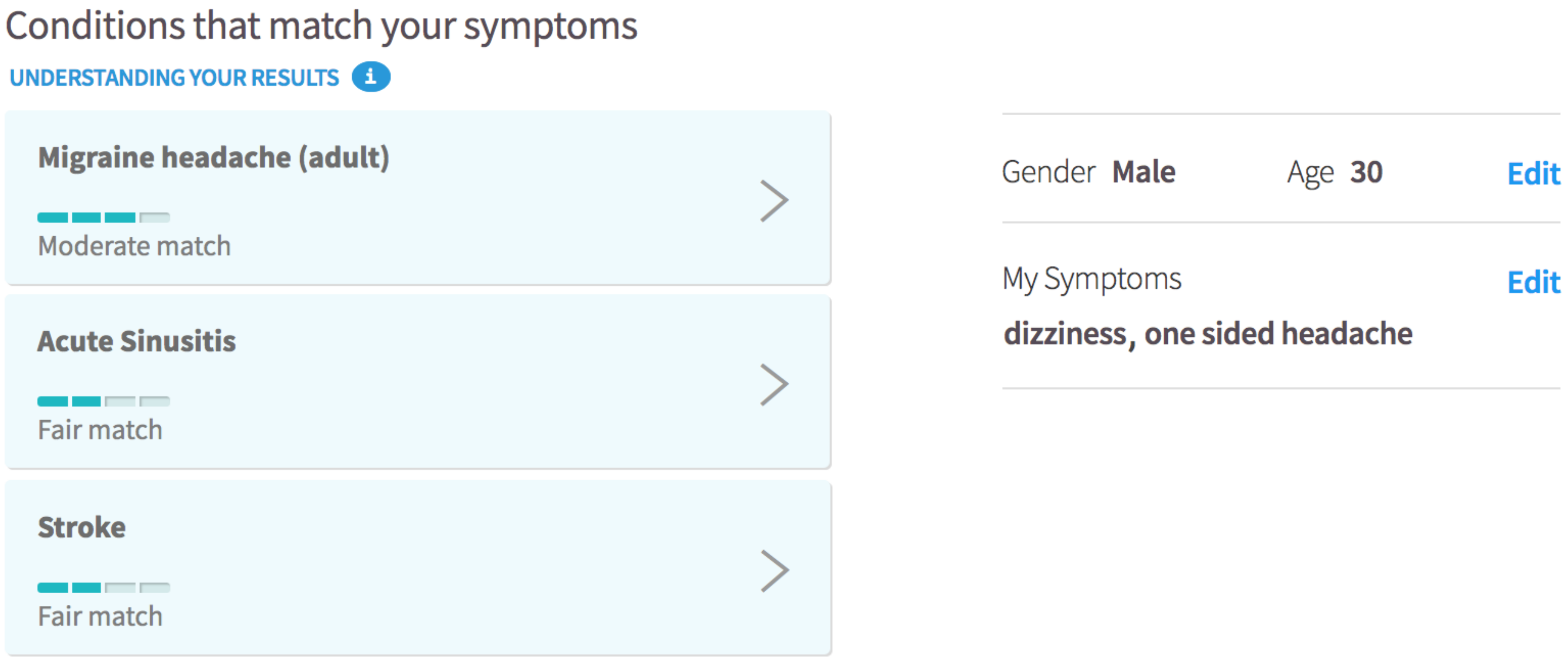
Why You Need a Model

WebMD®

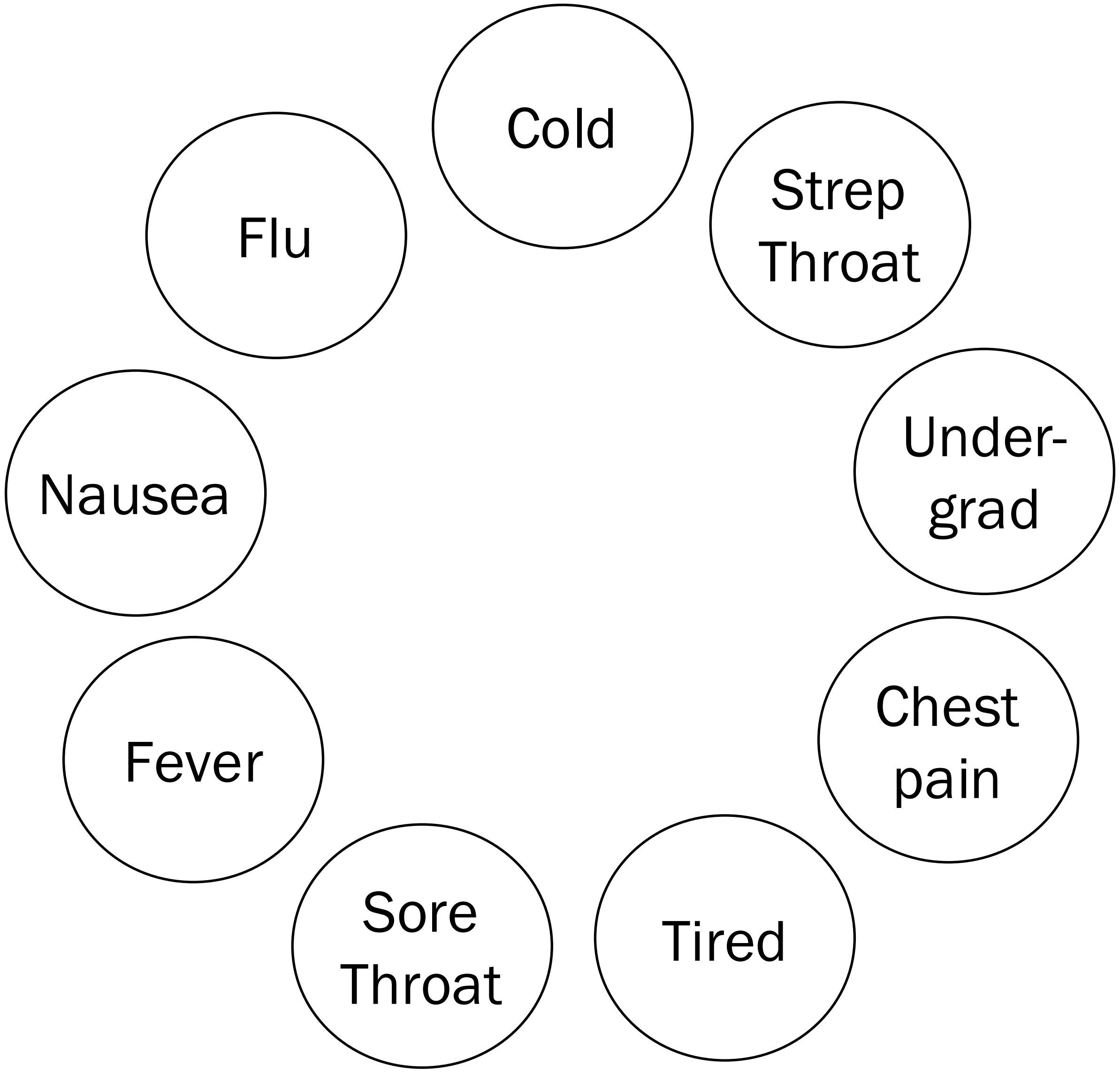
Why You Need a Model



Multiple Random Variables. Start of Digital Revolution



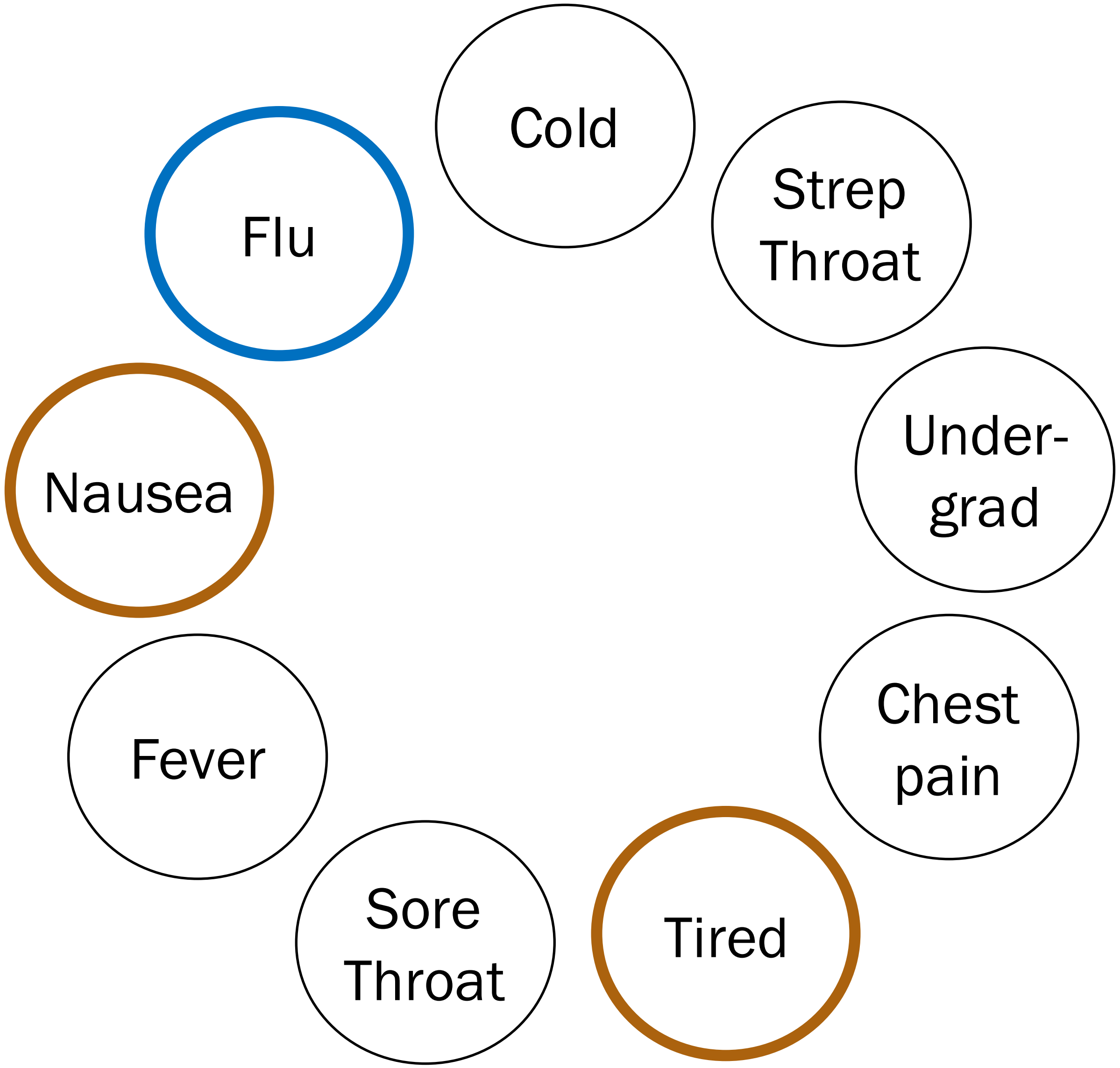
Challenge #1: Many Inference Questions



Inference question:

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

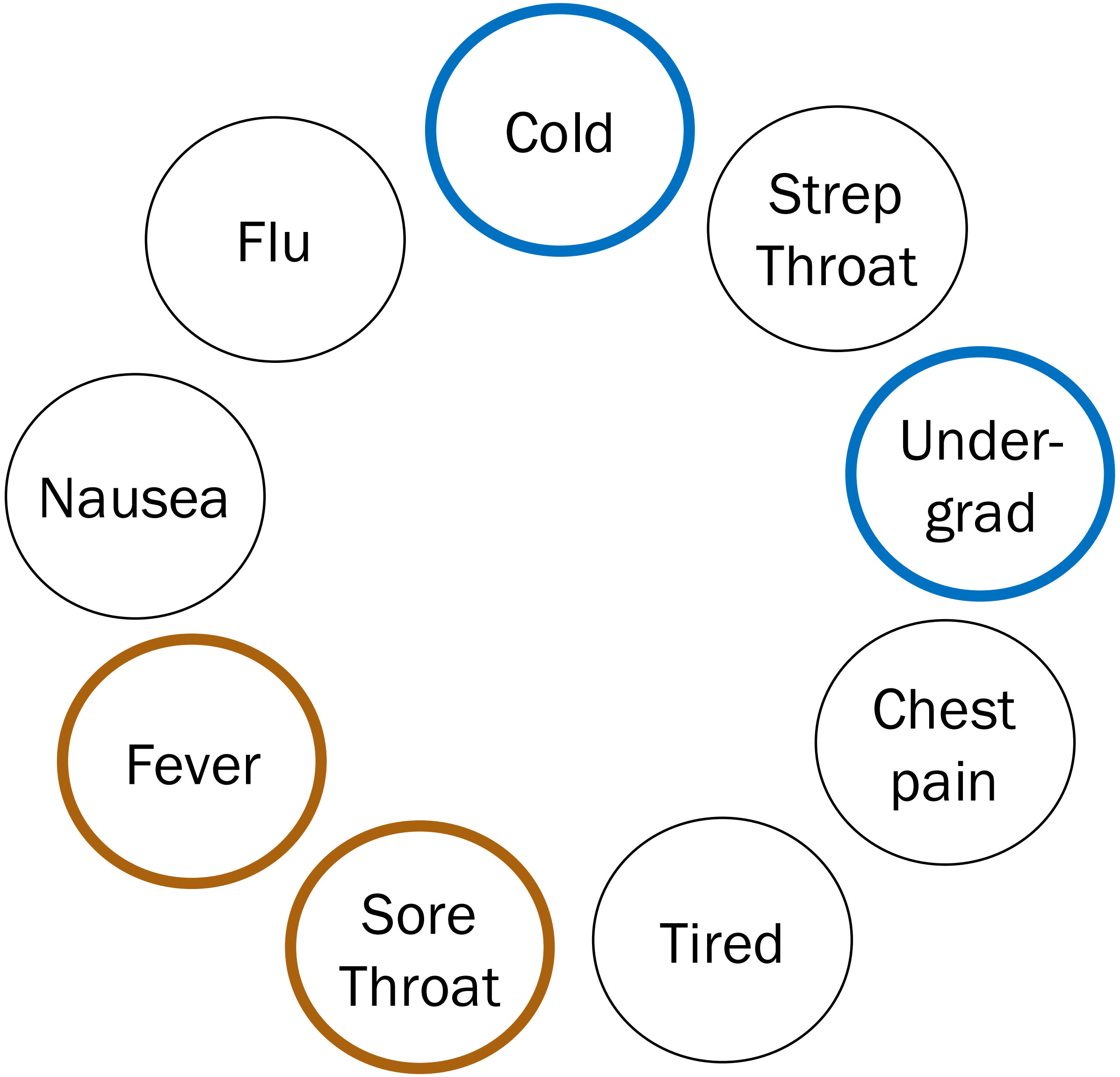
Challenge #1: Many Inference Questions



One inference question:

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

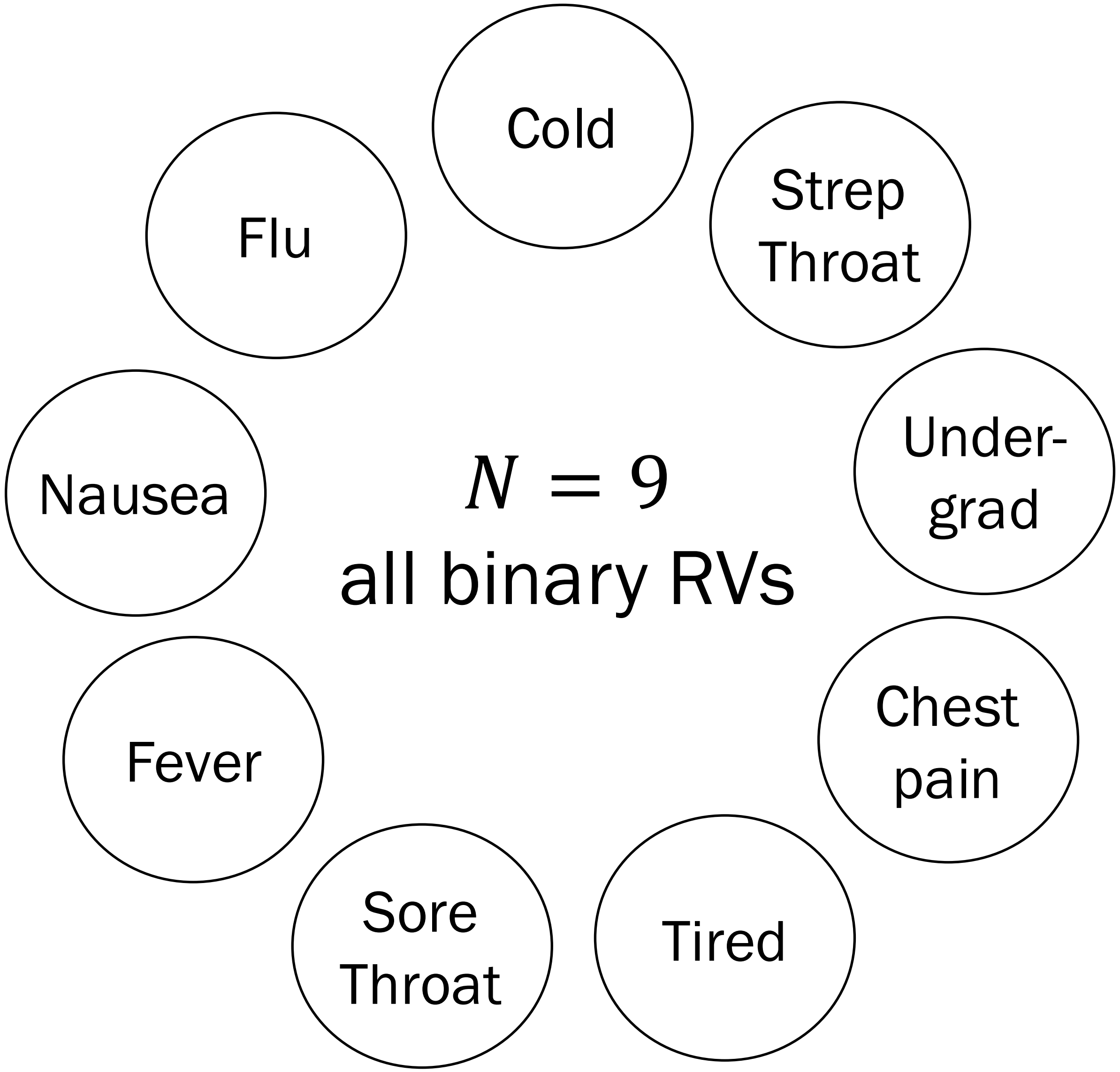
Challenge #1: Many Inference Questions



Another inference question:

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

Challenge #2: Joint is Large



$N = 9$

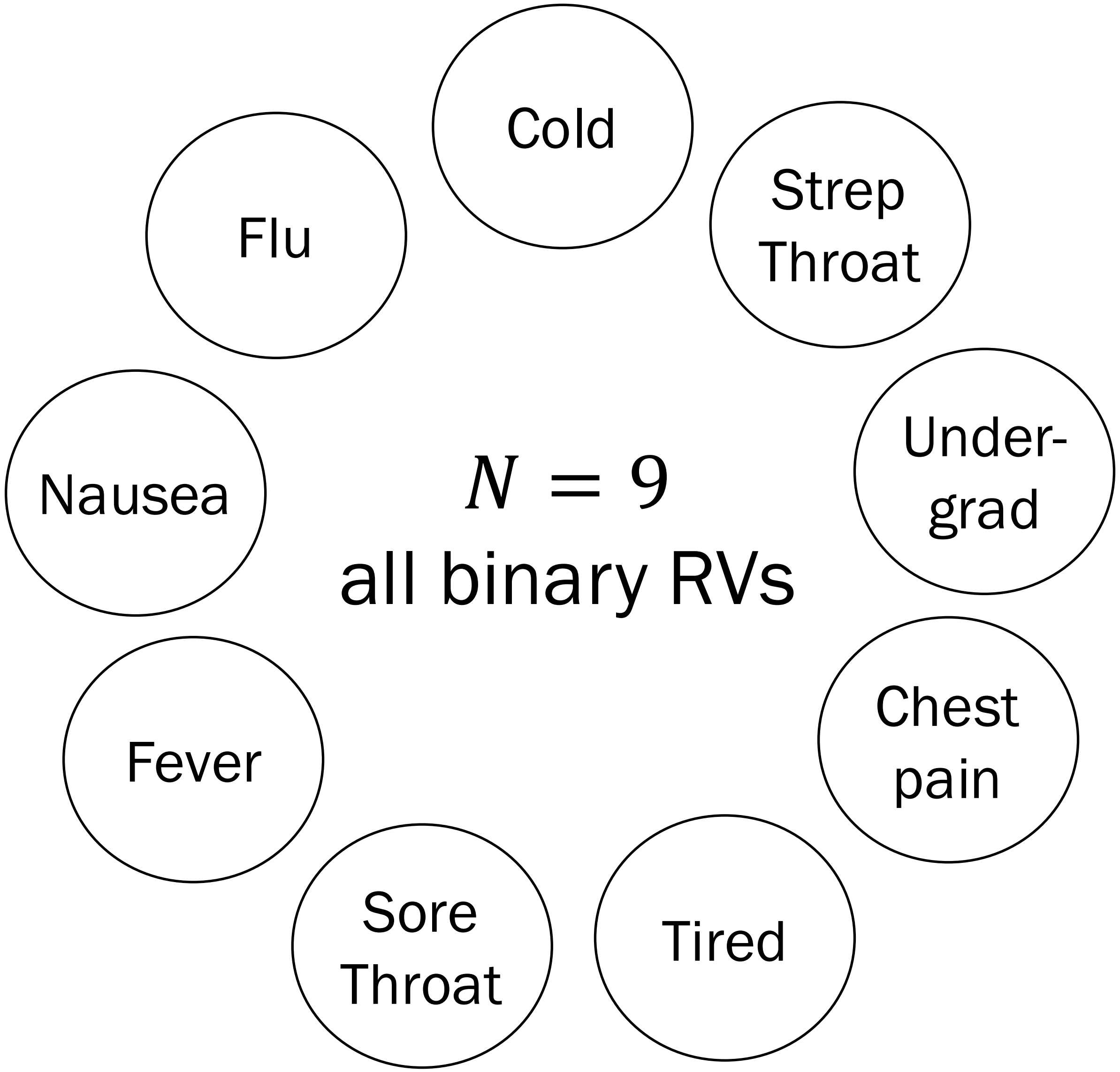
all binary RVs

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



$N = 9$
all binary RVs

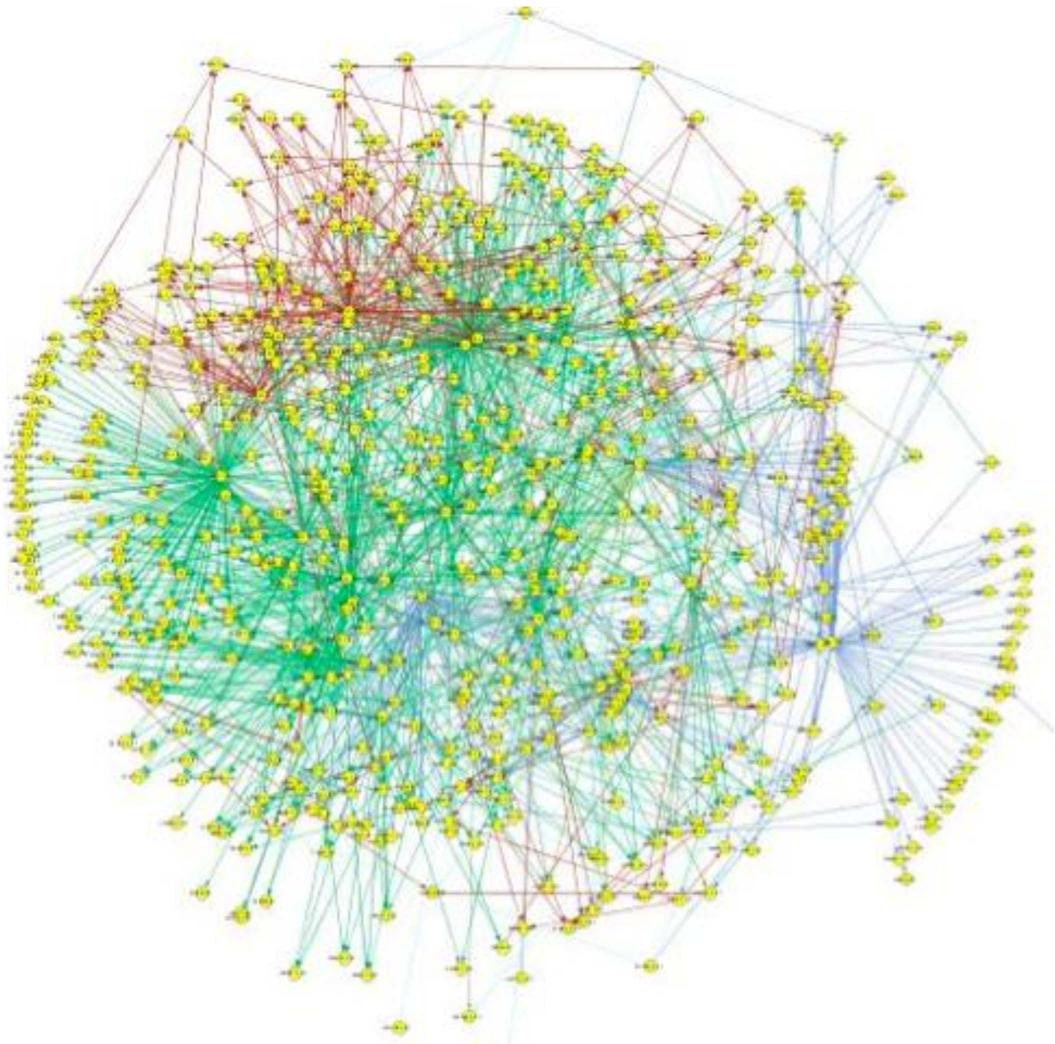
If we knew the **joint distribution**, we can answer all probabilistic inference questions.

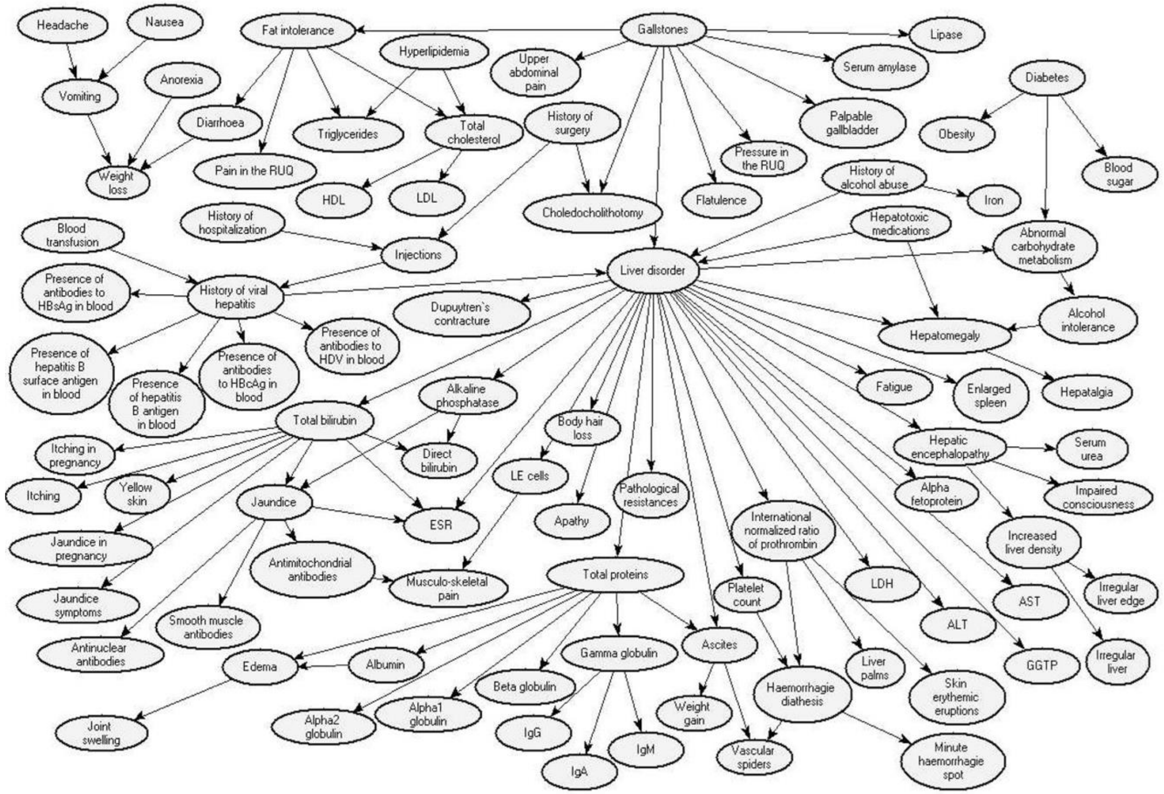
What is the size of the joint probability table?

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?

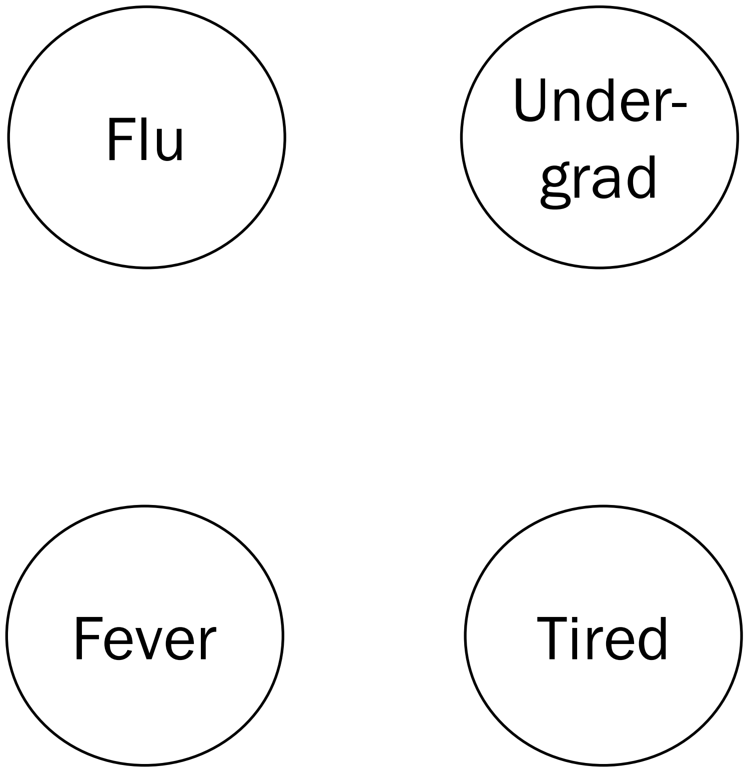
Three Guiding Questions

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

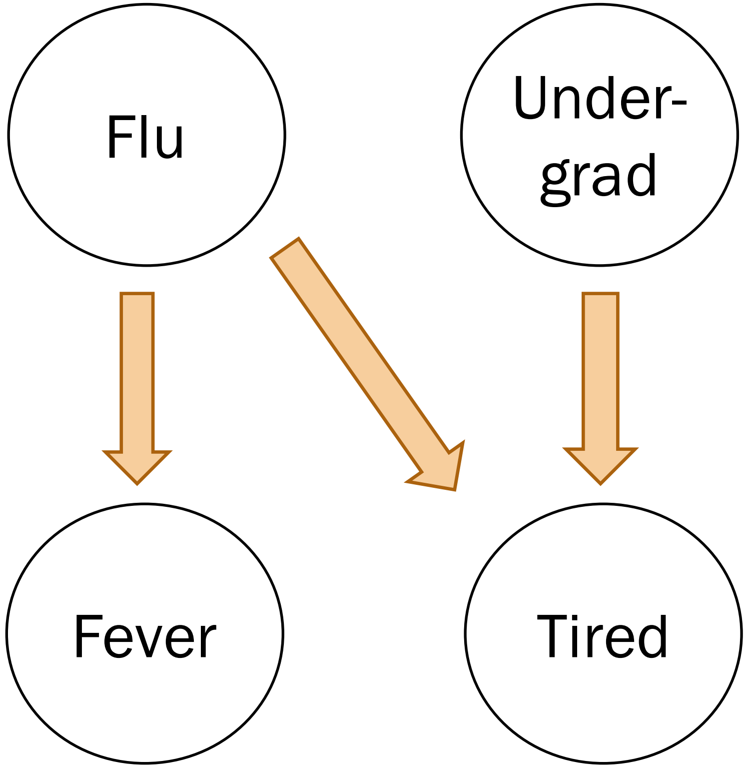


Great! Just specify $2^4 = 16$ joint probabilities…?

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

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

Constructing a Bayesian Network



What would a Stanford flu expert do?

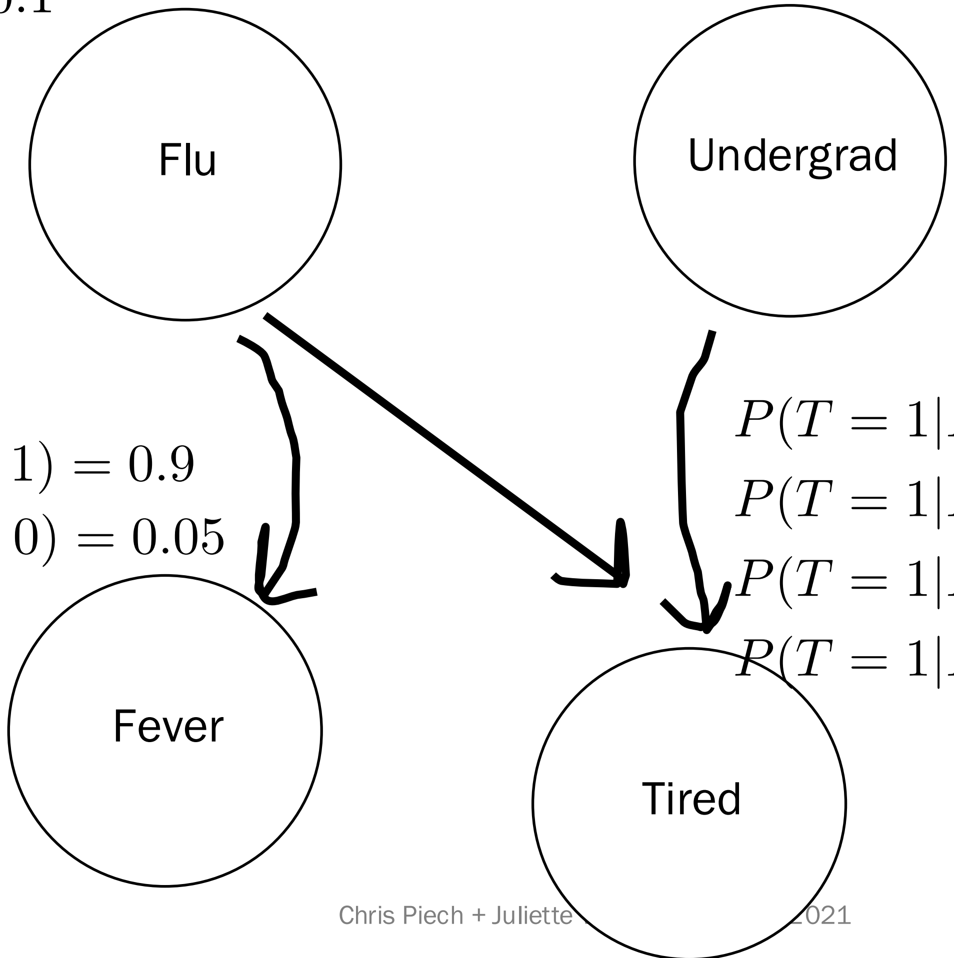
1. ✅ Describe the causality.
2. Provide $P(\text{values}|\text{causal parents})$ for each random variable
3. Implicitly assumes independences.

Recall: Probabilistic Model

✅ 2. Provide P(values|causal parents) for each random variable

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

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



$$P(Fev = 1|Flu = 1) = 0.9$$
$$P(Fev = 1|Flu = 0) = 0.05$$

$$P(T = 1|Flu = 0, U = 0) = 0.1$$
$$P(T = 1|Flu = 0, U = 1) = 0.8$$
$$P(T = 1|Flu = 1, U = 0) = 0.9$$
$$P(T = 1|Flu = 1, U = 1) = 1.0$$

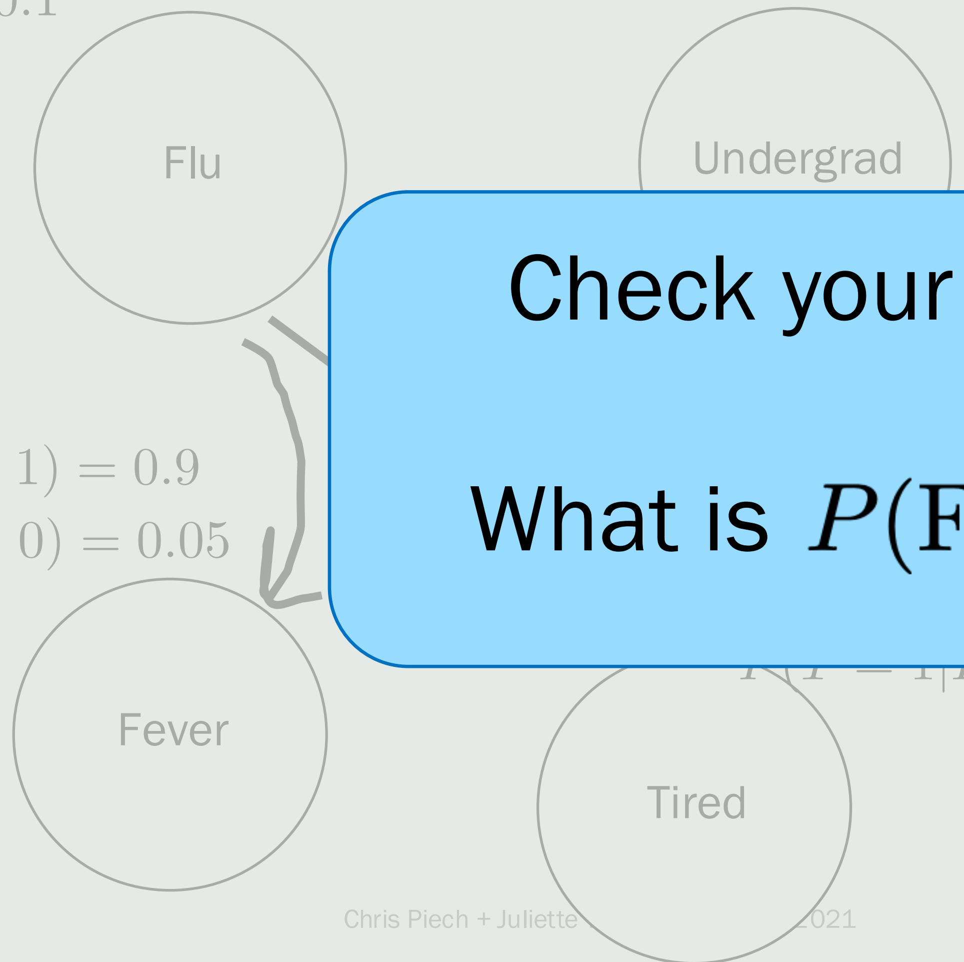
Recall: Probabilistic Model

2. Provide P(values|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”

```python
def make_sample():
    """
    Make Sample
    -------------------
    chose a single sample from the joint distribution
    """
    # prior on causal factors
    flu = bern(0.1)
    undergrad = bern(0.8)

    # choose fever based on flue
    if flu == 1: fever = bern(0.9)
    else:        fever = bern(0.05)

    # choose tired based on (undergrade and flu)
    if undergrad == 1 and flu == 1:   tired = bern(1.0)
    elif undergrad == 1 and flu == 0: tired = bern(0.8)
    elif undergrad == 0 and flu == 1: tired = bern(0.9)
    else:                             tired = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return {
            'flu':flu,
            'undergrad':undergrad,
            'fever':fever,
            'tired':tired
        }
```

```python
def make_sample():
    """
    Make Sample
    -------------------
    chose a single sample from the joint distribution
    """
    # prior on causal factors
    flu = bern(0.1)
    undergrad = bern(0.8)

    # choose fever based on flue
    if flu == 1: fever = bern(0.9)
    else:        fever = bern(0.05)

    # choose tired based on (undergrade and flu)
    if undergrad == 1 and flu == 1:   tired = bern(1.0)
    elif undergrad == 1 and flu == 0: tired = bern(0.8)
    elif undergrad == 0 and flu == 1: tired = bern(0.9)
    else:                             tired = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return {
            'flu':flu,
            'undergrad':undergrad,
            'fever':fever,
            'tired':tired
        }
```

```python
def make_sample():
    """
    Make Sample
    -------------------
    chose a single sample from the joint distribution
    """
    # prior on causal factors
    flu = bern(0.1)
    undergrad = bern(0.8)

    # choose fever based on flue
    if flu == 1: fever = bern(0.9)
    else:        fever = bern(0.05)

    # choose tired based on (undergrade and flu)
    if undergrad == 1 and flu == 1:   tired = bern(1.0)
    elif undergrad == 1 and flu == 0: tired = bern(0.8)
    elif undergrad == 0 and flu == 1: tired = bern(0.9)
    else:                             tired = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return {
        'flu':flu,
        'undergrad':undergrad,
        'fever':fever,
        'tired':tired
    }
```

```python
def make_sample():
    """
    Make Sample
    -------------------
    chose a single sample from the joint distribution
    """
    # prior on causal factors
    flu = bern(0.1)
    undergrad = bern(0.8)

    # choose fever based on flue
    if flu == 1: fever = bern(0.9)
    else:        fever = bern(0.05)

    # choose tired based on (undergrade and flu)
    if undergrad == 1 and flu == 1:   tired = bern(1.0)
    elif undergrad == 1 and flu == 0: tired = bern(0.8)
    elif undergrad == 0 and flu == 1: tired = bern(0.9)
    else:                             tired = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return {
            'flu':flu,
            'undergrad':undergrad,
            'fever':fever,
            'tired':tired
        }
```

```python
def make_sample():
    """
    Make Sample
    -------------------
    chose a single sample from the joint distribution
    """
    # prior on causal factors
    flu = bern(0.1)
    undergrad = bern(0.8)

    # choose fever based on flue
    if flu == 1: fever = bern(0.9)
    else:        fever = bern(0.05)

    # choose tired based on (undergrade and flu)
    if undergrad == 1 and flu == 1:   tired = bern(1.0)
    elif undergrad == 1 and flu == 0: tired = bern(0.8)
    elif undergrad == 0 and flu == 1: tired = bern(0.9)
    else:                             tired = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return {
          'flu':flu,
          'undergrad':undergrad,
          'fever':fever,
          'tired':tired
      }
```

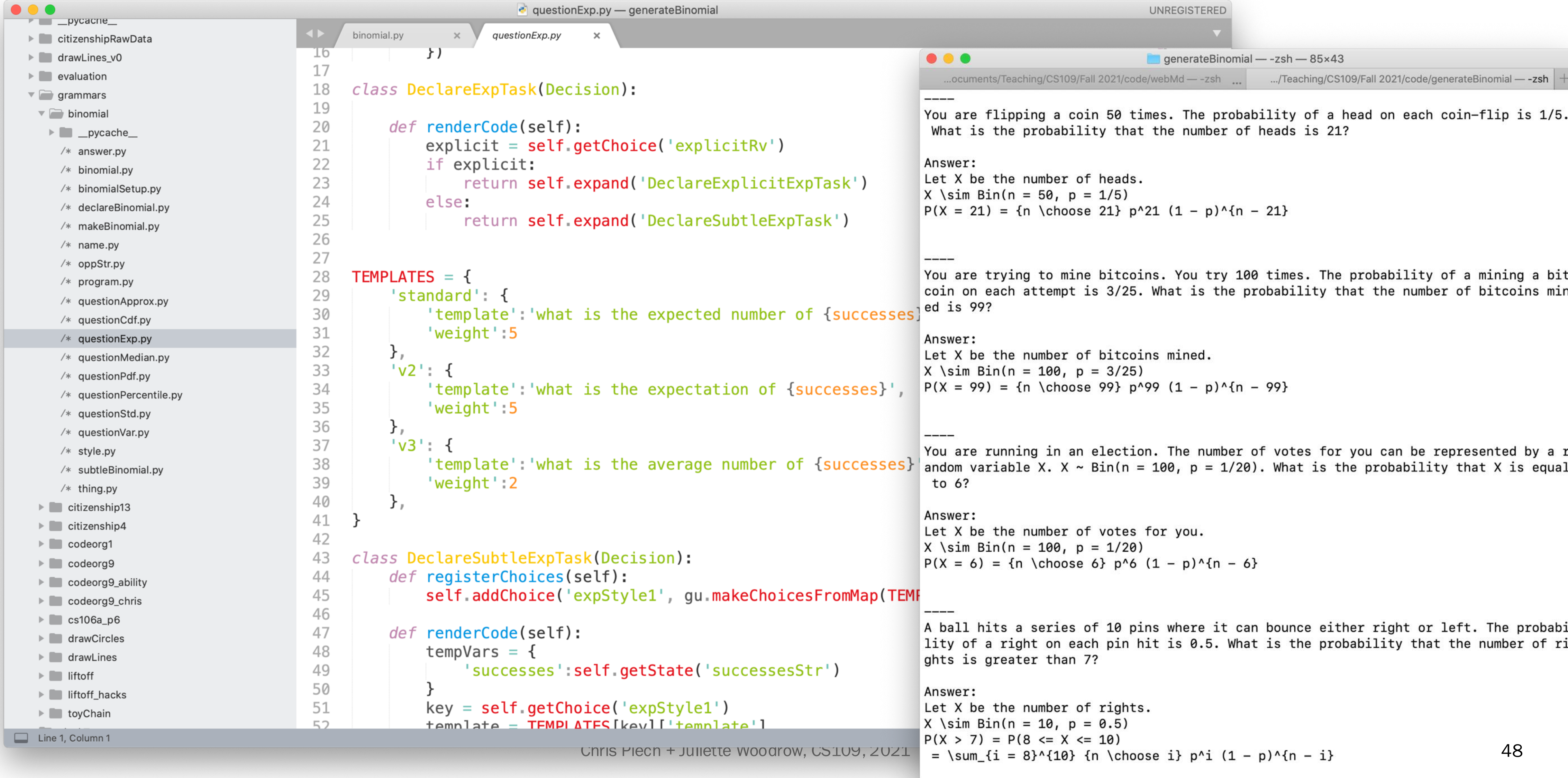
Can You Sample from the Joint?

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

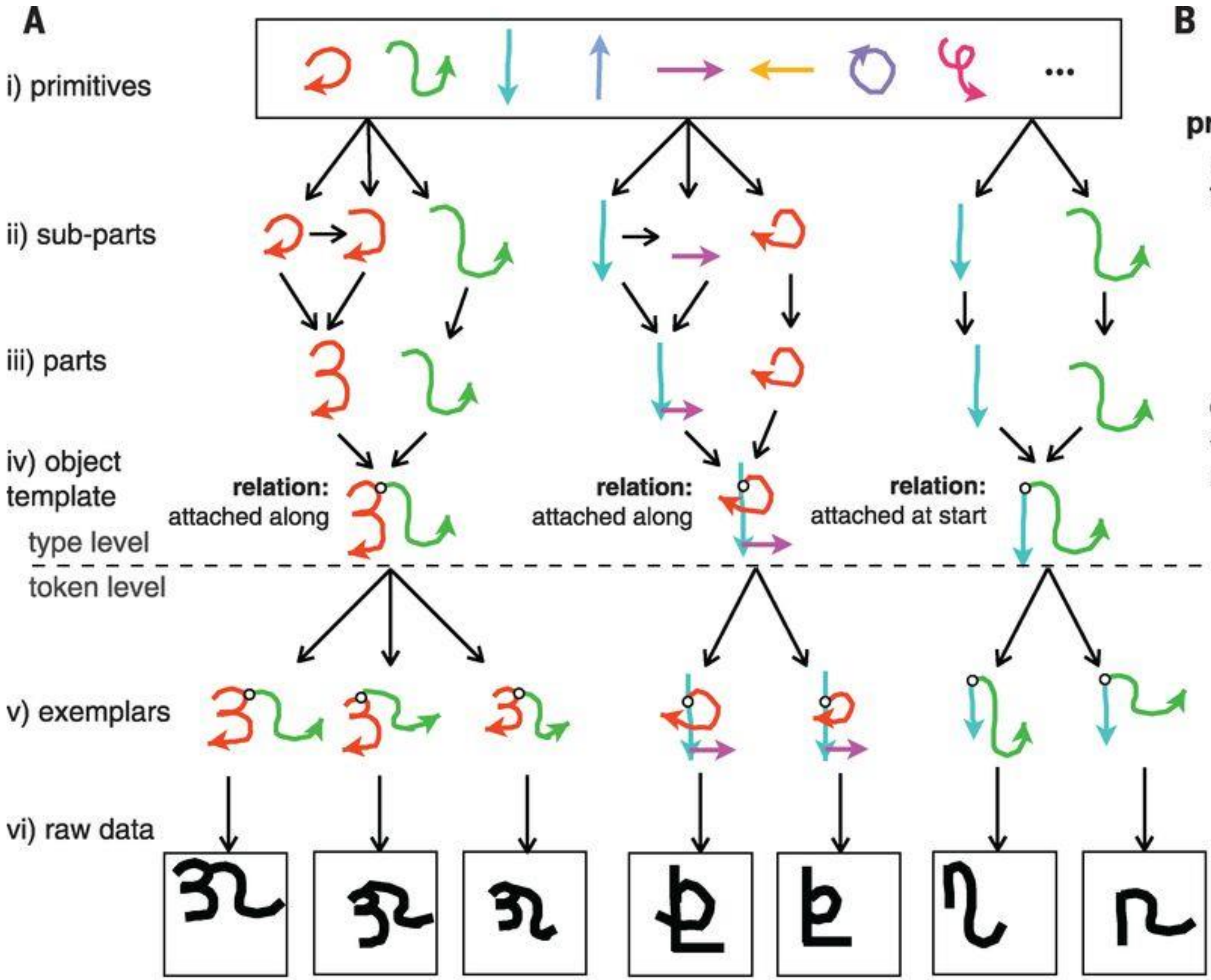
Make a *Generative* Model

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

Generative Model of Binomial Questions



Generative Model of Hand Written Letters

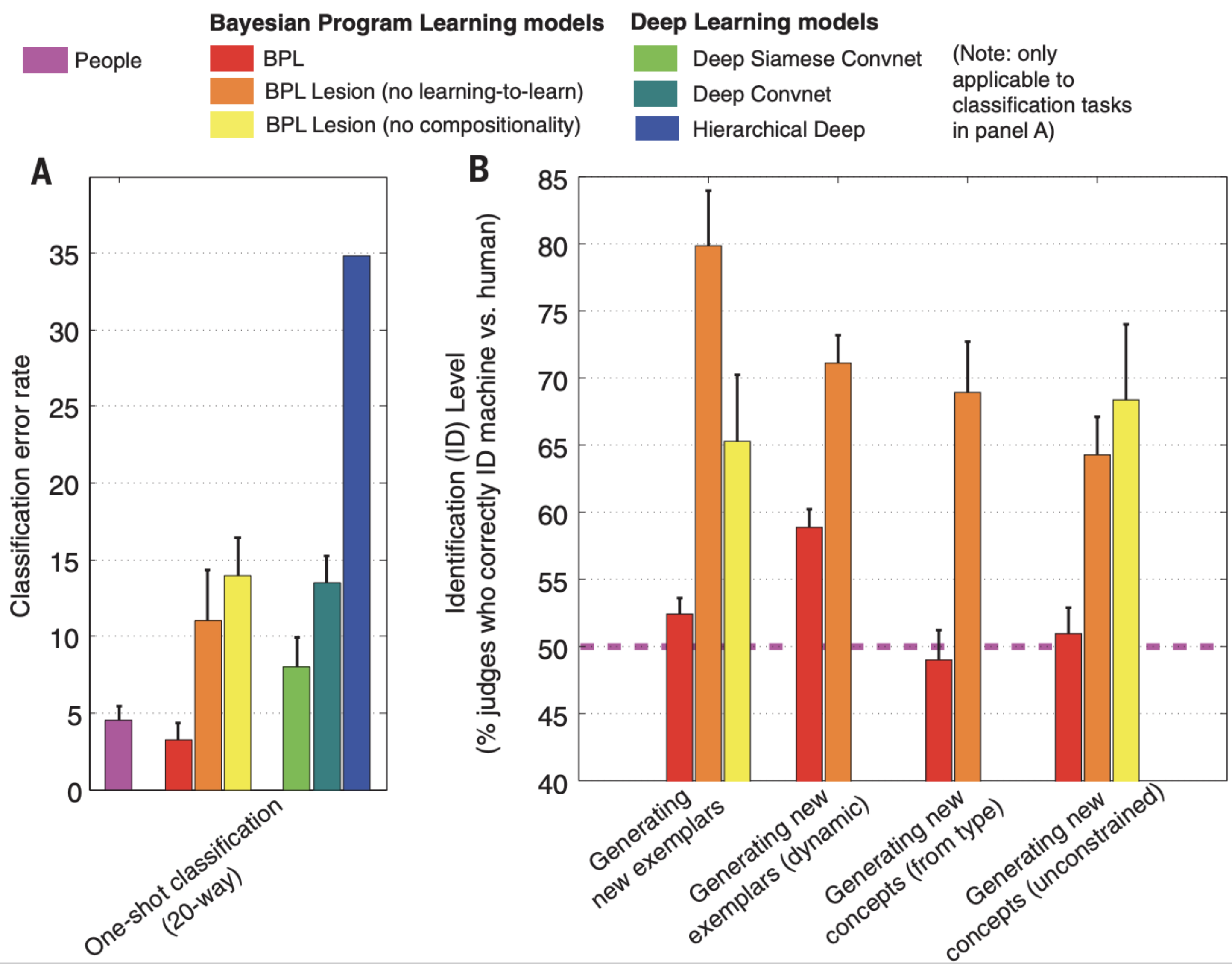


B

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

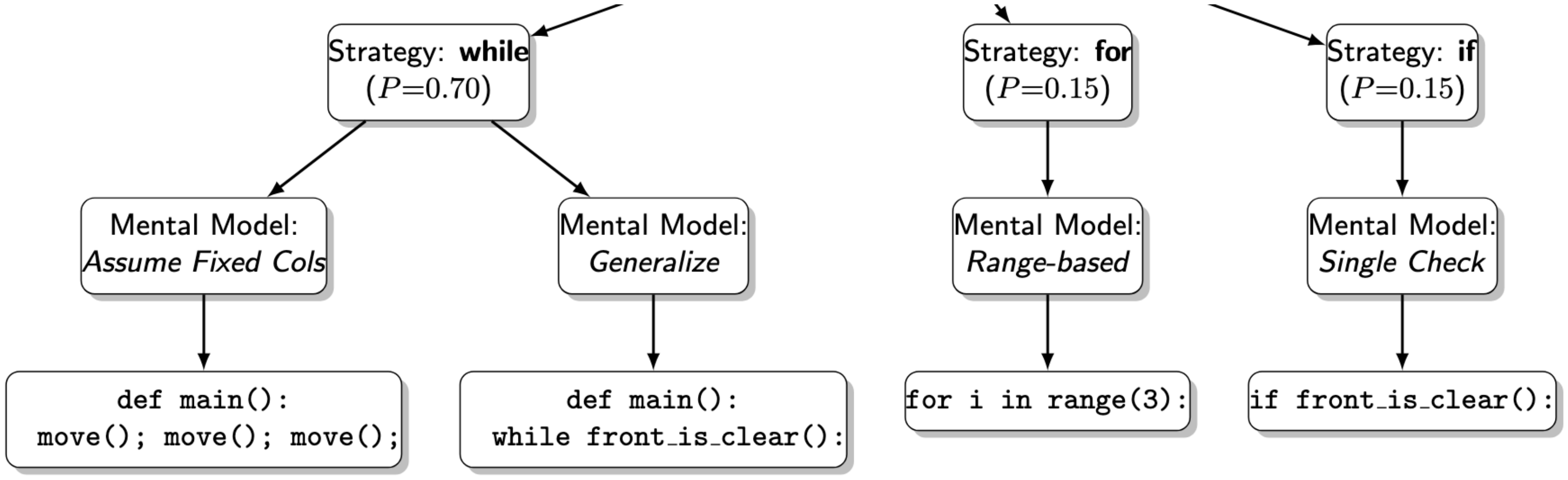
procedure GENERATETOKEN(ψ)
- **for** $i = 1 \ldots \kappa$ **do**
  - $S_i^{(m)} \leftarrow P(S_i^{(m)}|S_i)$ ▷ Add motor variance
  - $L_i^{(m)} \leftarrow P(L_i^{(m)}|R_i, T_1^{(m)}, ..., T_{i-1}^{(m)})$ ▷ Sample part's start location
  - $T_i^{(m)} \leftarrow f(L_i^{(m)}, S_i^{(m)})$ ▷ Compose a part's 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** $I^{(m)}$

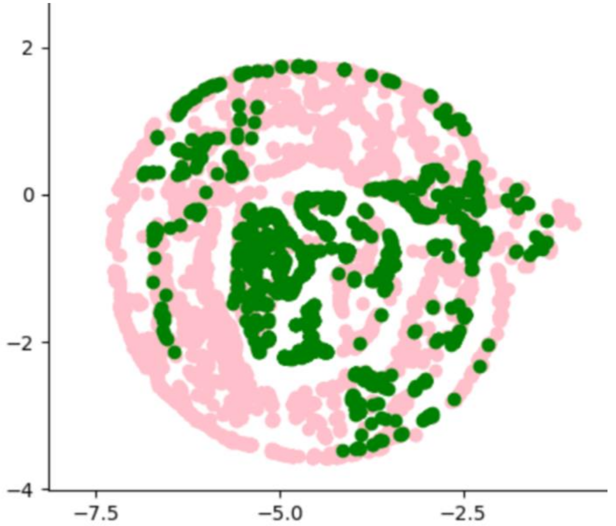
Human Level. And More!



Generative Student Modeling

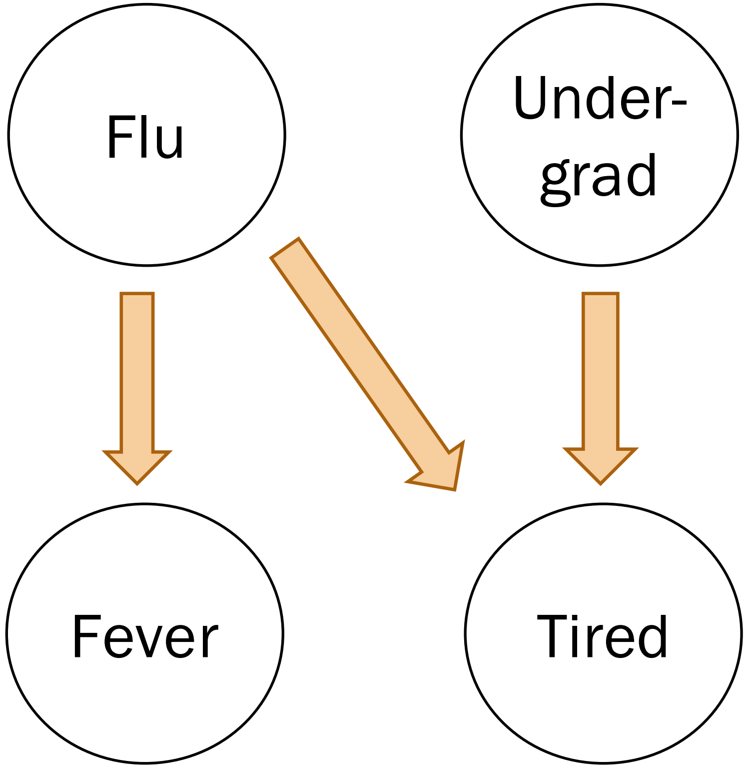
Juliette Woodrow, Chris Piech, 2021





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

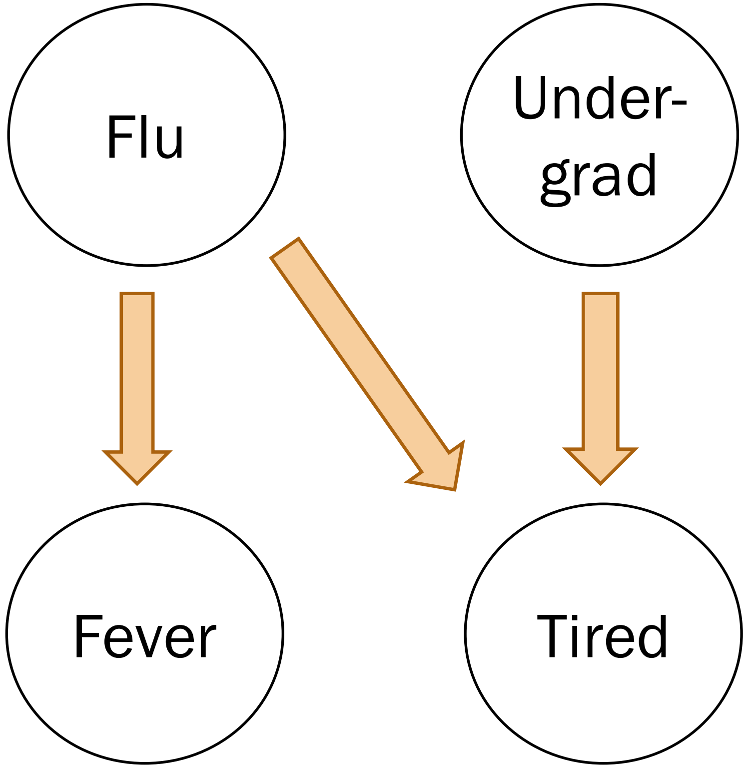
Generative Models make Independence Assumptions



What would a Stanford flu expert do?

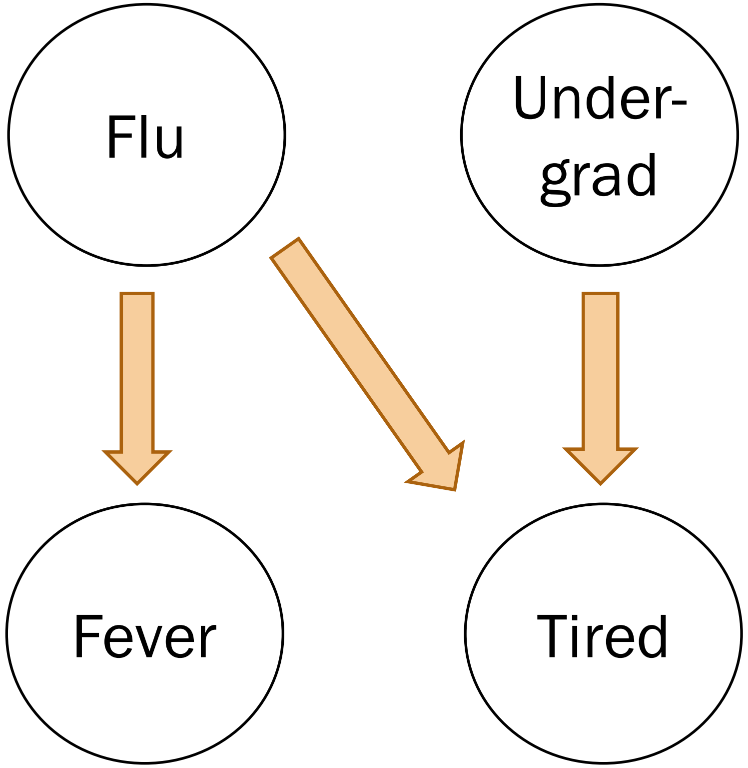
1. ☑ Describe the causality.
2. ☑ Provide $P(\text{values}|\text{causal parents})$ for each random variable
3. Implicitly assumes independences.

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

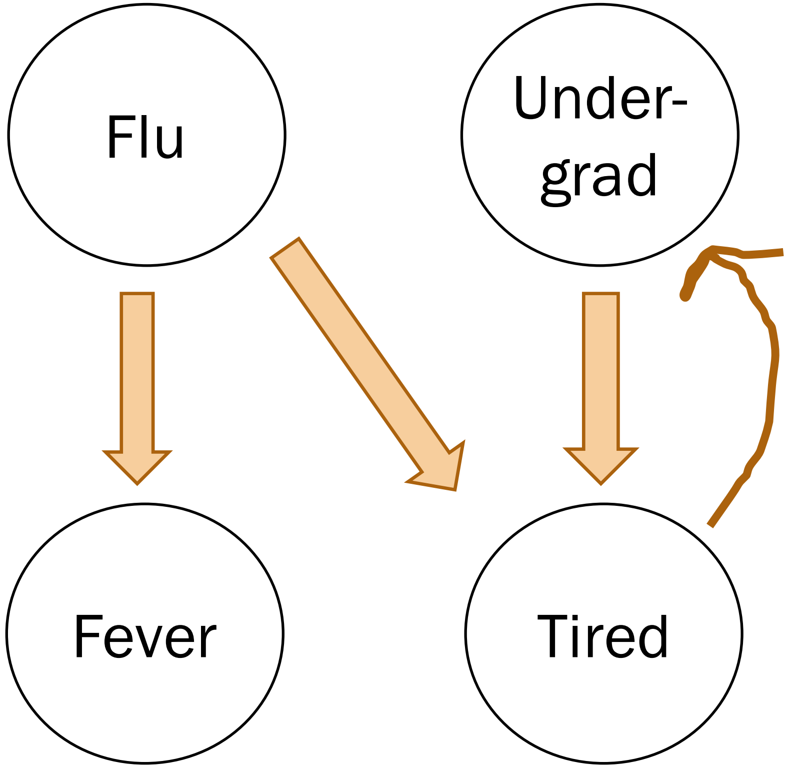


This model assumes that Flu and being an Undergraduate are independent.

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 by acyclic!

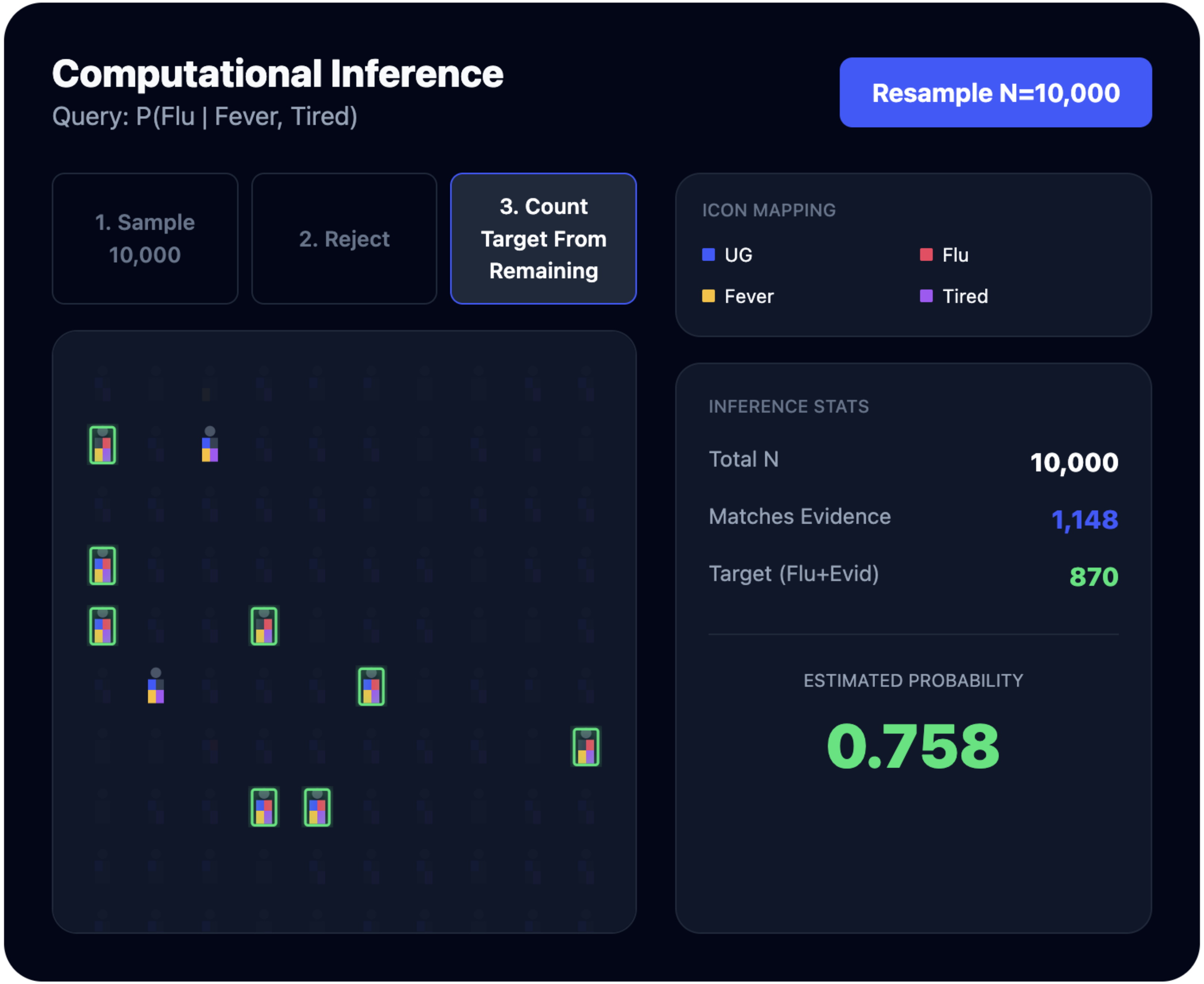
Three Guiding Questions

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(Flu | 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

```
def main():
    obs = get_observation()
    samples = sample_a_ton()
    prob = prob_flu_given_obs(samples, obs)
    print('Observation = ', obs)
    print('Pr(Flu | Obs) = ', prob)
```

Algorithm #2: Rejection Sampling

```python
def main():
    obs = get_observation()
    samples = sample_a_ton()
    prob = prob_flu_given_obs(sample
    print('Observation = ', obs)
    print('Pr(Flu | Obs) = ', prob)
```

webMd — -zsh — 56×42


```
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'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': 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': 1, 'fever': 0, 'tired': 0}
{'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': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 0, 'tired': 1}
{'flu': 0, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'flu': 0, 'undergrad': 0, 'fever': 0, 'tired': 0}
{'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': 1}
{'flu': 1, 'undergrad': 1, 'fever': 1, 'tired': 1}
{'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': 0, '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': 1, 'undergrad': 1, 'fever': 1, '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': 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}
```

Algorithm #2: Rejection Sampling

```python
def main():
    obs = get_observation()
    samples = sample_a_ton()
    prob = prob_flu_given_obs(samples, obs)
    print('Observation = ', obs)
    print('Pr(Flu | Obs) = ', prob)
```

Algorithm #2: Rejection Sampling

```python
def prob_flu_given_obs(samples, obs):
    """
    Calculate the probability of flu given many
    samples from the joint distribution and a set
    of ovservations to condition on.
    """
    # reject all samples which don't align
    # with condition
    keep_samples = []
    for sample in samples:
        if check_obs_match(sample, obs):
            keep_samples.append(sample)

    # from remaining, simply count...
    flu_count = 0
    for sample in keep_samples:
        if sample['flu'] == 1:
            flu_count += 1

    # counting can be so sweet...
    return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```python
def prob_flu_given_obs(samples, obs):
    """
    Calculate the probability of flu given many
    samples from the joint distribution and a set
    of ovservations to condition on.
    """
    # reject all samples which don't align
    # with condition
    keep_samples = []
    for sample in samples:
        if check_obs_match(sample, obs):
            keep_samples.append(sample)

    # from remaining, simply count...
    flu_count = 0
    for sample in keep_samples:
        if sample['flu'] == 1:
            flu_count += 1

    # counting can be so sweet...
    return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```python
def prob_flu_given_obs(samples, obs):
    """
    Calculate the probability of flu given many
    samples from the joint distribution and a set
    of ovservations to condition on.
    """
    # reject all samples which don't align
    # with condition
    keep_samples = []
    for sample in samples:
        if check_obs_match(sample, obs):
            keep_samples.append(sample)

    # from remaining, simply count...
    flu_count = 0
    for sample in keep_samples:
        if sample['flu'] == 1:
            flu_count += 1

    # counting can be so sweet...
    return float(flu_count) / len(keep_samples)
```

Algorithm #2: Rejection Sampling

```python
def prob_flu_given_obs(samples, obs):
    """
    Calculate the probability of flu given ma
    samples from the joint distribution and a
    of ovservations to condition on.
    """
    # reject all samples which don't align
    # with condition
    keep_samples = []
    for sample in samples:
        if check_obs_match(sample, obs):
            keep_samples.append(sample)

    # from remaining, simply count...
    flu_count = 0
    for sample in keep_samples:
        if sample['flu'] == 1:
            flu_count += 1

    # counting can be so sweet...
    return float(flu_count) / len(keep_samples)
```

webMd — -zsh — 53×25

```
{'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:

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

n = # of total trials

$n(E)$ = # trials where E occurs

webMd — -zsh — 53×25

```
{'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) =  0.12286465177398l6
piech@Chriss-MBP-5 webMd %
```

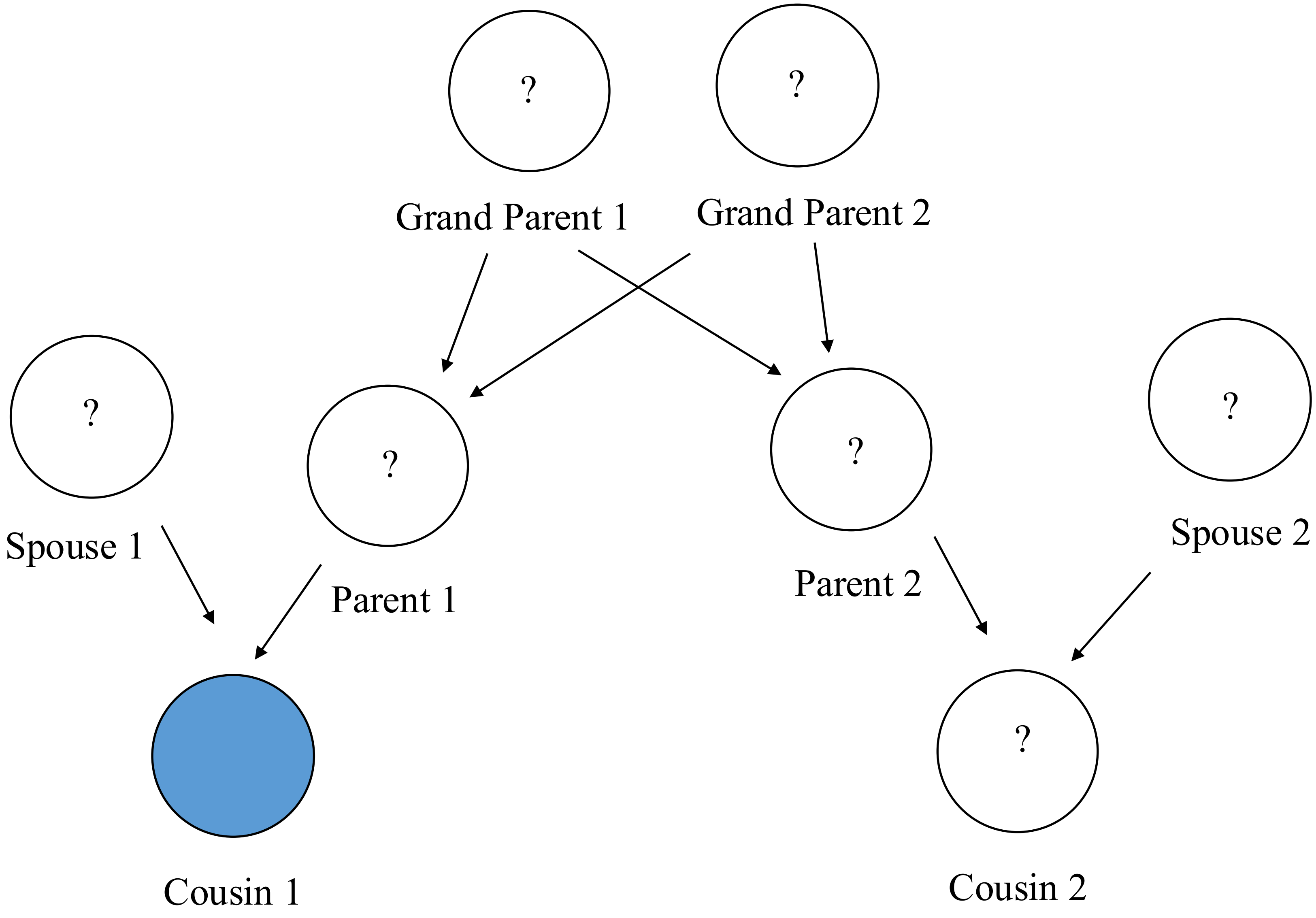
If you can sample enough from the joint distribution, you can answer any probability question

Each one of these is one joint sample

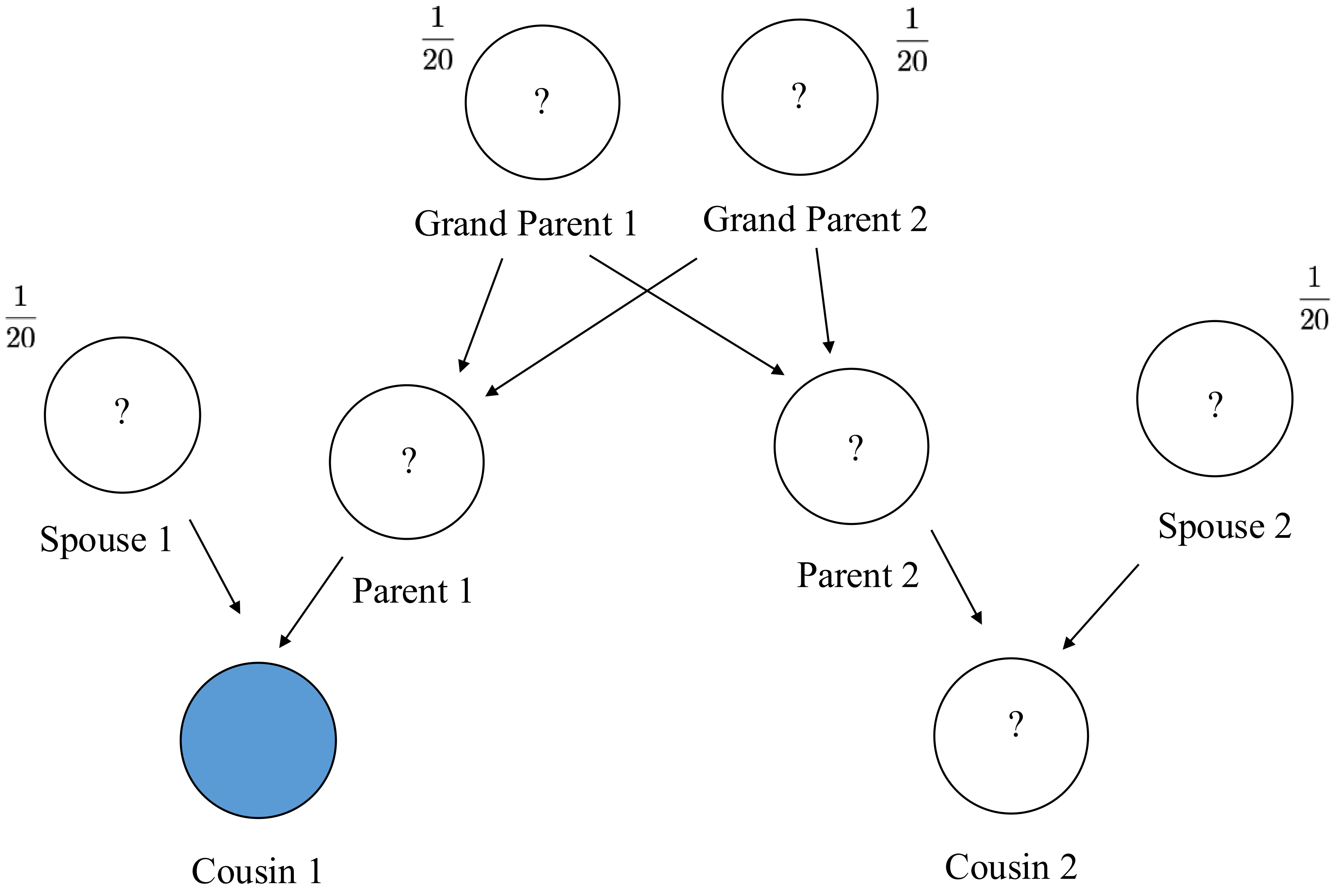
Lets try another question

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.

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

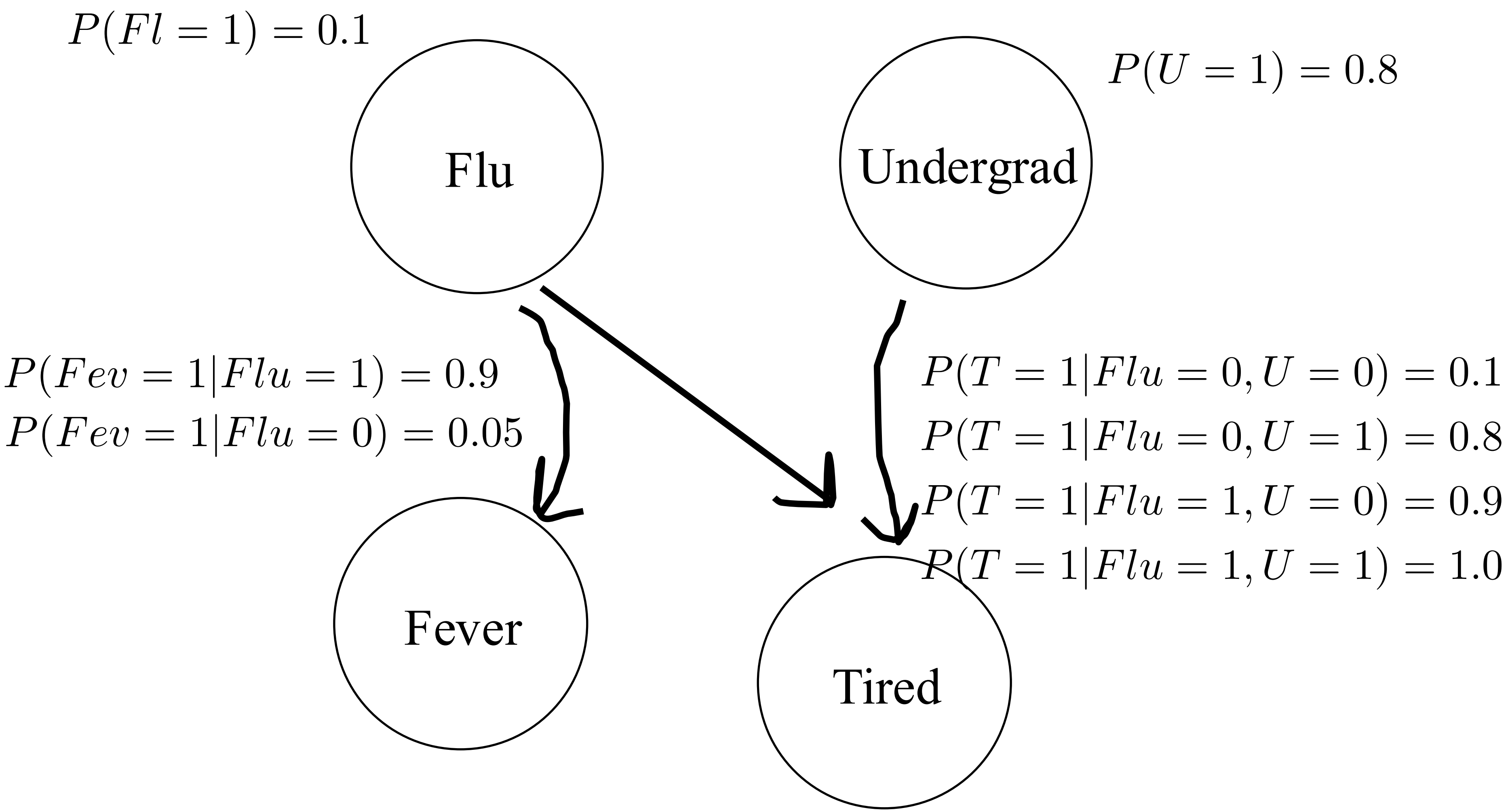


You observe that someone has a **recessive** gene.
What is the probability that their **cousin** has the same recessive gene?

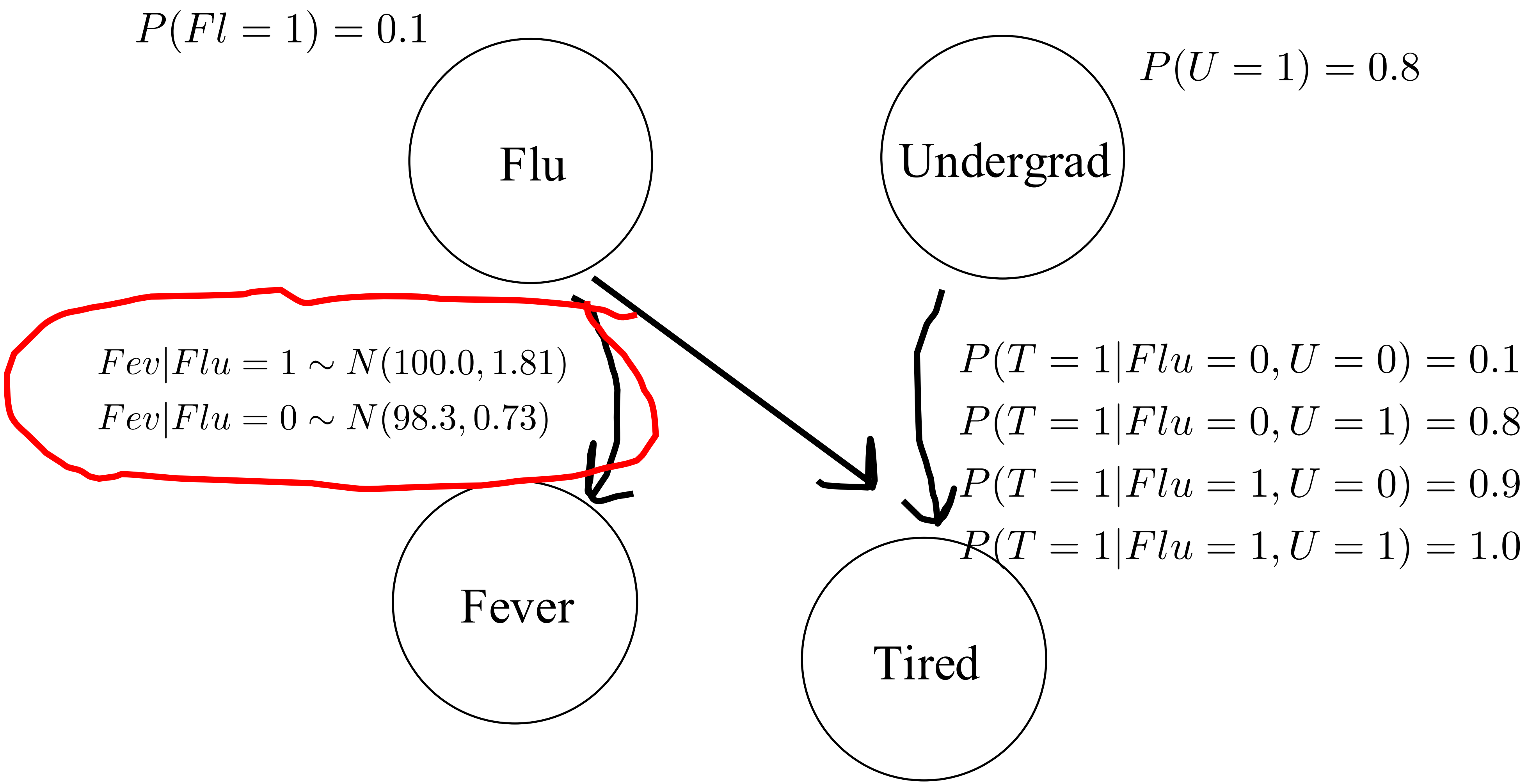


What’s the matter with
rejection sampling?

Probabilistic Model



Probabilistic Model



Back to the code !!

Many Algorithms

Markov Chain

MCMC

Monte Carlo

Many Algorithms

```
[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, 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]

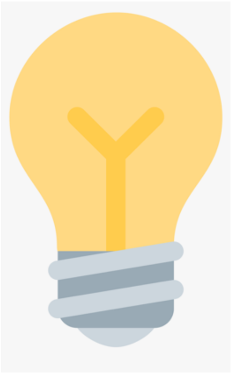
Many Algorithms

Rejection Sampling

MCMC

Pyro

Idea2Text



Three Guiding Questions

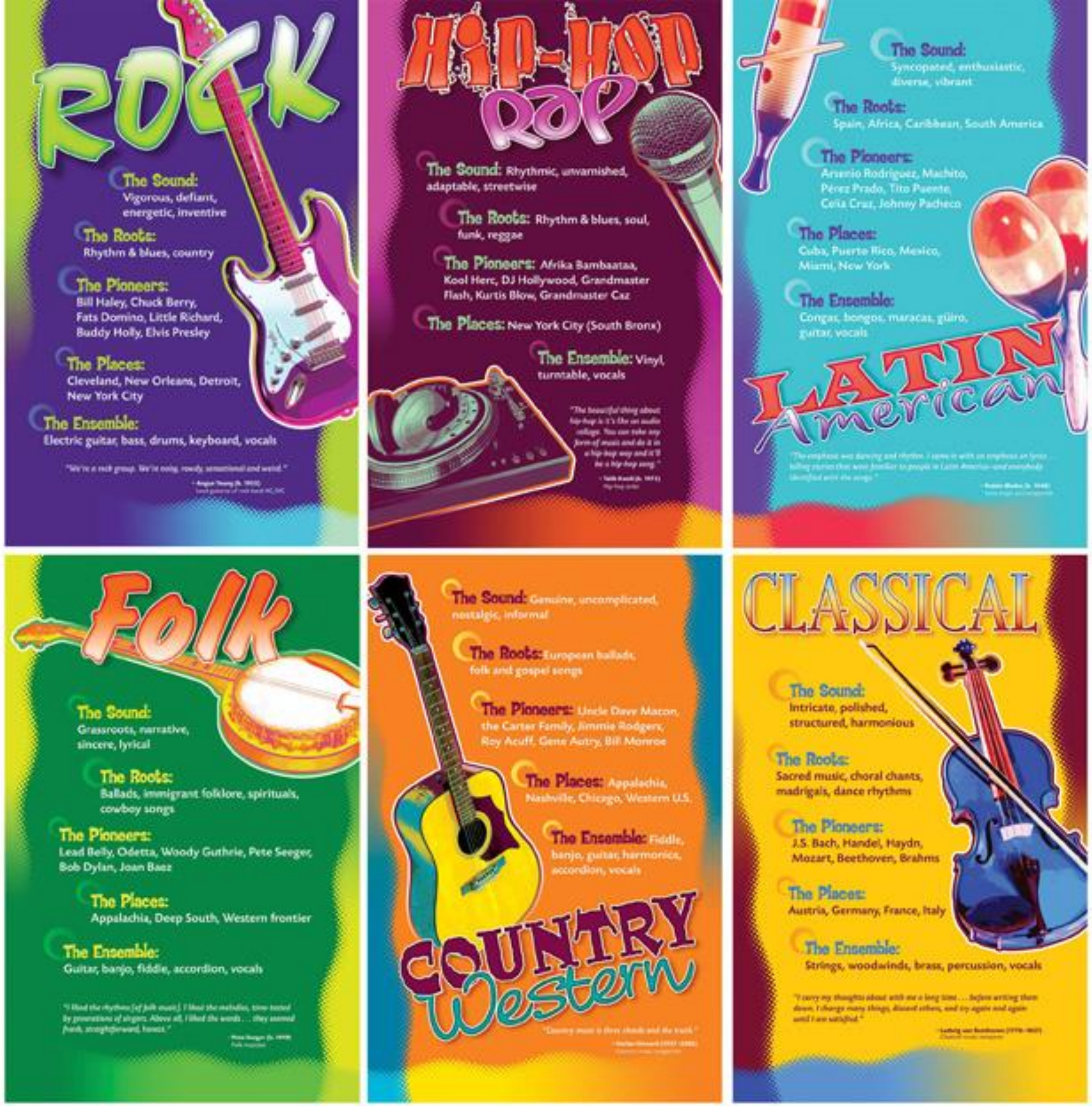
1. How do people define large models?
2. How can we do inference in large models?
3. What data can inform the design process?

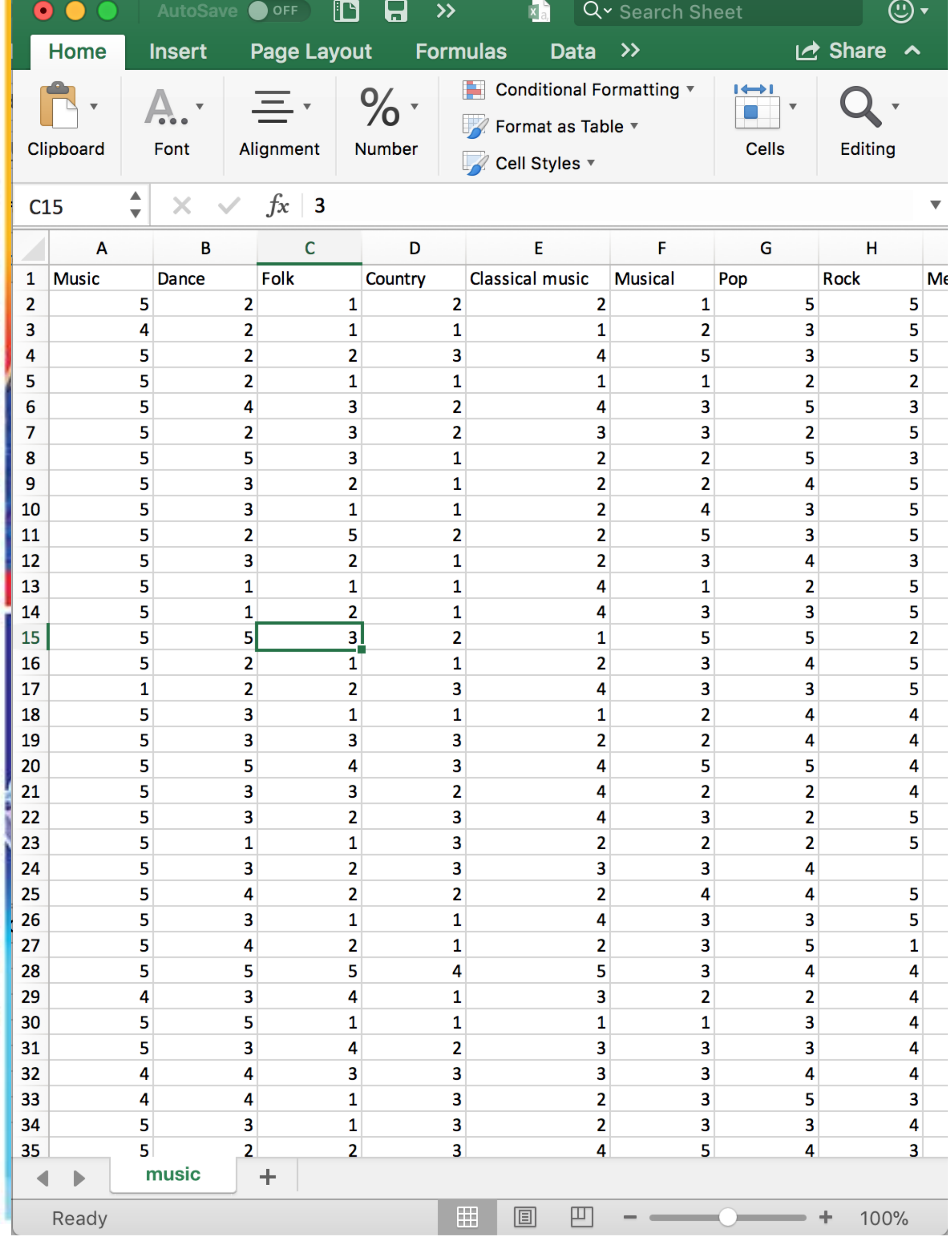
Three Guiding Questions

1. How do people define large models?

2. How can we do inference in large models?

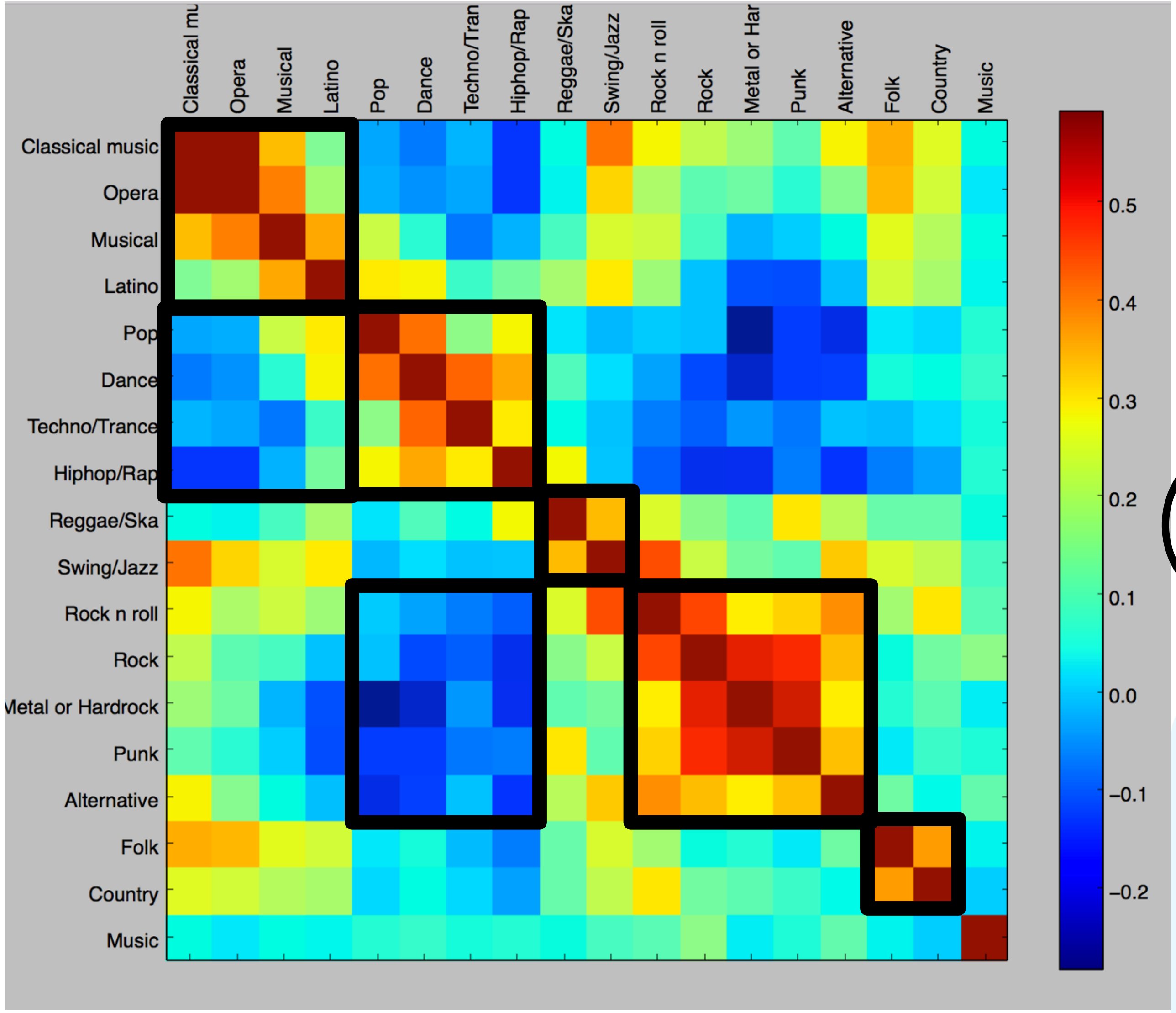
3. What data can inform the design process?

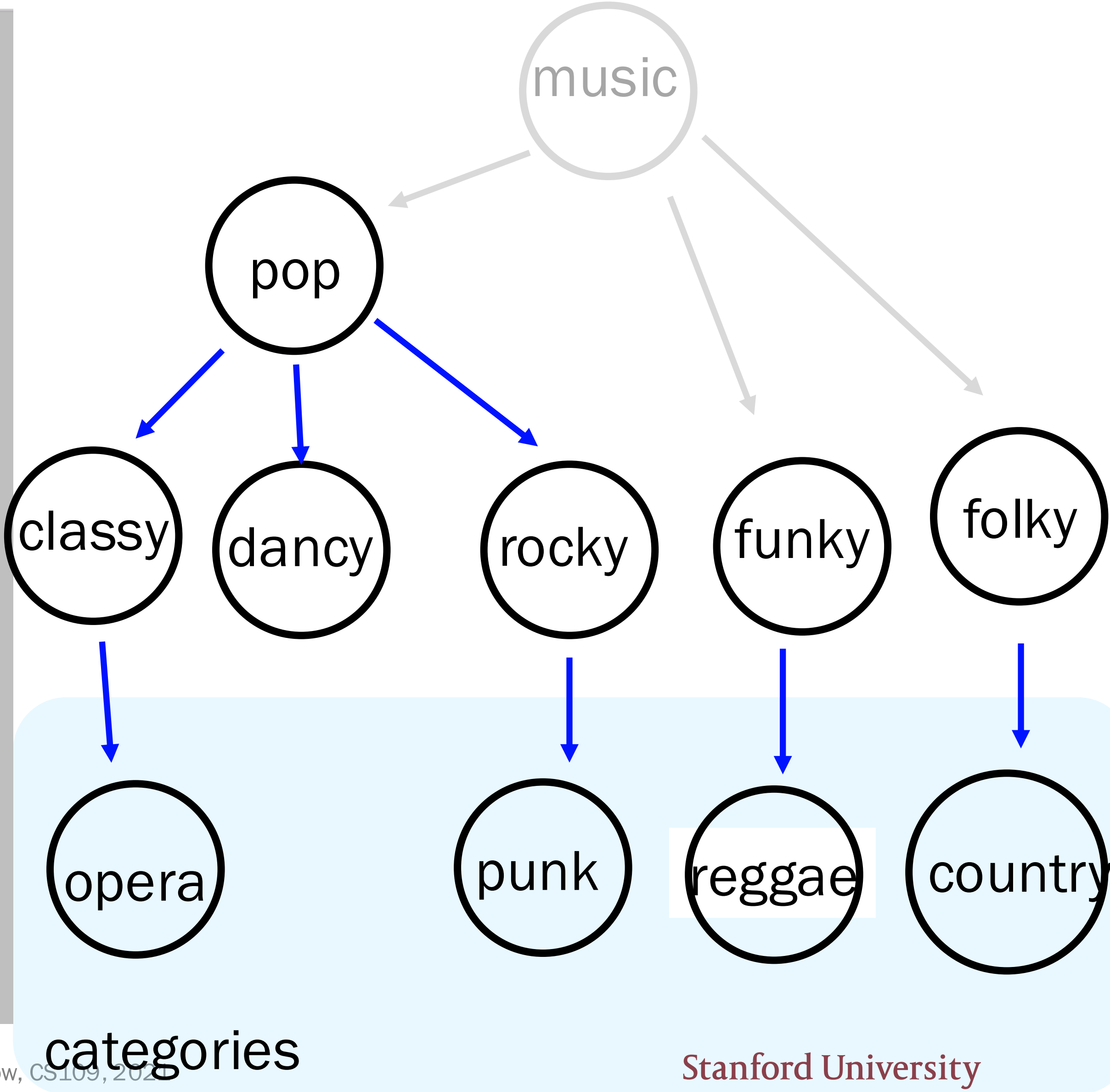




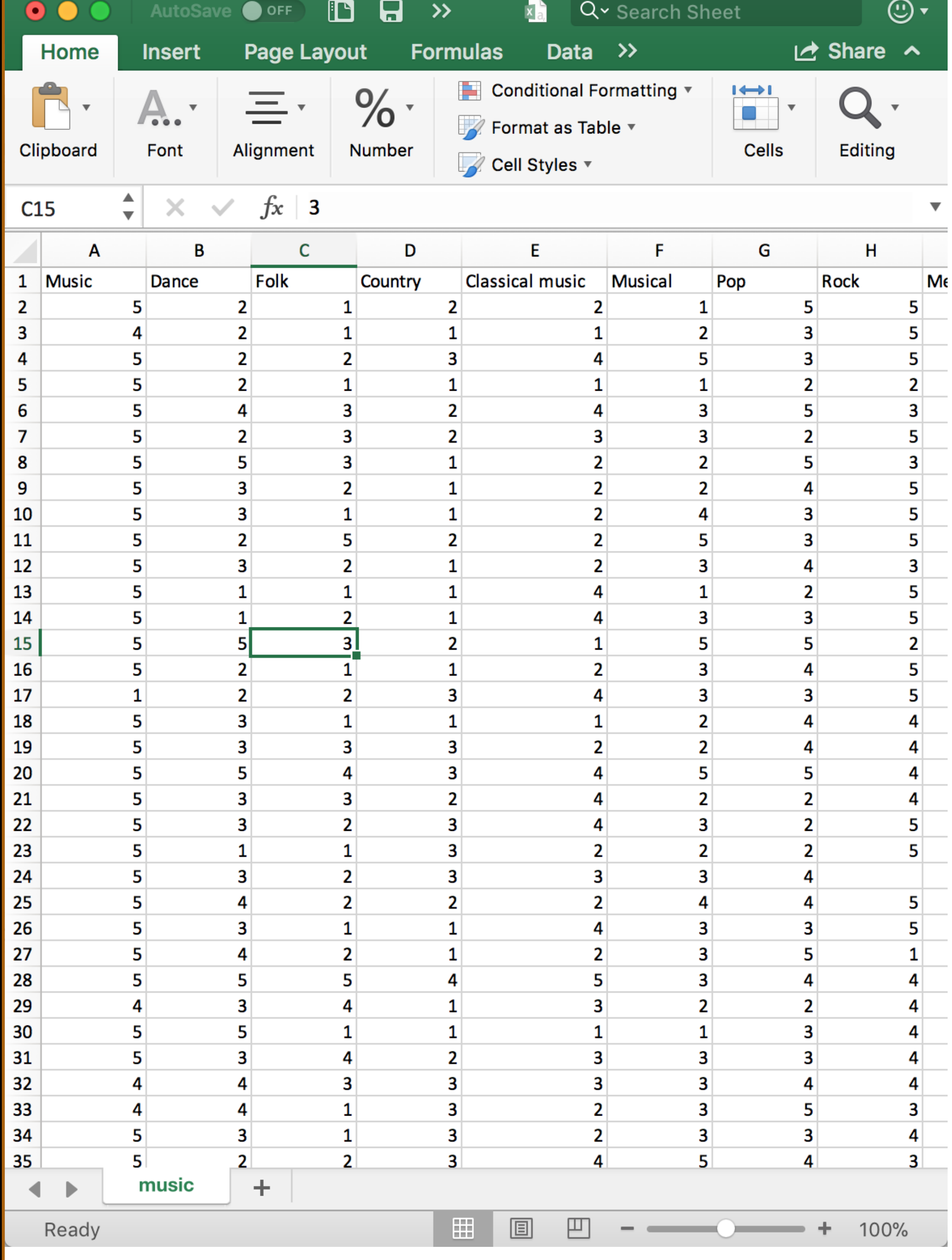
	A	B	C	D	E	F	G	H
1	Music	Dance	Folk	Country	Classical music	Musical	Pop	Rock
2	5	2	1	2	2	1	5	5
3	4	2	1	1	1	2	3	5
4	5	2	2	3	4	5	3	5
5	5	2	1	1	1	1	2	2
6	5	4	3	2	4	3	5	3
7	5	2	3	2	3	3	2	5
8	5	5	3	1	2	2	5	3
9	5	3	2	1	2	2	4	5
10	5	3	1	1	2	4	3	5
11	5	2	5	2	2	5	3	5
12	5	3	2	1	2	3	4	3
13	5	1	1	1	4	1	2	5
14	5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

From Correlation to Bayes Net!





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



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

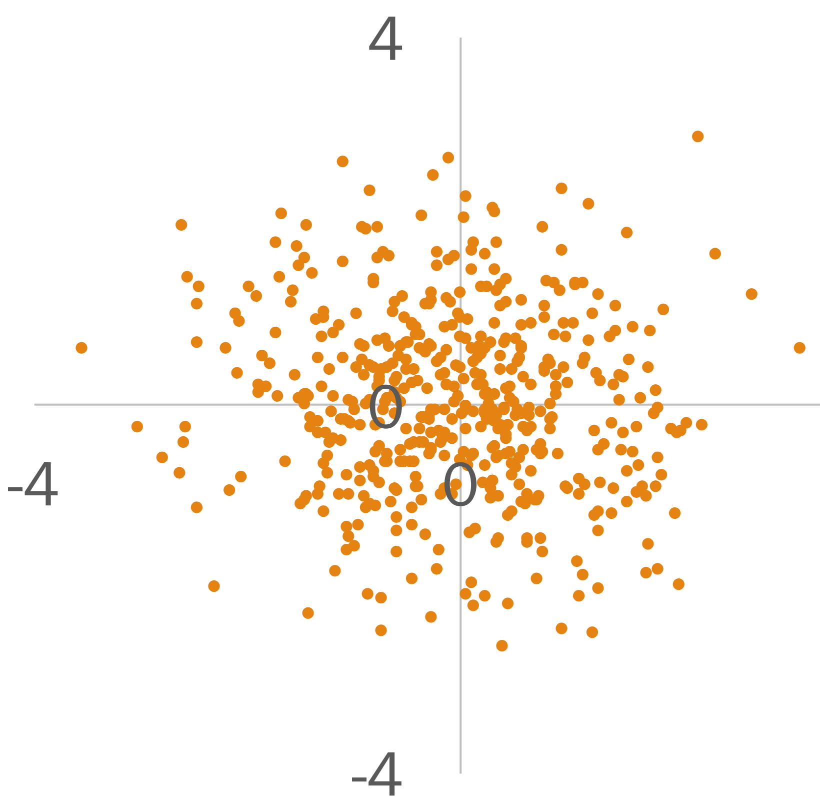
Bat Data

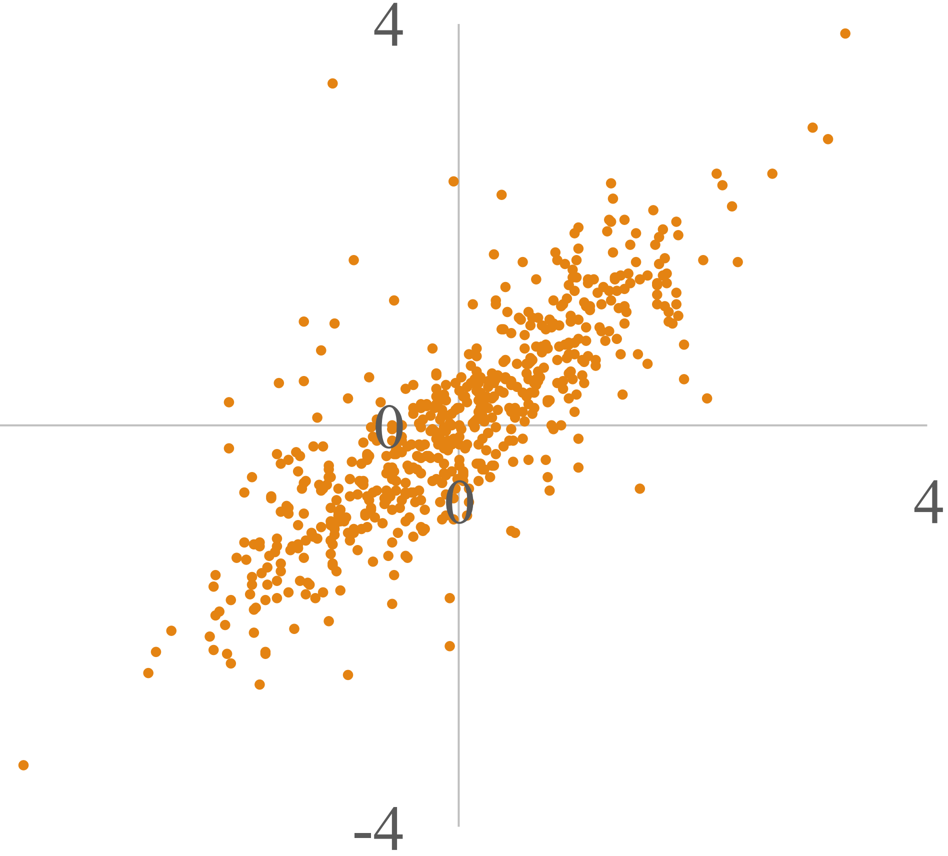
Gene1	Gene2	Gene3	Gene4	Gene5	Trait
TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	TRUE	FALSE	FALSE	FALSE
TRUE	FALSE	TRUE	TRUE	TRUE	FALSE
FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	FALSE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

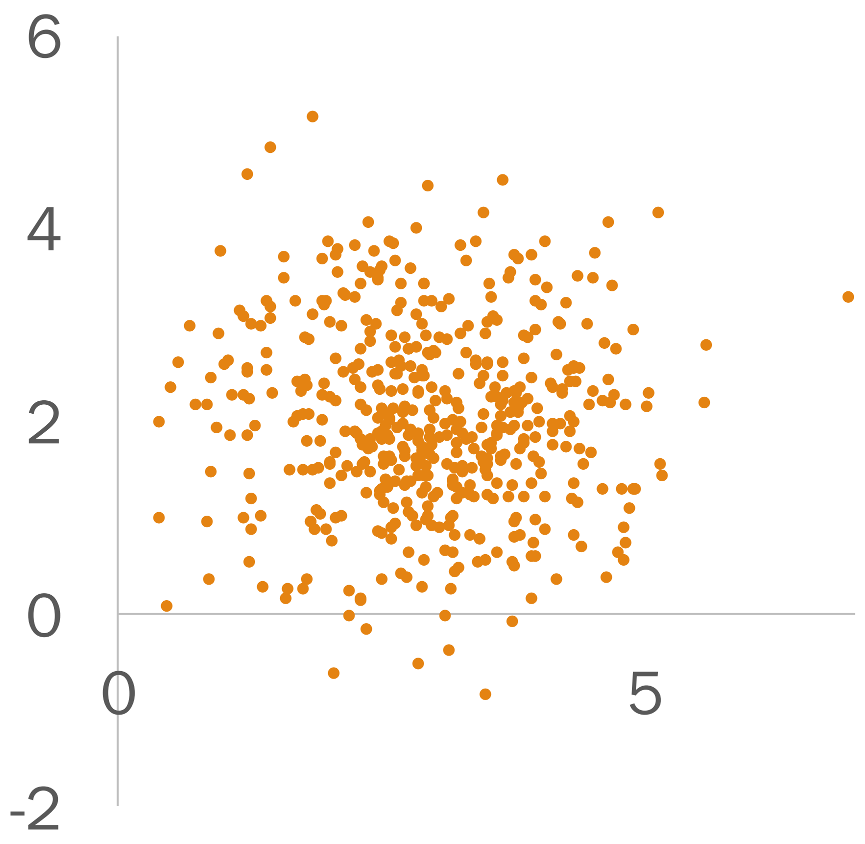
Gene5	Trait
0.76	0.83
0.94	0.85
0.82	0.03
0.94	0.32
0.50	0.10
0.40	0.53
0.90	0.67
0.29	0.71
0.72	0.25
0.15	0.24
0.79	0.98
0.68	0.77
0.71	0.37
0.36	0.18
0.62	0.08
0.59	0.38
0.82	0.76

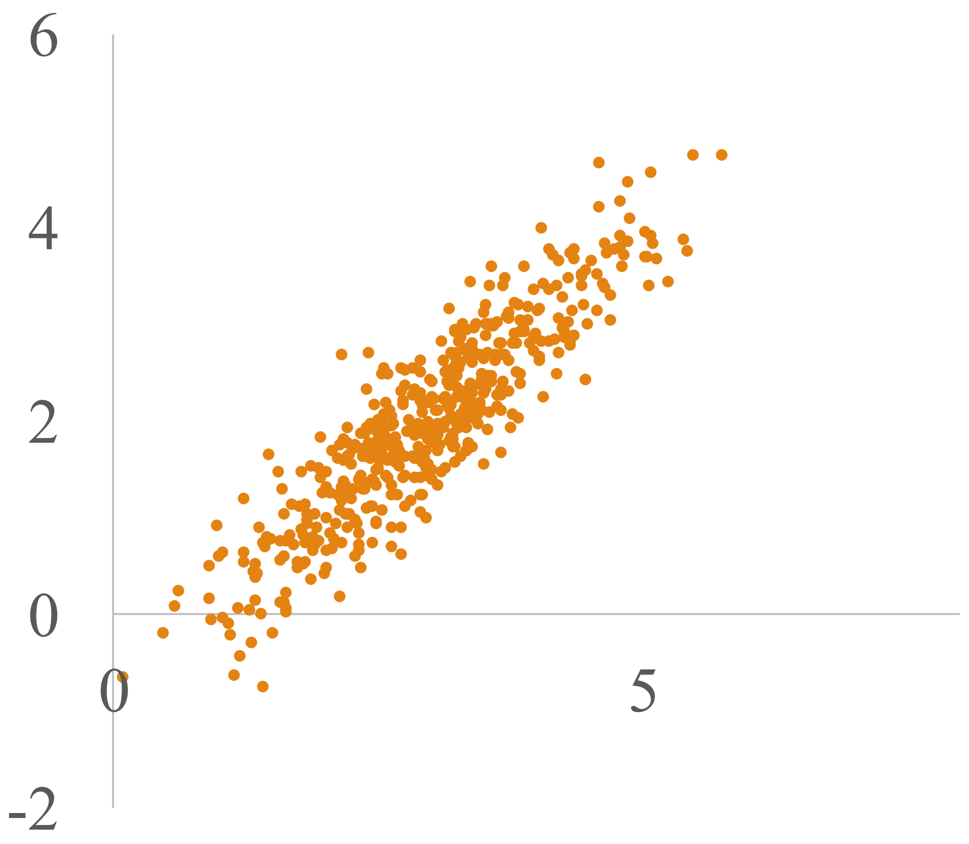
Spot The Difference



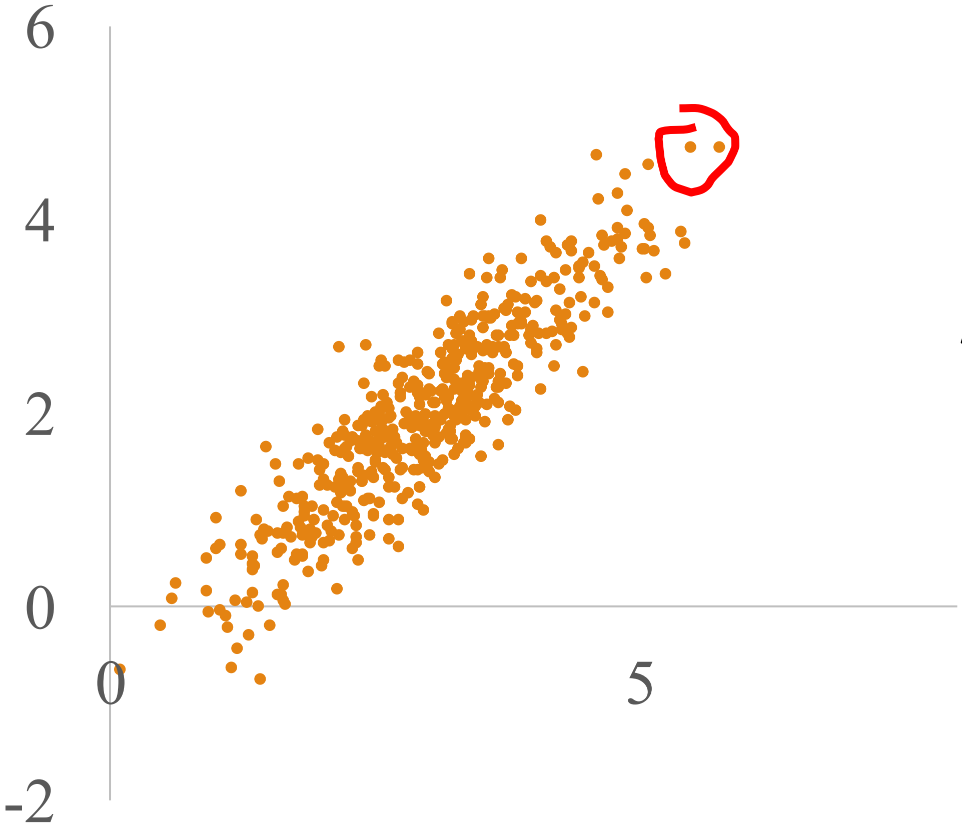


Spot The Difference





Vary Together

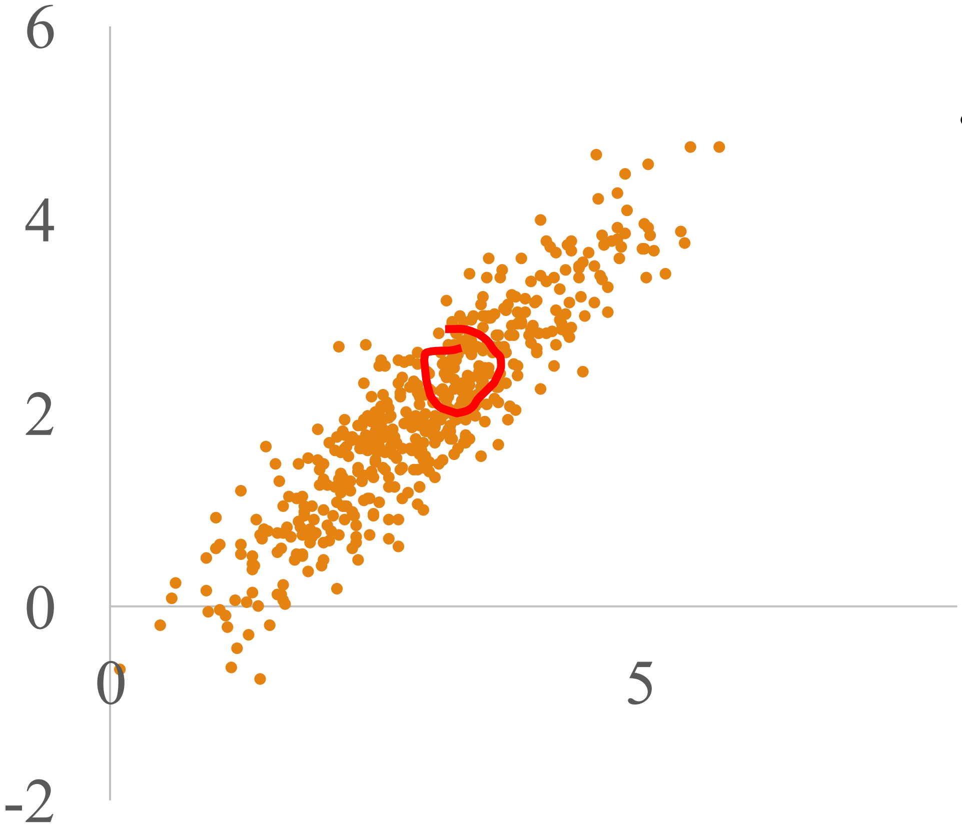


$$x - E[x] = 3$$

$$y - E[y] = 2.6$$

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

Vary Together

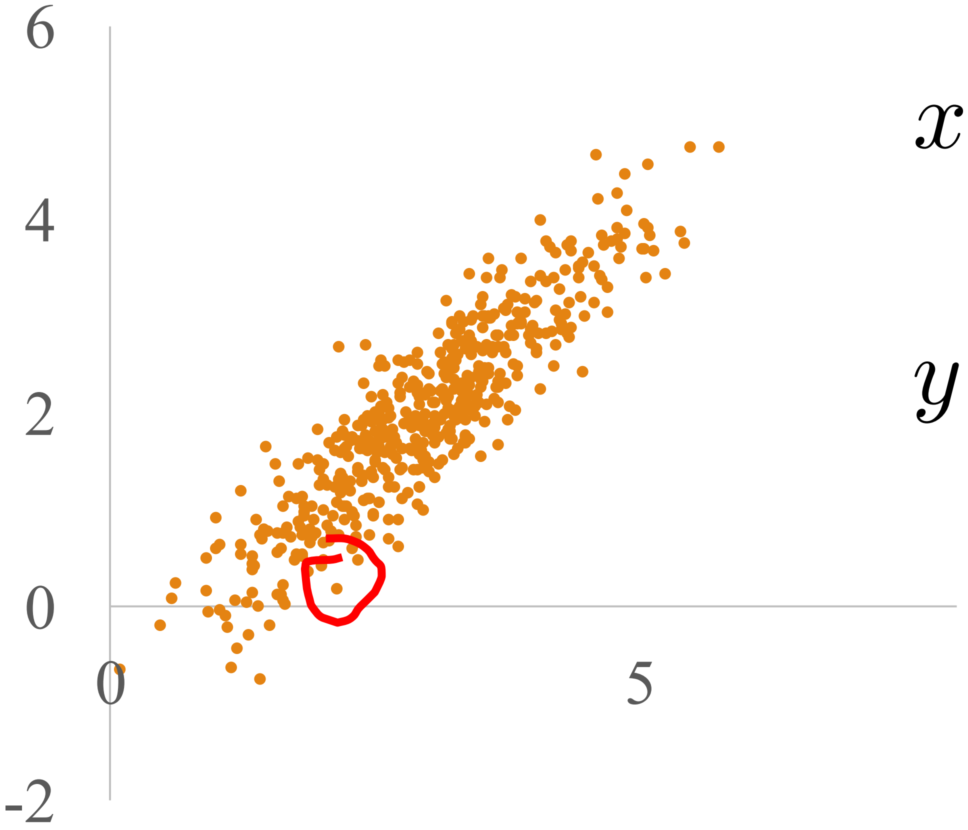


$$x - E[x] \approx 0$$

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

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

Vary Together

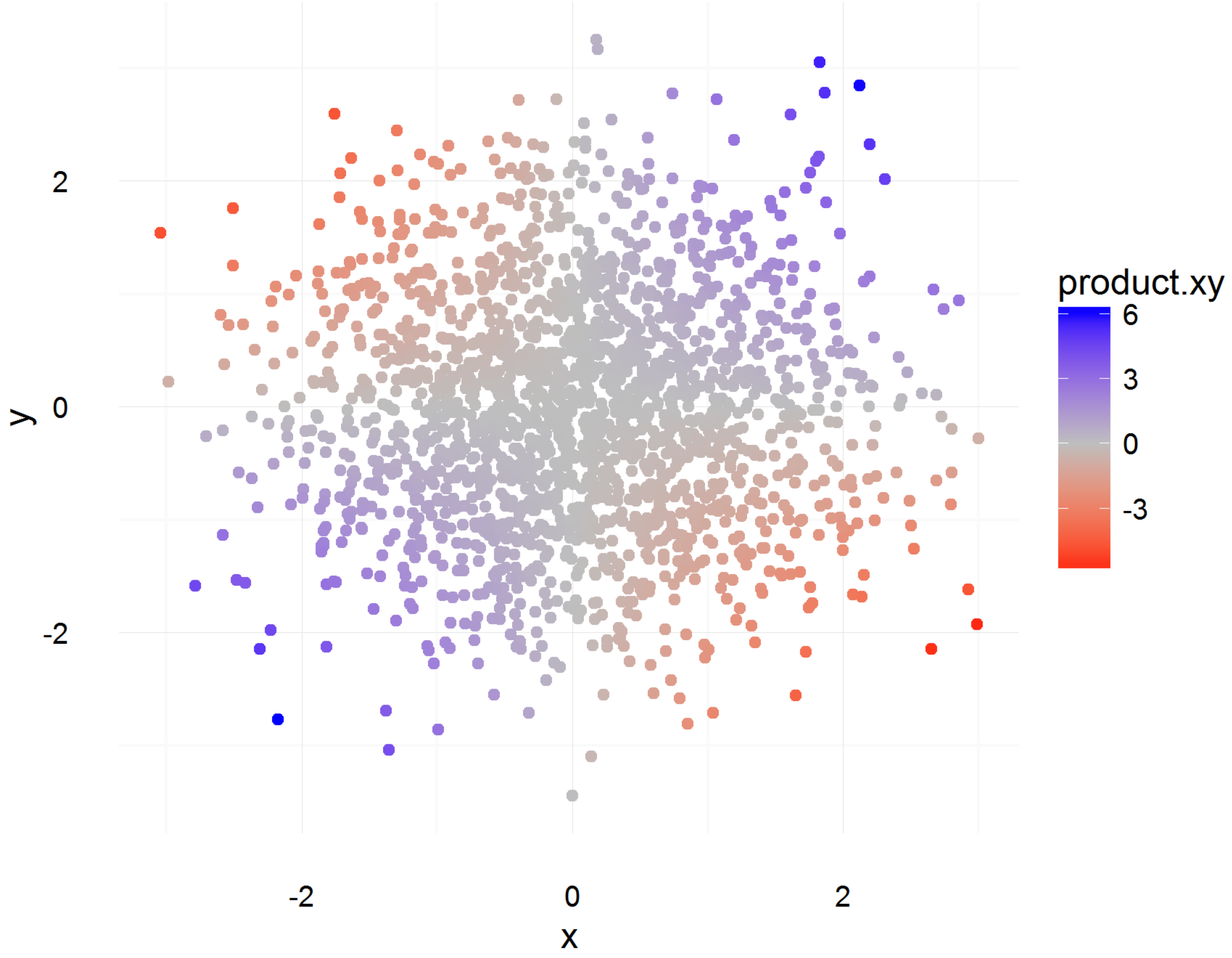


$$x - E[x] = -1.1$$

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

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

Understanding Covariance



The Dance of the Covariance

Say X and Y are arbitrary random variables

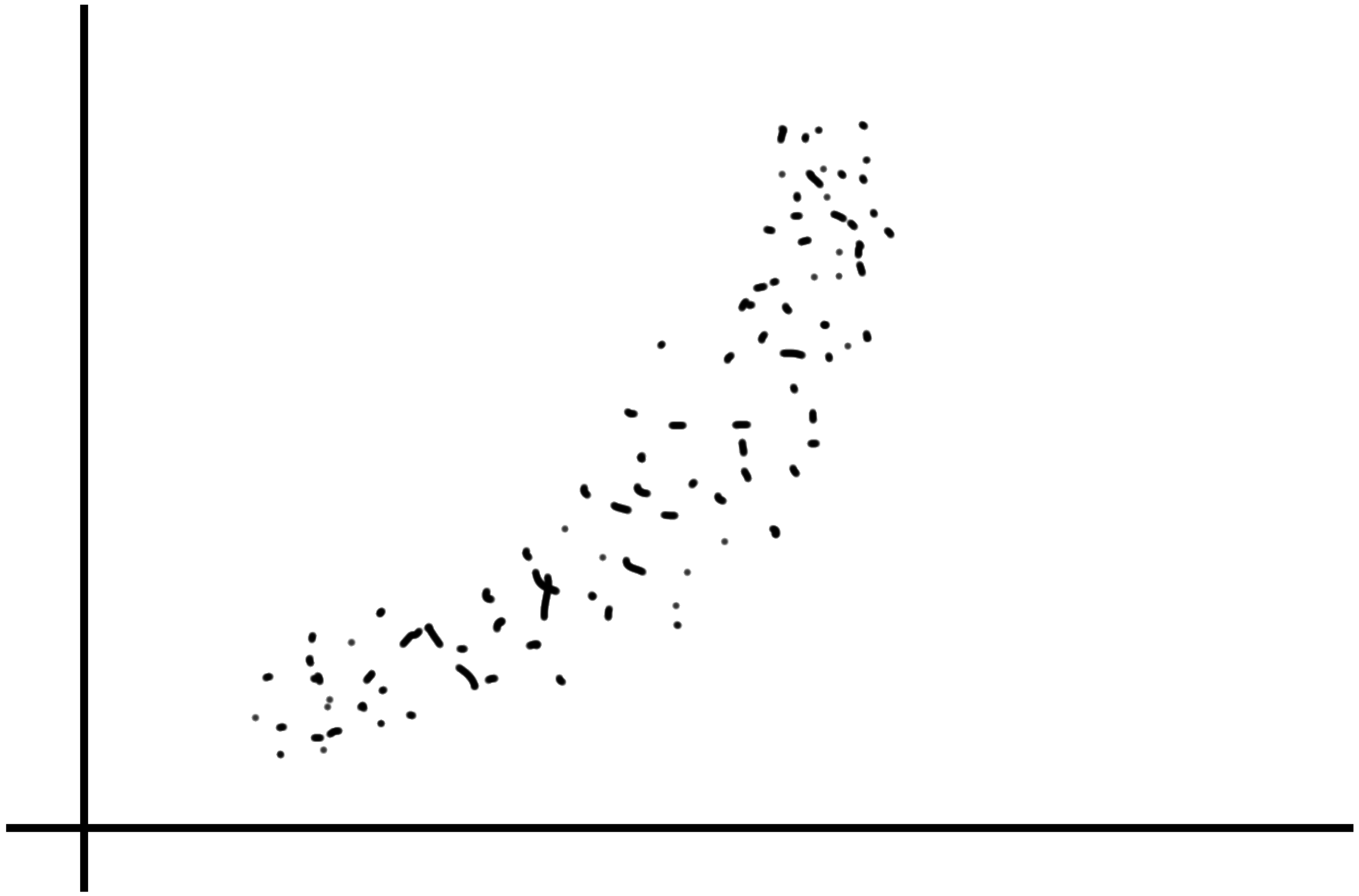
Covariance of X and Y:

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

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

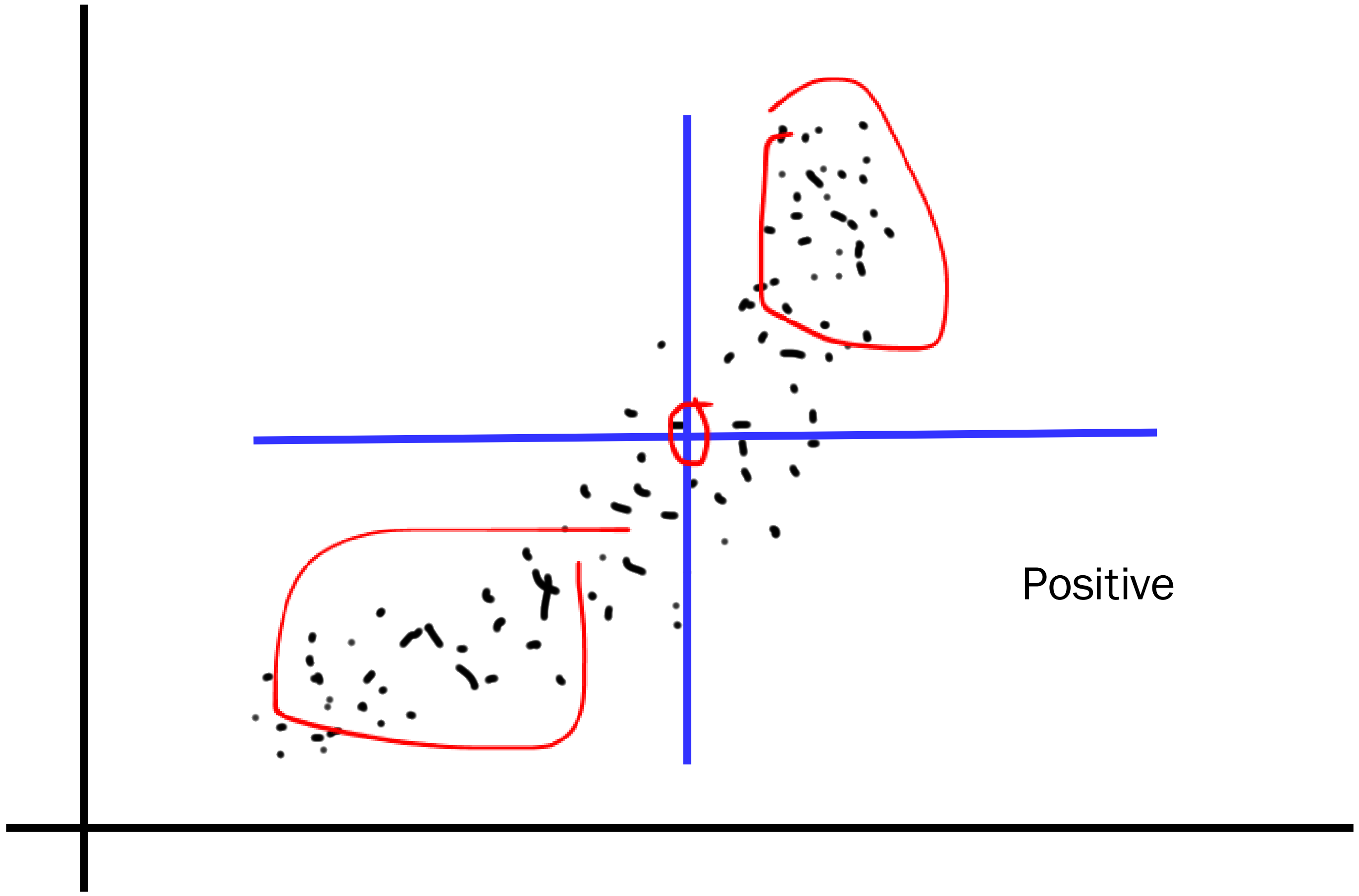
Covariance

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



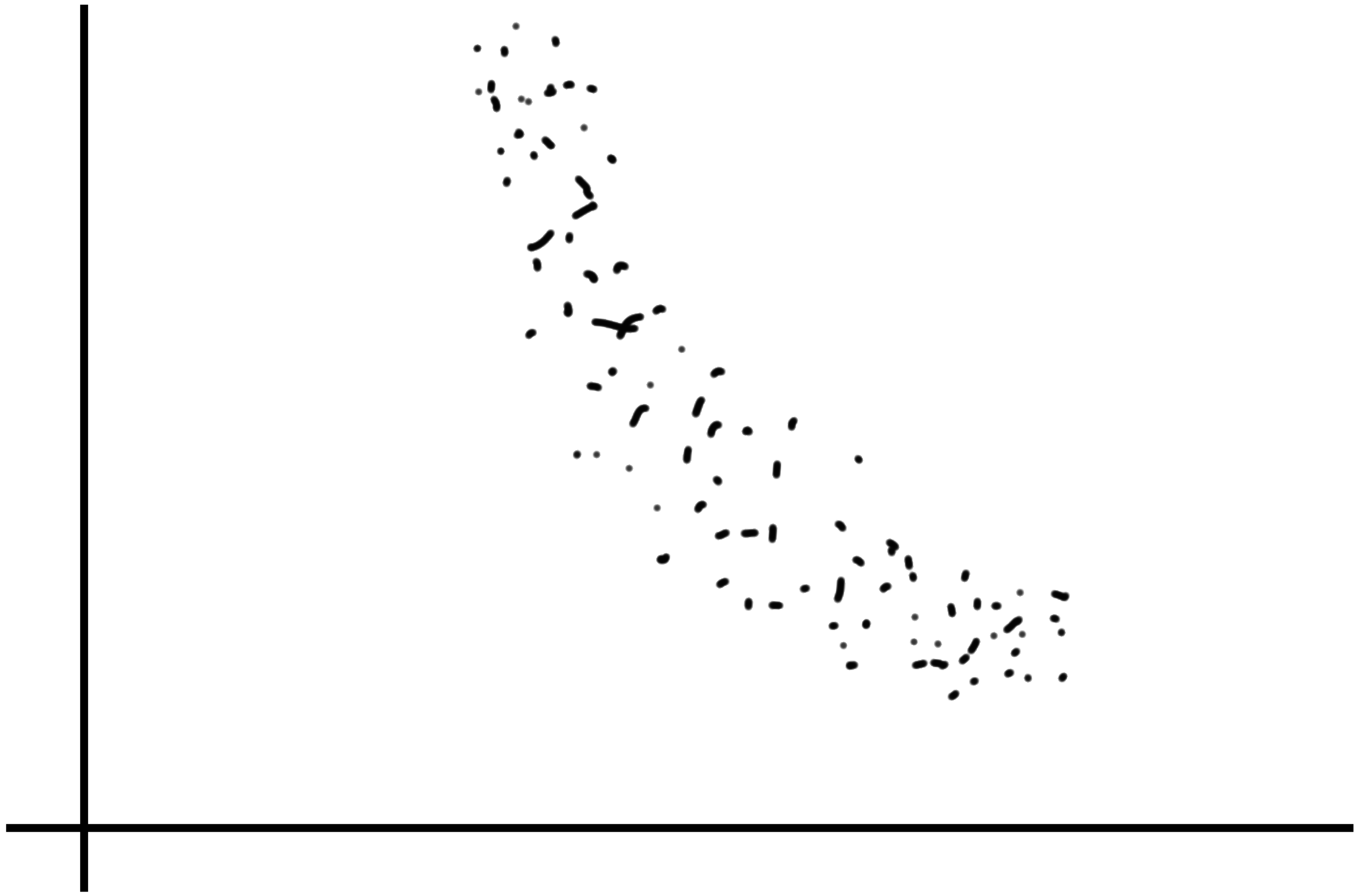
Covariance

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



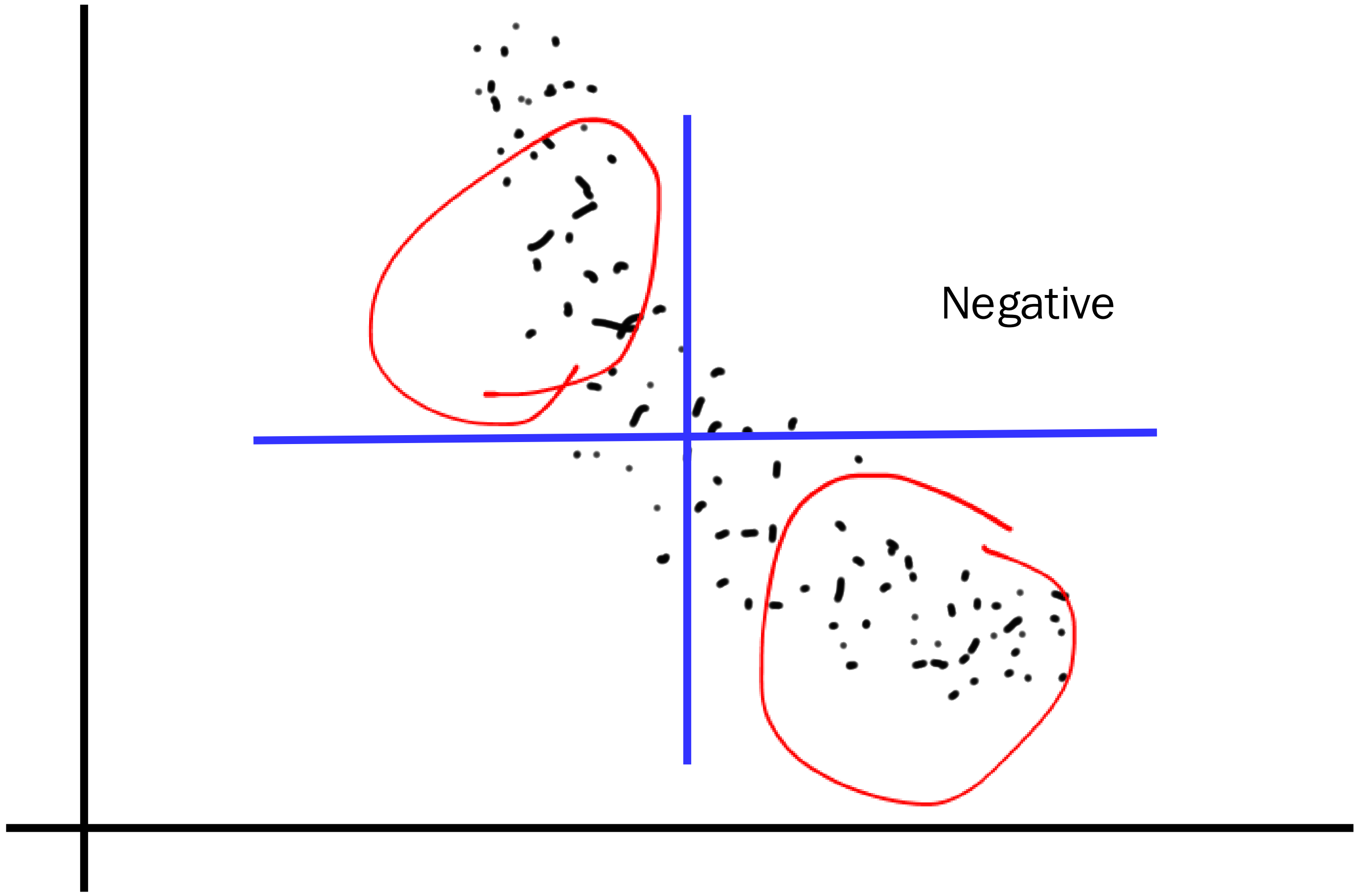
Covariance

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



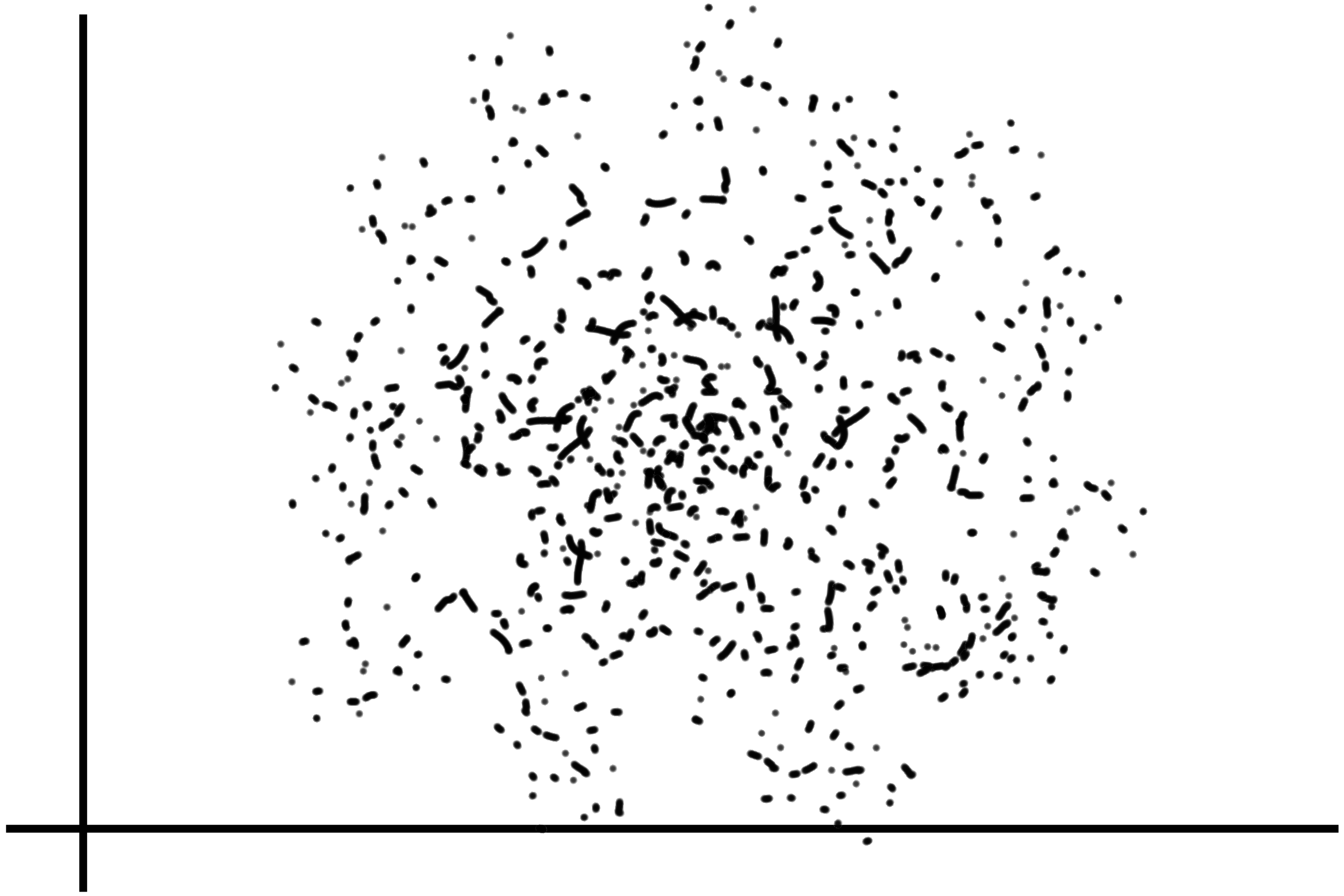
Covariance

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



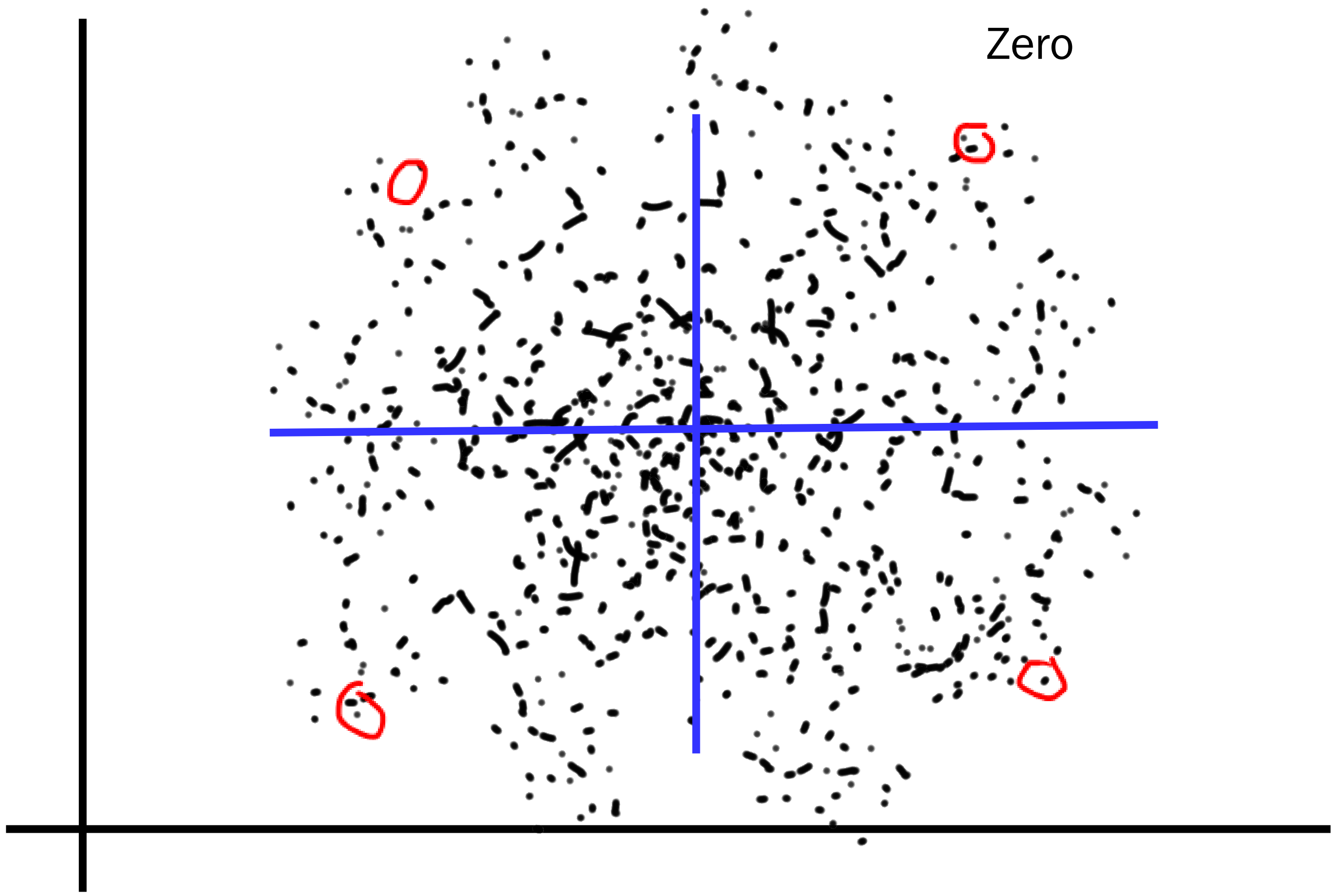
Covariance

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:

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

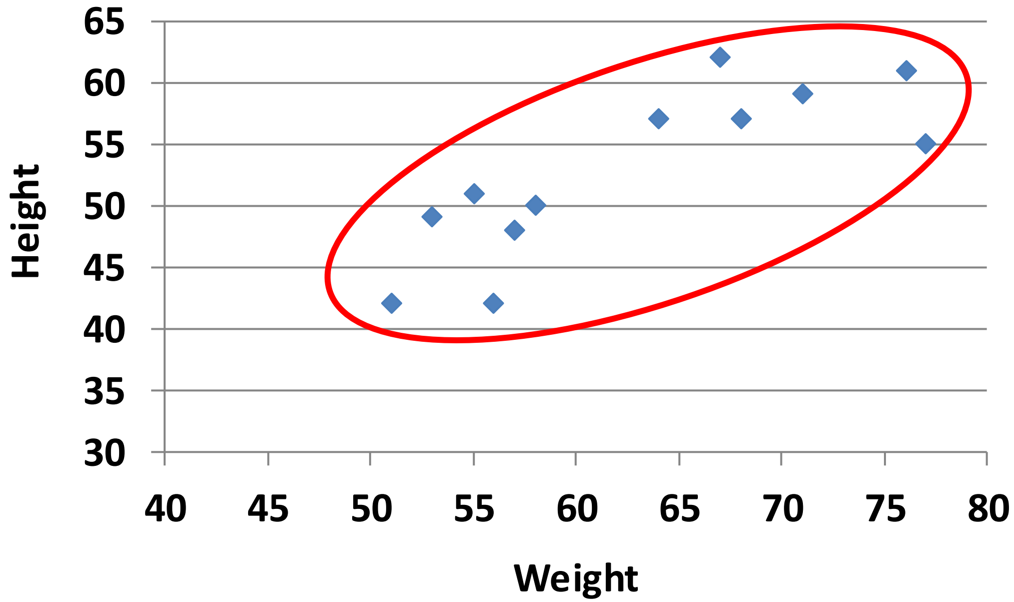
- X and Y independent → Cov(X,Y) = 0
- But Cov(X,Y) = 0 does **not** imply X and Y independent!

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



Cov(W, H) = E[W*H] – E[W]E[H]
= 3355.83 – (62.75)(52.75)
= 45.77

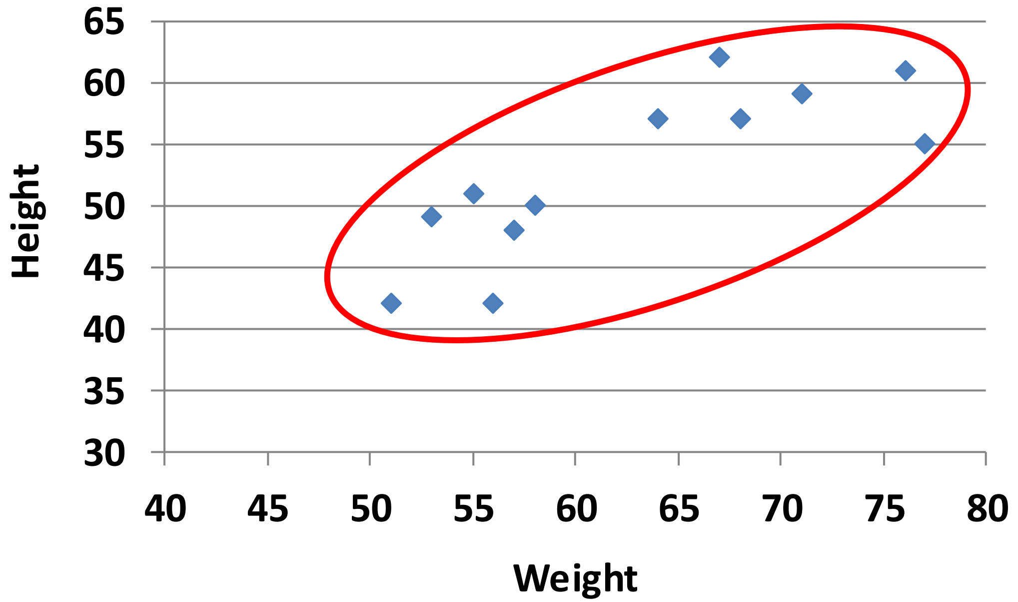
Correlation

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

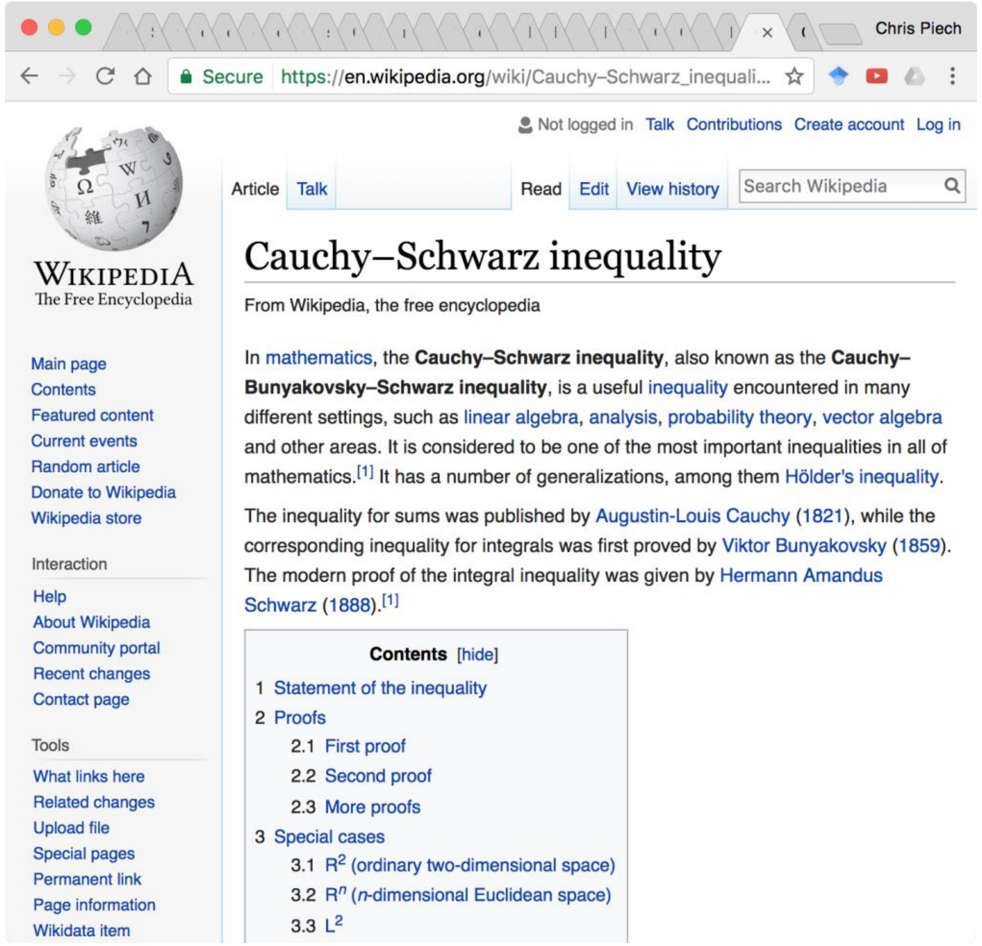


Cov(W, H) = E[W*H] – E[W]E[H]

= 3355.83 – (62.75)(52.75)

= 45.77

Cauchy Schwarz, a great way to normalize!



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

Viva La Correlatión

Say X and Y are arbitrary random variables

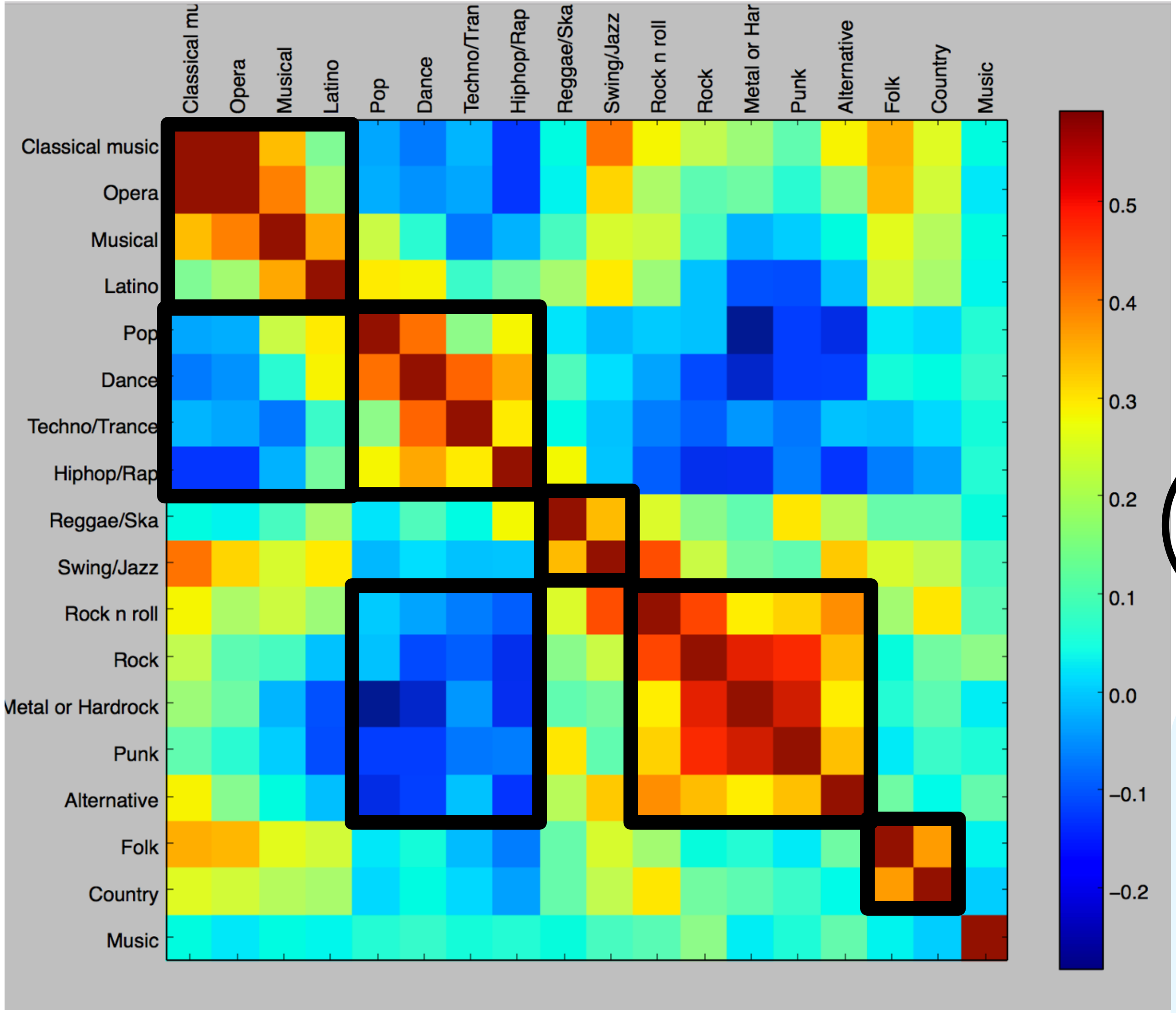
- Correlation of X and Y, denoted $\rho(X, Y)$:

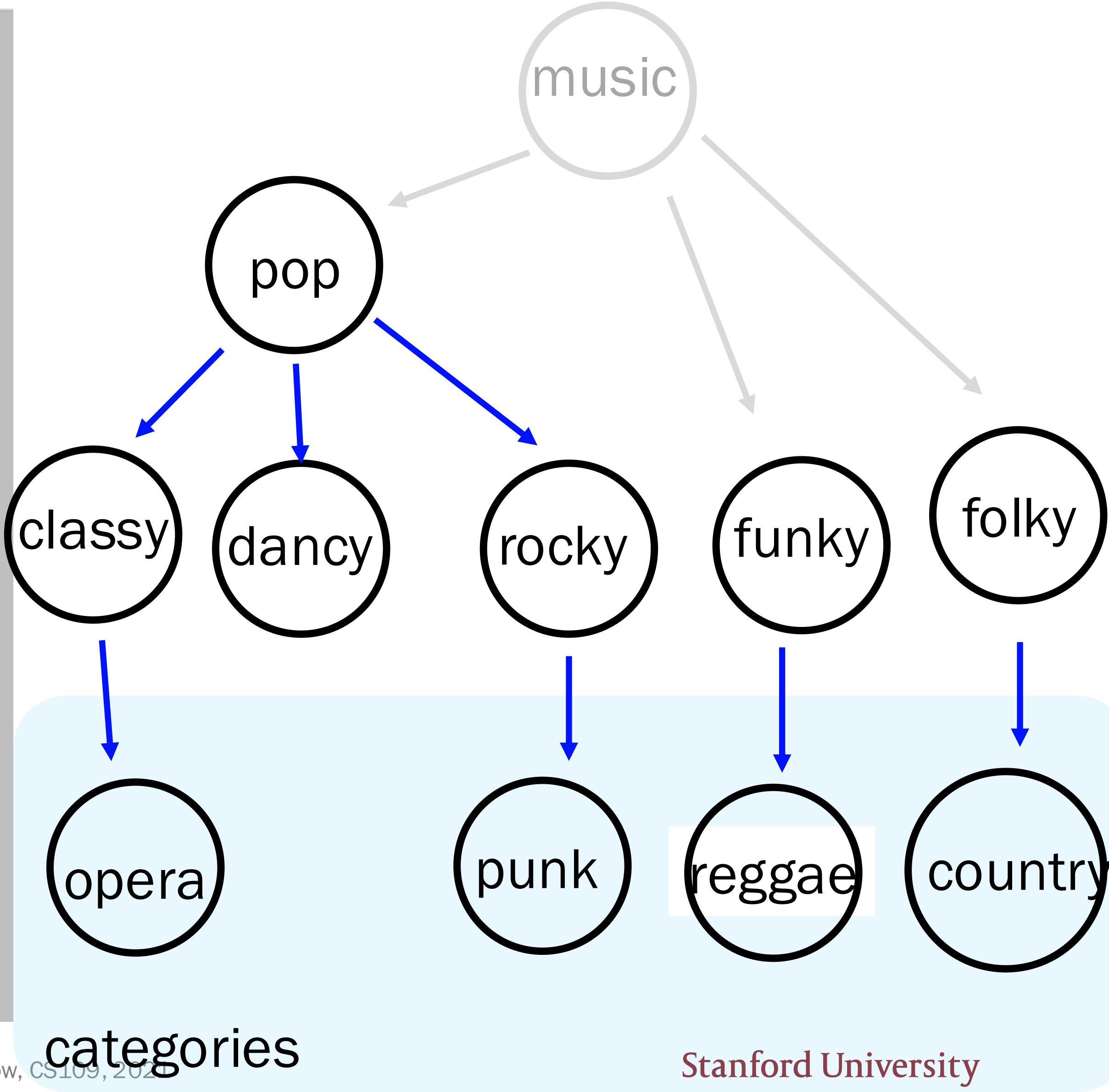
$$\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

- Note: $-1 \leq \rho(X, Y) \leq 1$

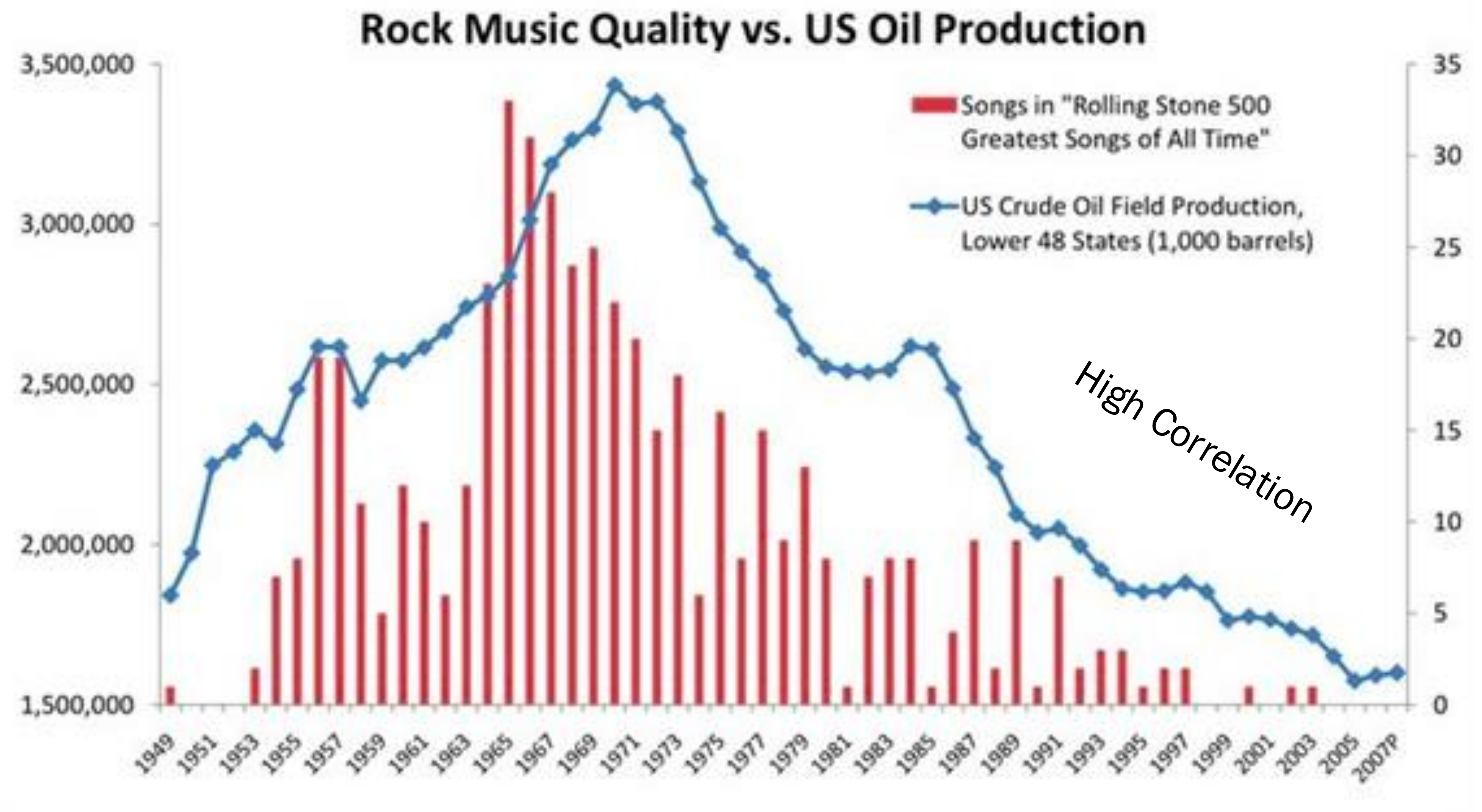
- $\rho(X, Y) = 1$ $\Rightarrow$ perfectly correlated
- $\rho(X, Y) = -1$ $\Rightarrow$ perfectly negatively correlated
- $\rho(X, Y) = 0$ $\Rightarrow$ absence of linear relationship
  - But, X and Y can still be related in some other way!
- If $\rho(X, Y) = 0$, we say X and Y are "uncorrelated"

Recall: It is a useful starting point





Rock Music Vs Oil?

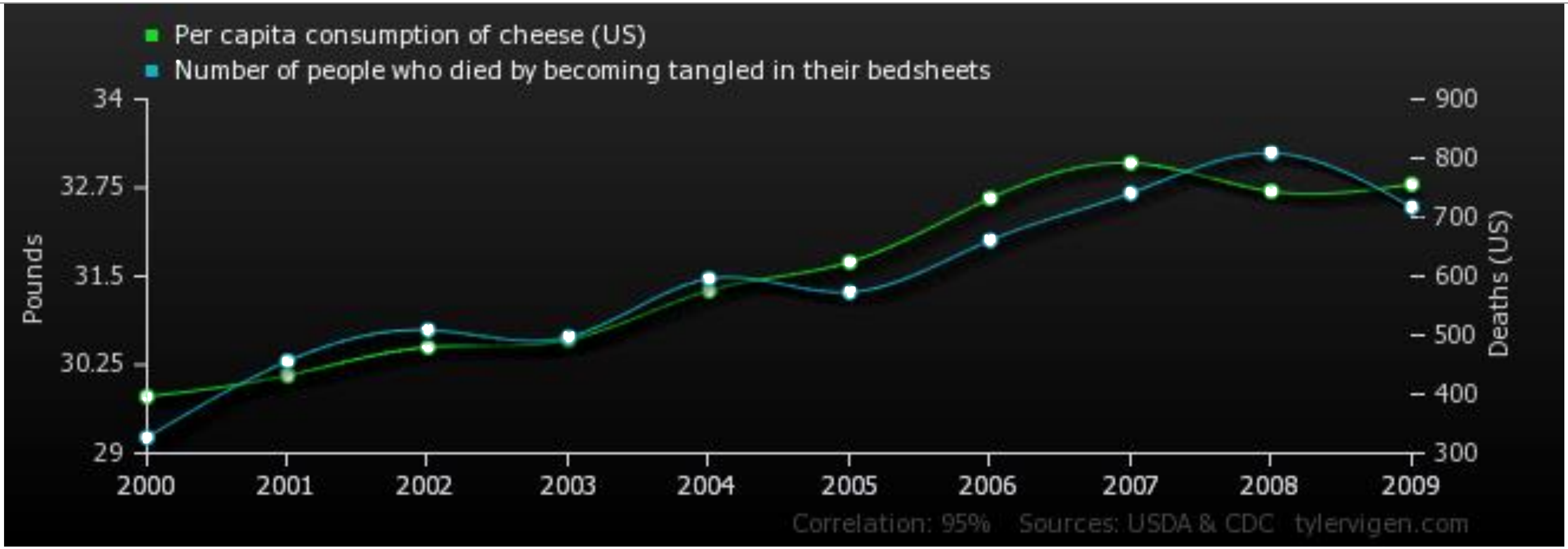


High Correlation

Hubbert Peak Theory

http://www.aei.org/publication/blog/

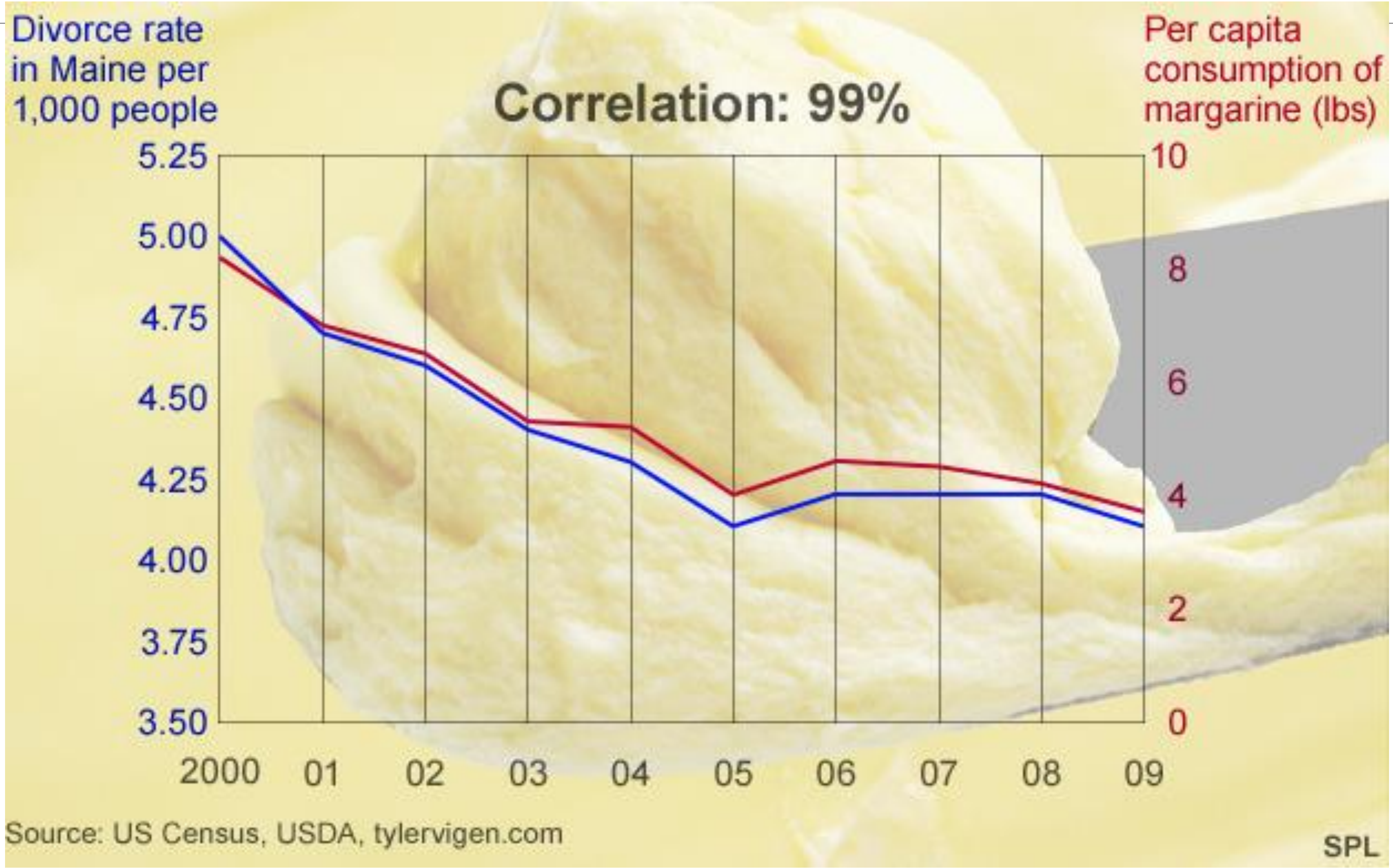
Tell your friends!



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

Correlation: 0.947091

Divorce Vs Butter?



http://www.bbc.com/news/magazine-27537142

Three Guiding Questions

1. How do people actually define large models?

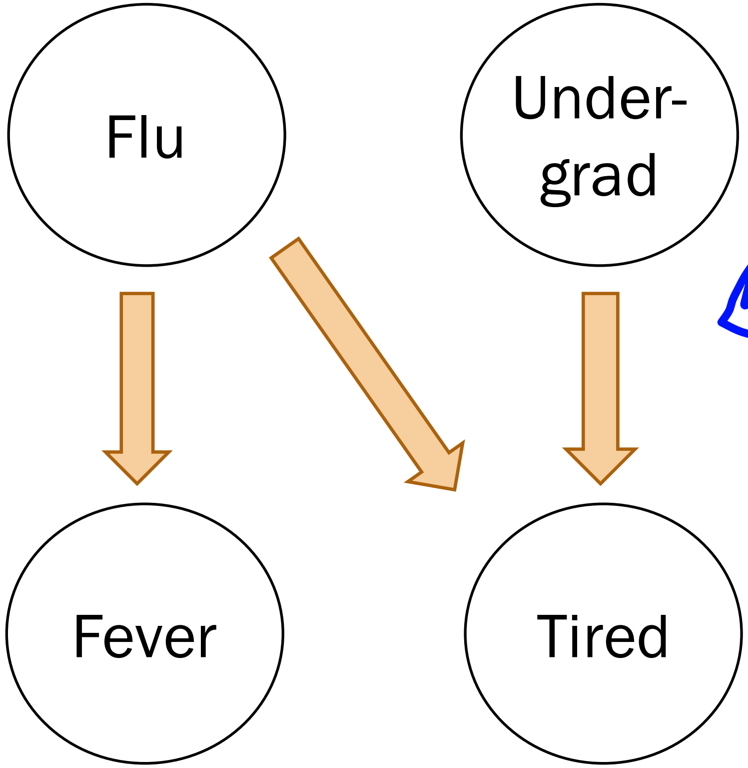
2. How can we do inference in large models?

3. What data can inform the design process?

What haven’t we talked about?

Machine Learning (last section of CS109)

$P(F_{lu} = 1) = 0.1$ $\quad$ $P(U = 1) = 0.8$



1. Learn this from data

2. Learn this from data

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

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