

# General Inference

CS109, Stanford University

**THE MOMENT YOU  
REALISE**

**IT'S FRIDAY**



# Learning Goals

1. Finish conversation on correlations
2. Learn rejection sampling



**BAYES NETS!**

# Multiple Random Variables. Start of Digital Revolution

---



# Multiple Random Variables. Start of Digital Revolution

## Conditions that match your symptoms

UNDERSTANDING YOUR RESULTS [i](#)

### Migraine headache (adult)



Moderate match



### Acute Sinusitis



Fair match



### Stroke



Fair match



Gender **Male**

Age **30**

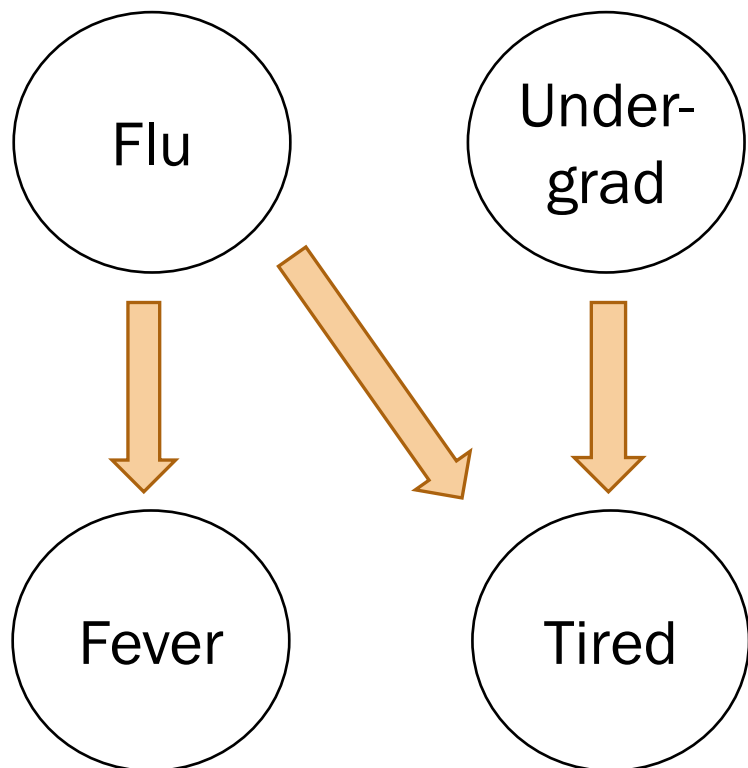
[Edit](#)

My Symptoms

[Edit](#)

**dizziness, one sided headache**

# Constructing a Bayesian Network



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

In a Bayesian Network,  
Each random variable is caused by  
its **parents**. Def  $P(\text{node} \mid \text{parents})$

- Node: random variable
- Directed edge: causality

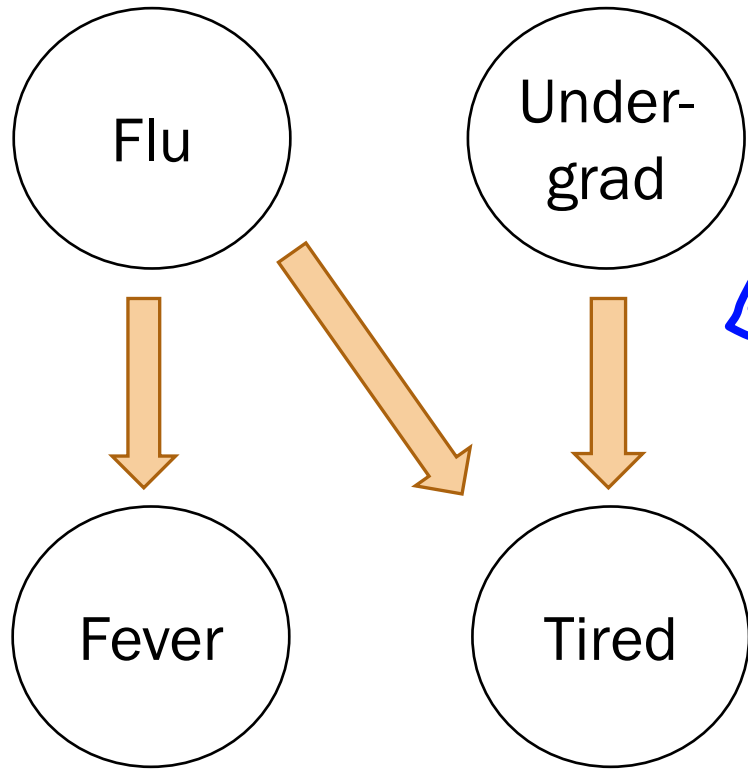
Examples:

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

# The art of modelling

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

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



1. Design this

2. Also design this.  
Later in CS109: learn  
this from data

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

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

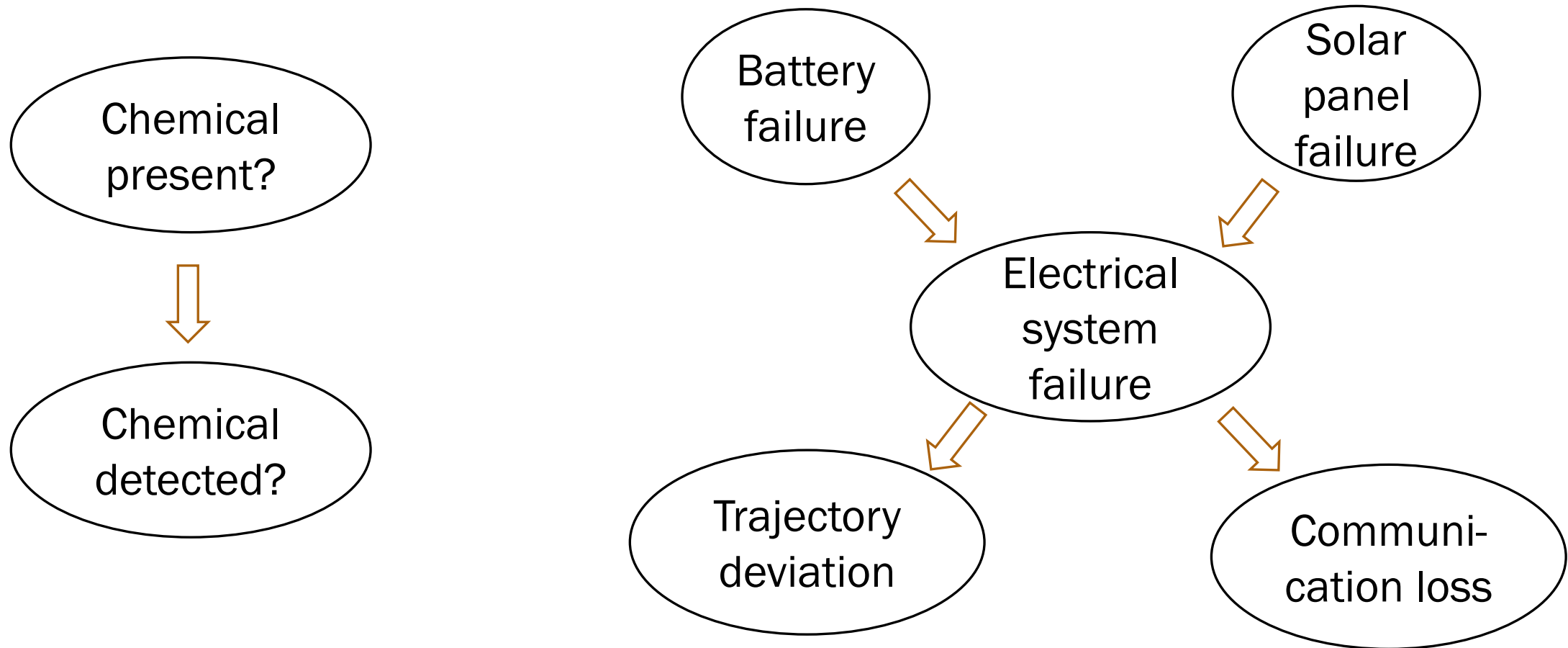
# Make a *Generative* Model

---



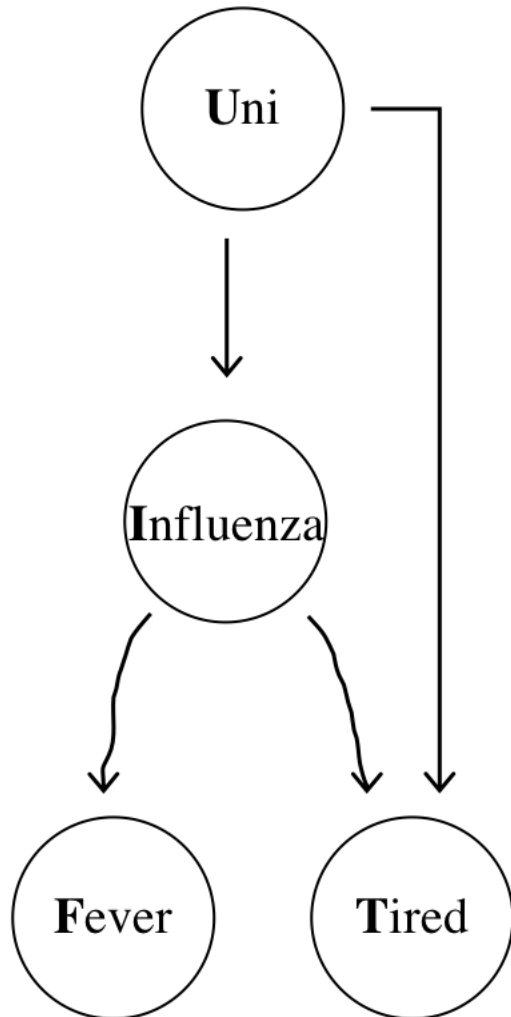
A good probabilistic model is **generative**. It explains the process through which the joint is **created**.

# Other applications



# Bayesian Network

## Simple Disease Model



```
def get_prob_Xi(x, parents):
```

```
    # what is the probability that Xi = x
```

```
    # given the list parents of assignments to
```

```
    # the parents variables Xi
```

$$P(\text{Uni} = 1) = 0.8$$

$$P(\text{Influenza} = 1 | \text{Uni} = 1) = 0.2$$

$$P(\text{Influenza} = 1 | \text{Uni} = 0) = 0.1$$

$$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 0) = 0.1$$

$$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 0) = 0.8$$

$$P(\text{Fever} = 1 | \text{Influenza} = 1) = 0.9$$

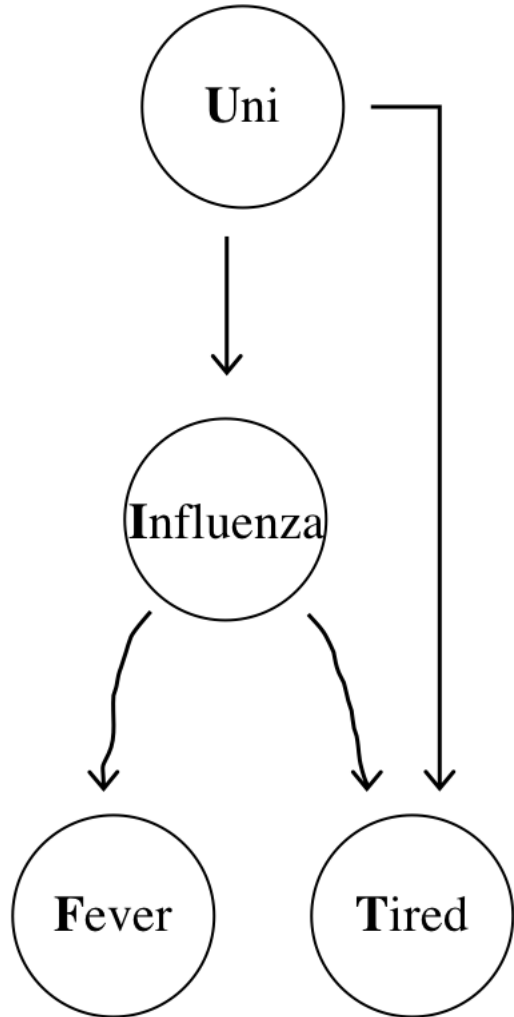
$$P(\text{Fever} = 1 | \text{Influenza} = 0) = 0.05$$

$$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 1) = 0.9$$

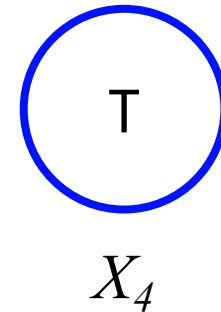
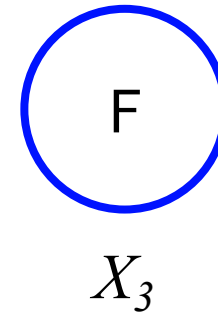
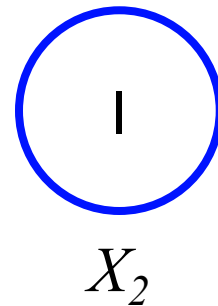
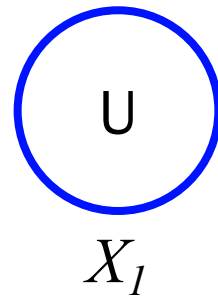
$$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 1) = 1.0$$

# Bayesian Network Assumption

## Simple Disease Model



Order nodes by ancestry



$$P(\text{Joint}) = \prod_i P(x_i | x_{i-1}, \dots, x_1)$$

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

Assume: Once you know the value of the parents of a variable in your network,  $X_i$ , any further information about non-descendants will not change your belief in  $X_i$ .

End Review

How do you tell if two RVs are related?

# ROCK

**The Sound:** Vigorous, defiant, energetic, inventive

**The Roots:** Rhythm & blues, country

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

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

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

"We're a rock group. We're noisy, raucous, emotional and wild."

— Angus Young (c. 1960)  
Lead guitarist of rock band AC/DC

# HIP-HOP R&B

**The Sound:** Rhythmic, unvarnished, adaptable, streetwise

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

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

**The Places:** New York City (South Bronx)

**The Ensemble:** Vinyl, turntable, vocals

"The beautiful thing about hip-hop is it's like an audio collage. You can take any form of music and do it in a hip-hop way and it'll be a hip-hop song."

— Tom Mchale (1971)  
Hip-hop artist

# LATIN American

**The Sound:** Syncopated, enthusiastic, diverse, vibrant

**The Roots:** Spain, Africa, Caribbean, South America

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

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

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

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

— Juan Manuel (c. 1940)  
Latin singer and composer

# Folk

**The Sound:** Grassroots, narrative, sincere, lyrical

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

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

**The Places:** Appalachia, Deep South, Western frontier

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

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

— Peter Seeger (c. 1940)  
Folk musician

# COUNTRY Western

**The Sound:** Genuine, uncomplicated, nostalgic, informal

**The Roots:** European ballads, folk and gospel songs

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

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

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

"Country music is three chords and the truth."

— Hank Williams (1917–1953)  
Country music singer

# CLASSICAL

**The Sound:** Intricate, polished, structured, harmonious

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

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

**The Places:** Austria, Germany, France, Italy

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

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

— Ludwig van Beethoven (1770–1827)  
Classical music composer

AutoSave OFF Search Sheet

Home Insert Page Layout Formulas Data >> Share

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles

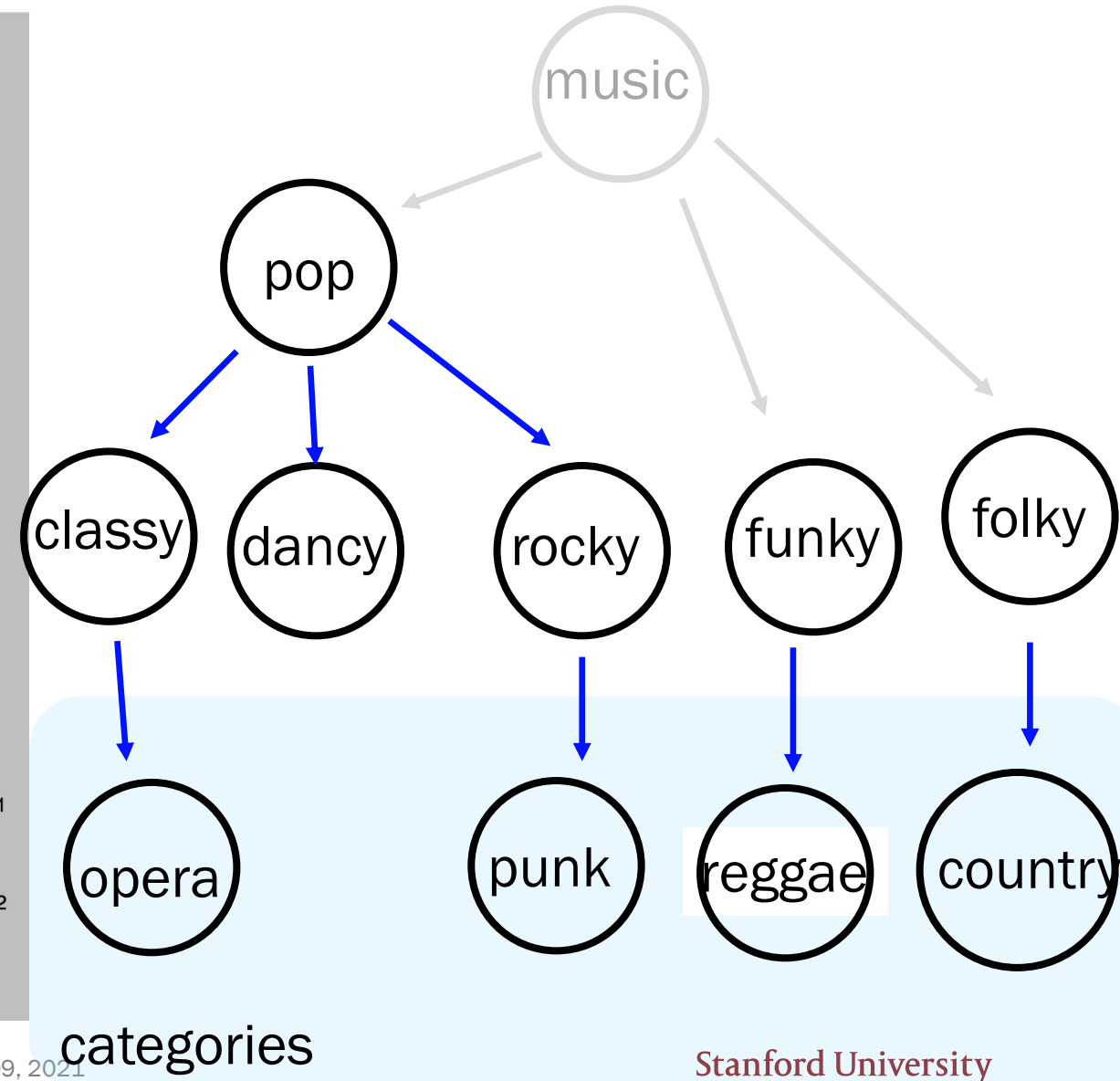
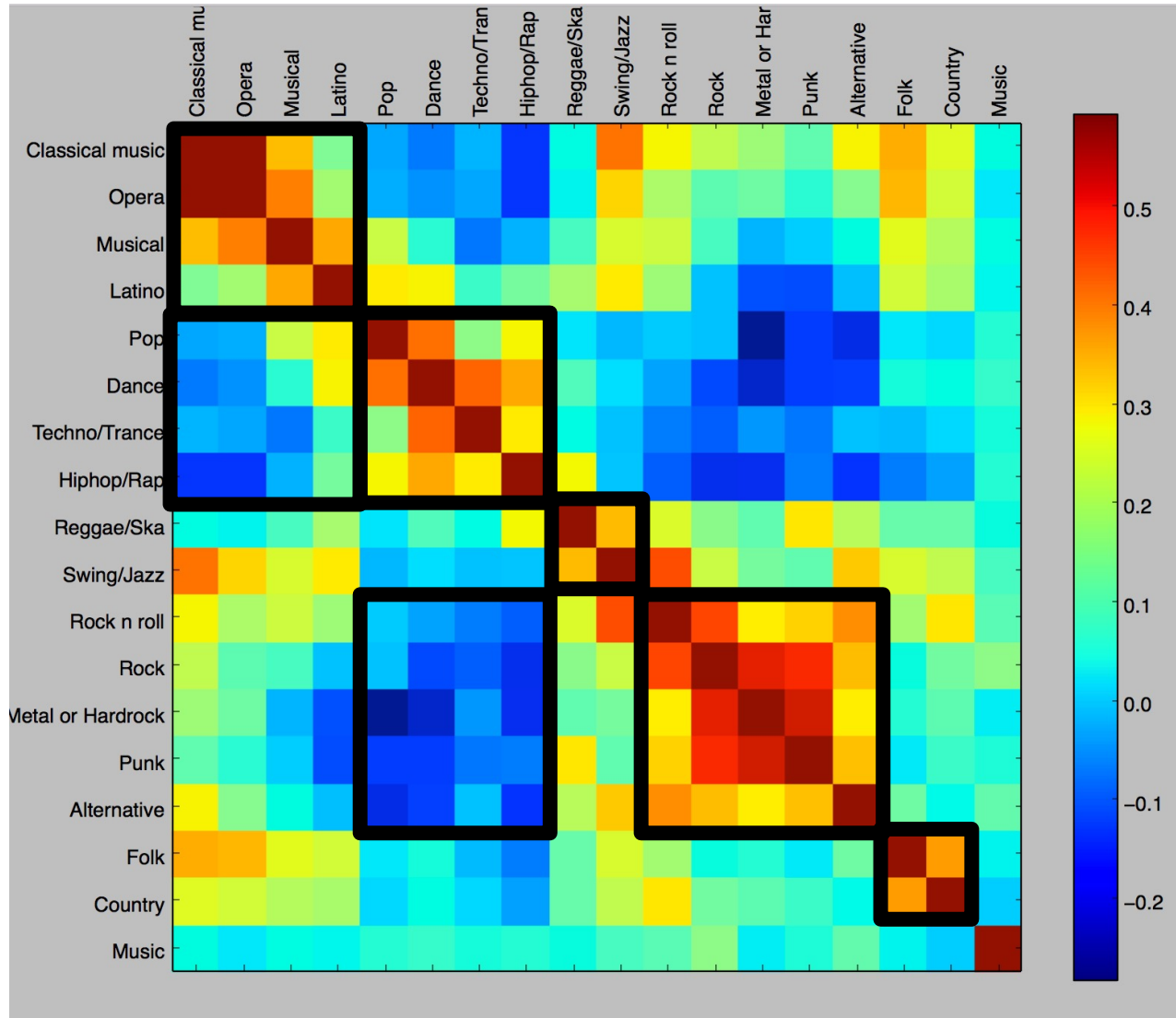
C15 fx 3

	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	

music +

Ready 100%

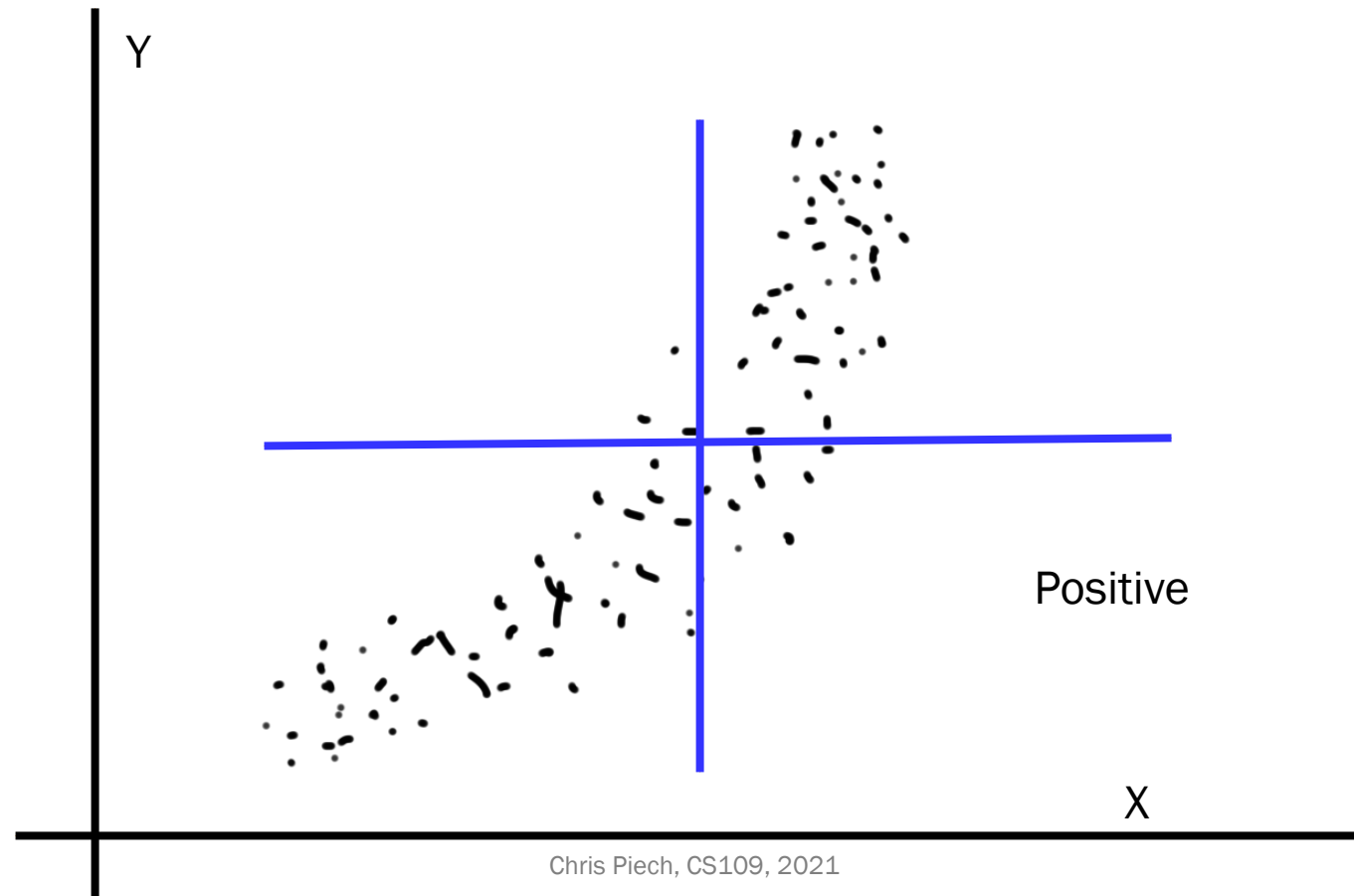
# From Correlation to Bayes Net. Alternative!



# Calculate the Covariance / Correlation (new stat!)

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

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



# Covariance of Zero Does Not Mean Independence!

---

$X$  and  $Y$  are random variables:

$X$  is -1, 0 or 1 with equal probability

$$Y = \begin{cases} 0 & \text{if } X \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

# Covariance of Zero Does Not Mean Independence!

$X$  and  $Y$  are random variables with PMF:

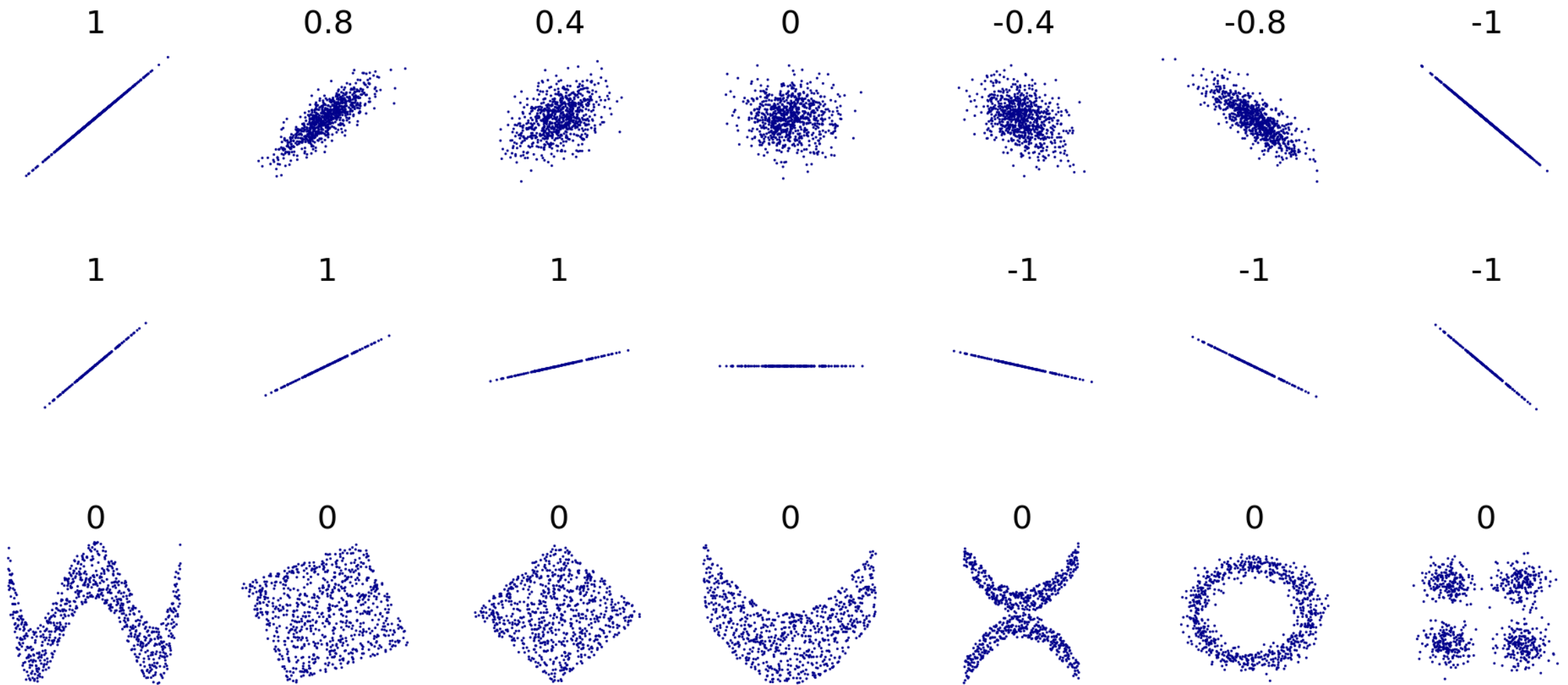
$Y \backslash X$	-1	0	1	$p_Y(y)$
0	1/3	0	1/3	2/3
1	0	1/3	0	1/3
$p_X(x)$	1/3	1/3	1/3	1

$$Y = \begin{cases} 0 & \text{if } X \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

- $E[X] = -1(1/3) + 0(1/3) + 1(1/3) = 0$
- $E[Y] = 0(2/3) + 1(1/3) = 1/3$
- Since  $XY = 0$ ,  $E[XY] = 0$
- $\text{Cov}(X, Y) = E[XY] - E[X]E[Y] = 0 - 0 = 0$

But,  $X$  and  $Y$  are clearly dependent!

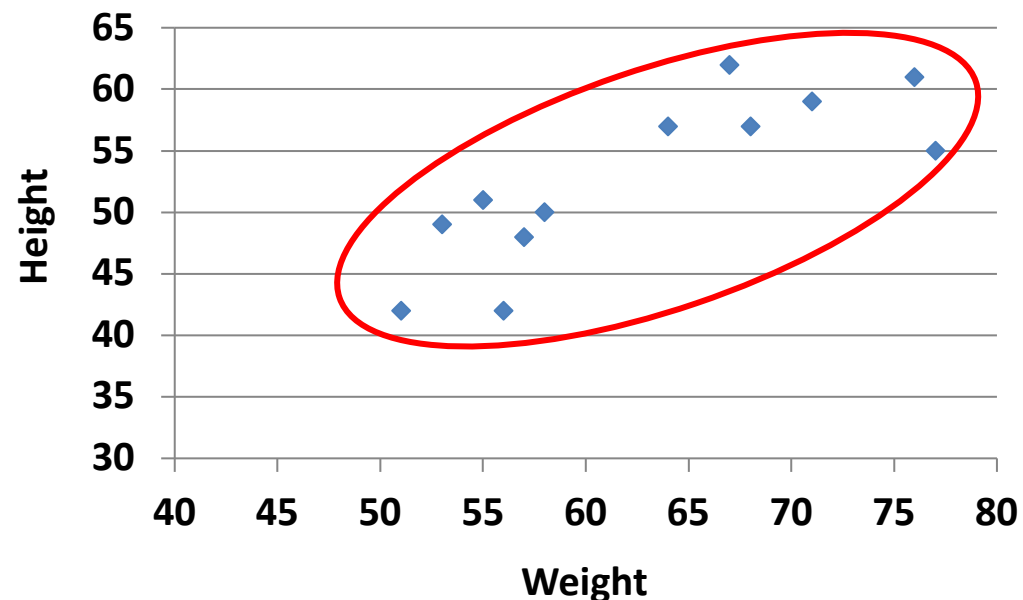
# Covariance of Zero Does Not Mean Independence!



# What is Wrong With This?

Consider the following data:

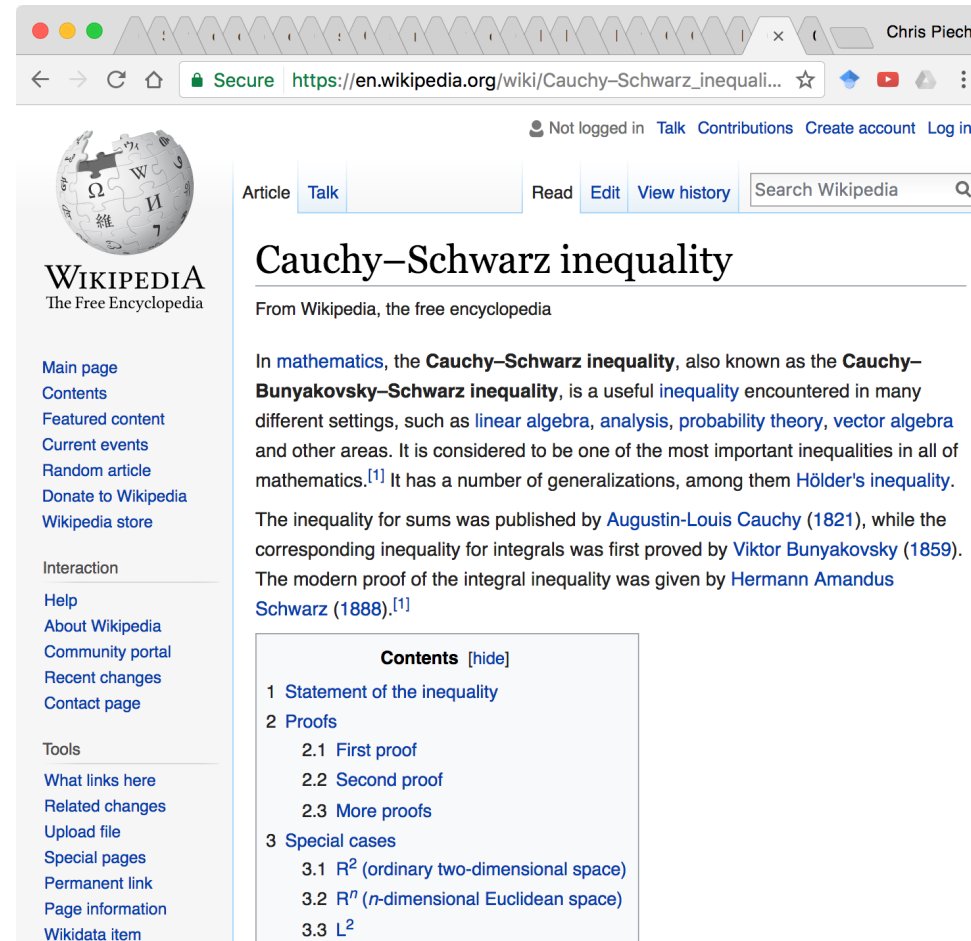
Weight	Height	Weight * Height
64	57	3648
71	59	4189
53	49	2597
67	62	4154
55	51	2805
58	50	2900
77	55	4235
57	48	2736
56	42	2352
51	42	2142
76	61	4636
68	57	3876



$$\begin{array}{lll} E[W] & E[H] & E[W*H] \\ = 62.75 & = 52.75 & = 3355.83 \end{array}$$

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

# Cauchy Schwarz, a great way to normalize!



The screenshot shows a web browser window displaying the Wikipedia article for "Cauchy–Schwarz inequality". The browser's address bar shows the URL "https://en.wikipedia.org/wiki/Cauchy–Schwarz\_inequali...". The page title is "Cauchy–Schwarz inequality". The article text states: "In **mathematics**, the **Cauchy–Schwarz inequality**, also known as the **Cauchy–Bunyakovsky–Schwarz inequality**, is a useful **inequality** encountered in many different settings, such as **linear algebra**, **analysis**, **probability theory**, **vector algebra** and other areas. It is considered to be one of the most important inequalities in all of mathematics.<sup>[1]</sup> It has a number of generalizations, among them **Hölder's inequality**. The inequality for sums was published by **Augustin-Louis Cauchy** (1821), while the corresponding inequality for integrals was first proved by **Viktor Bunyakovsky** (1859). The modern proof of the integral inequality was given by **Hermann Amandus Schwarz** (1888).<sup>[1]</sup>" Below the text is a "Contents" section with the following items: 1 Statement of the inequality, 2 Proofs (with sub-items 2.1 First proof, 2.2 Second proof, 2.3 More proofs), and 3 Special cases (with sub-items 3.1 R<sup>2</sup> (ordinary two-dimensional space), 3.2 R<sup>n</sup> (n-dimensional Euclidean space), 3.3 L<sup>2</sup>).

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

# Correlation is just normalized Covariance



Correlation

Covariance

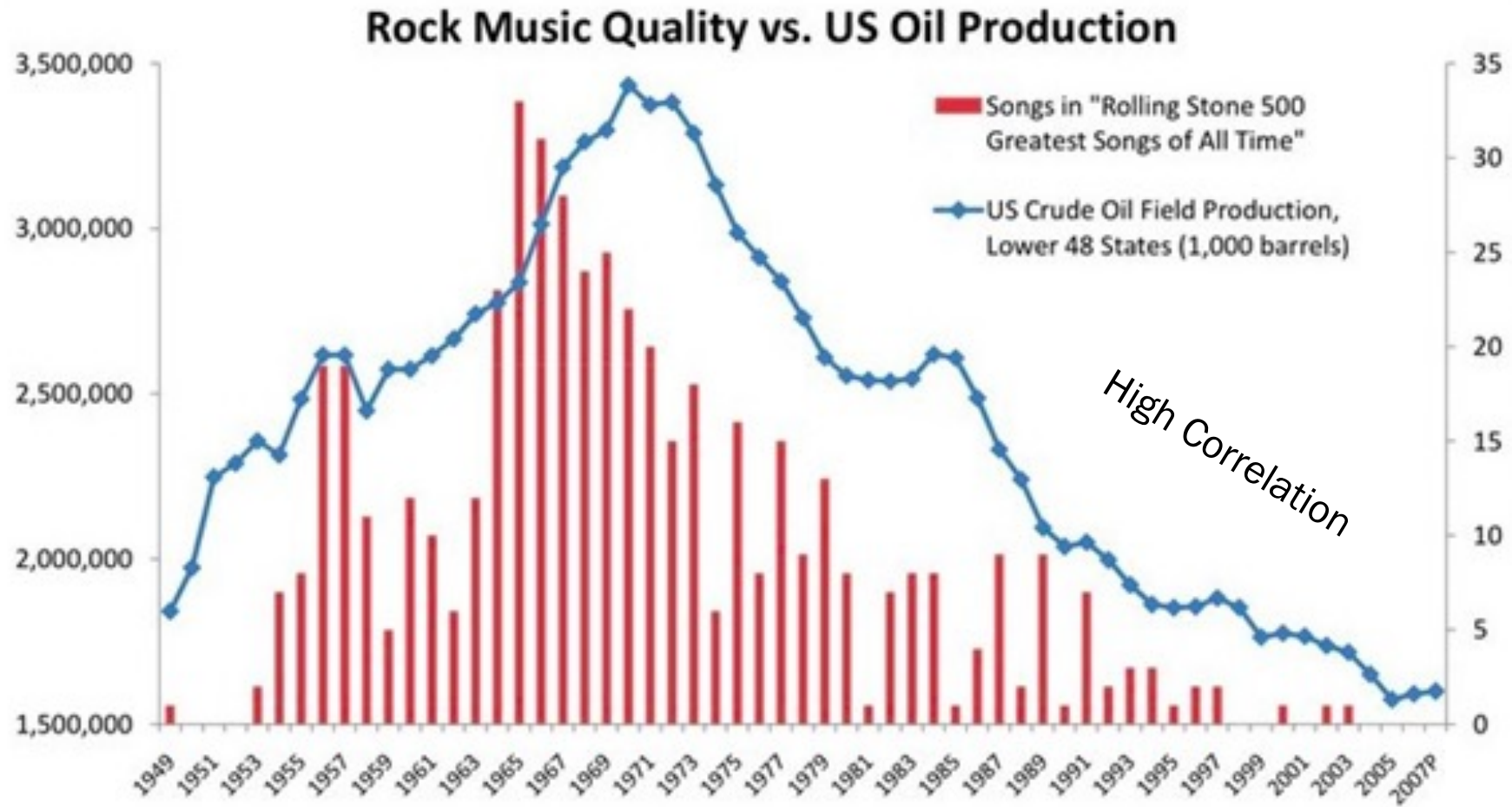
$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

It is always true that

$$\text{Cov}(X, Y) < \sqrt{\text{Var}(X)\text{Var}(Y)}$$

$$\text{Cov}(X, Y) > -\sqrt{\text{Var}(X)\text{Var}(Y)}$$

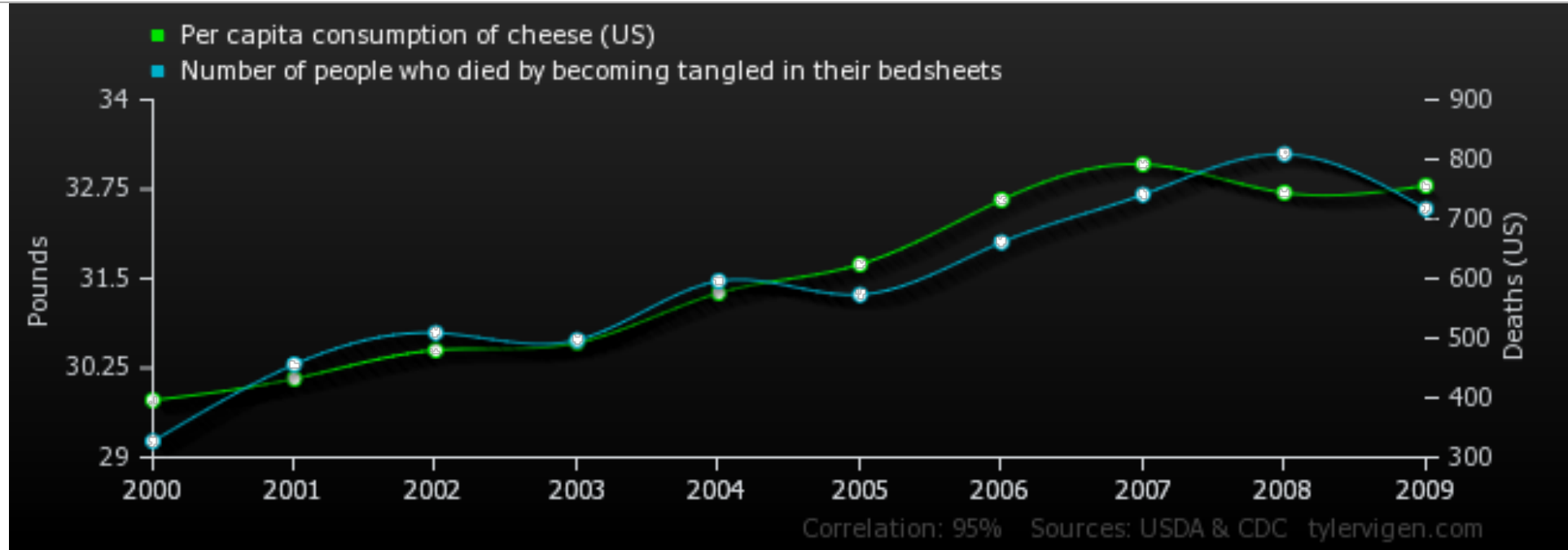
# Rock Music Vs Oil?



Hubbert Peak Theory

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

# Tell your friends!



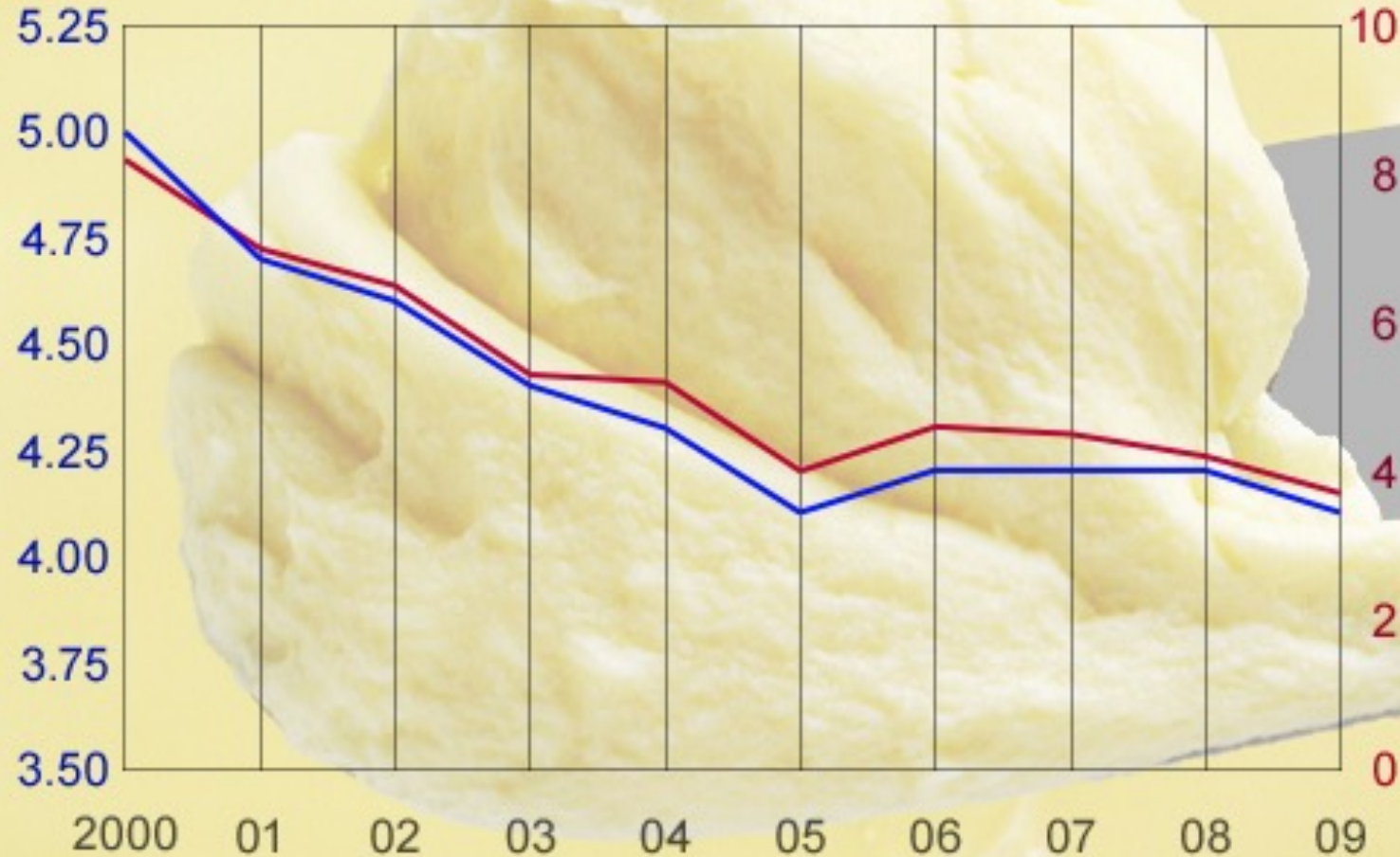
	<u>2000</u>	<u>2001</u>	<u>2002</u>	<u>2003</u>	<u>2004</u>	<u>2005</u>	<u>2006</u>	<u>2007</u>	<u>2008</u>	<u>2009</u>
<i>Per capita consumption of cheese (US) Pounds (USDA)</i>	29.8	30.1	30.5	30.6	31.3	31.7	32.6	33.1	32.7	32.8
<i>Number of people who died by becoming tangled in their bedsheets Deaths (US) (CDC)</i>	327	456	509	497	596	573	661	741	809	717
<b>Correlation: 0.947091</b>										

# Divorce Vs Butter?

Divorce rate  
in Maine per  
1,000 people

Per capita  
consumption of  
margarine (lbs)

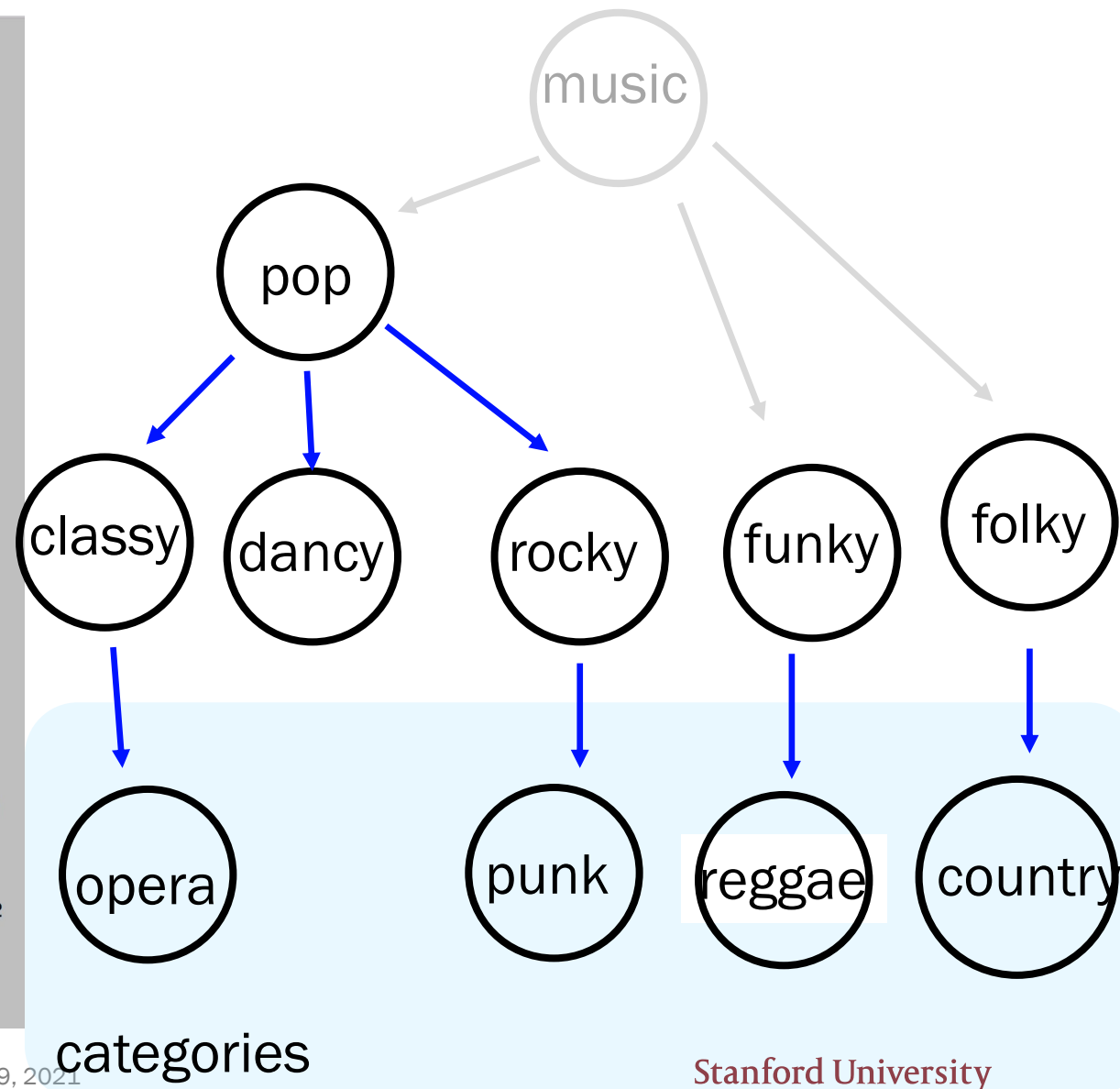
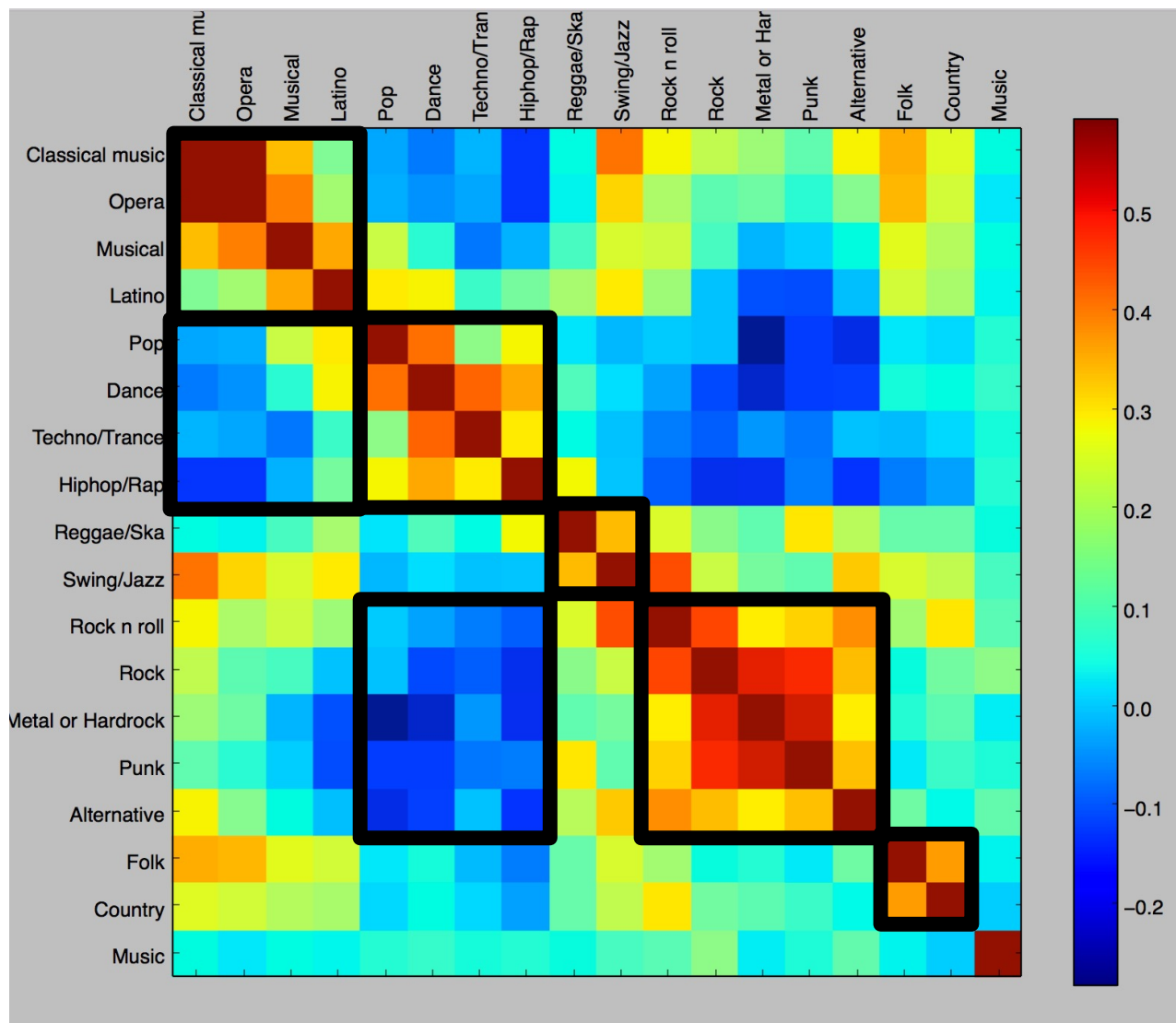
Correlation: 99%



Source: US Census, USDA, tylervigen.com

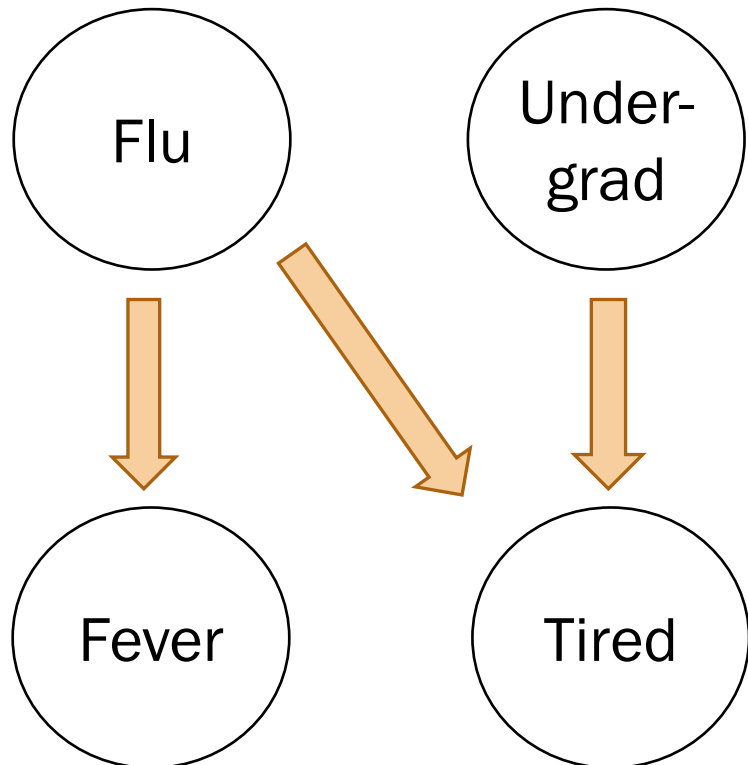
SPL

# Recall: It is a useful starting point



We have models. Need to solve  
problems

# Inference: Algebra



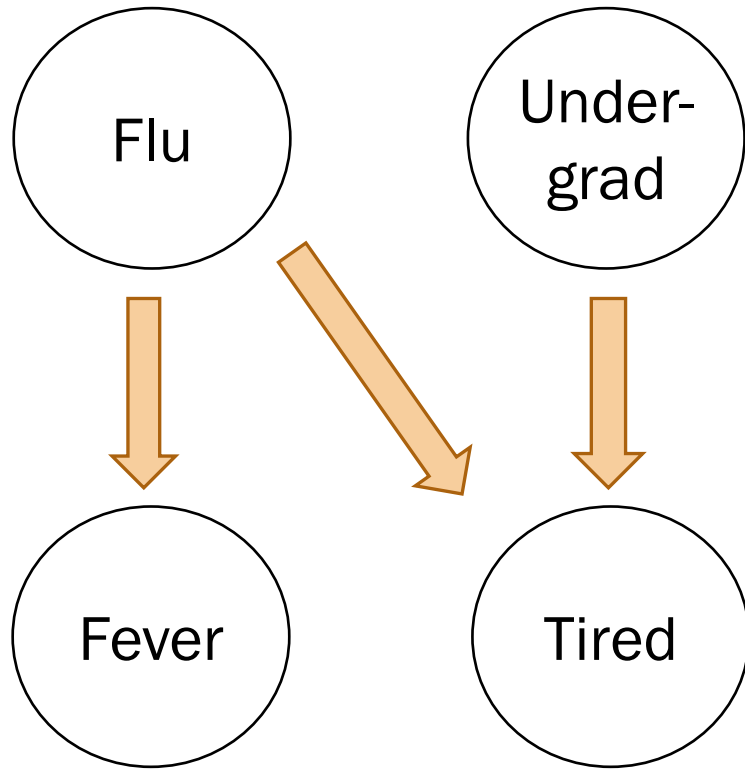
In a Bayesian Network,  
Each random variable is **conditionally independent** of its non-descendants, **given its parents**.

- Node: random variable
- Directed edge: conditional dependency

# Inference via math

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

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



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

Compute joint probabilities using chain rule.

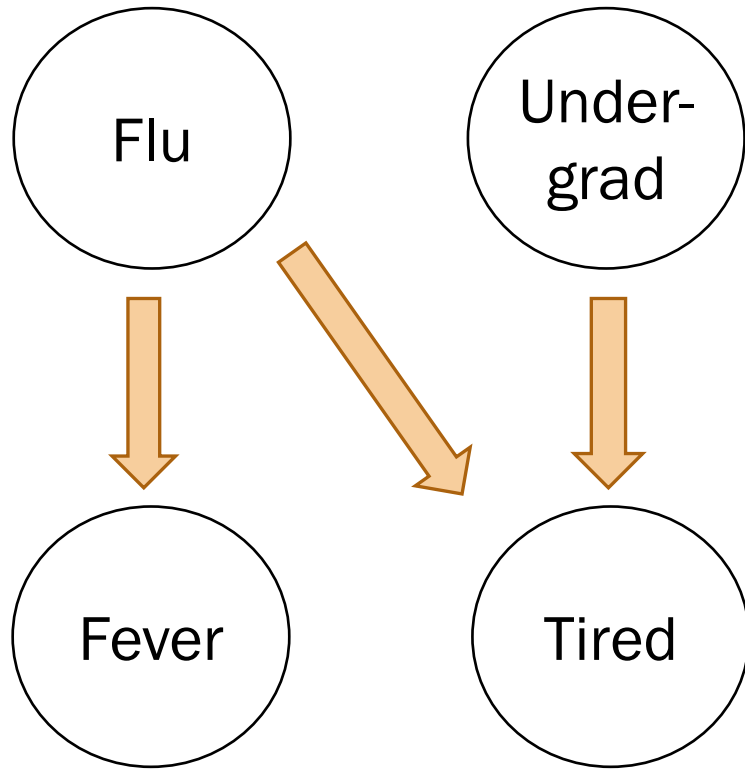
$$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$$
$$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$$

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

# Inference via math

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

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



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

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

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

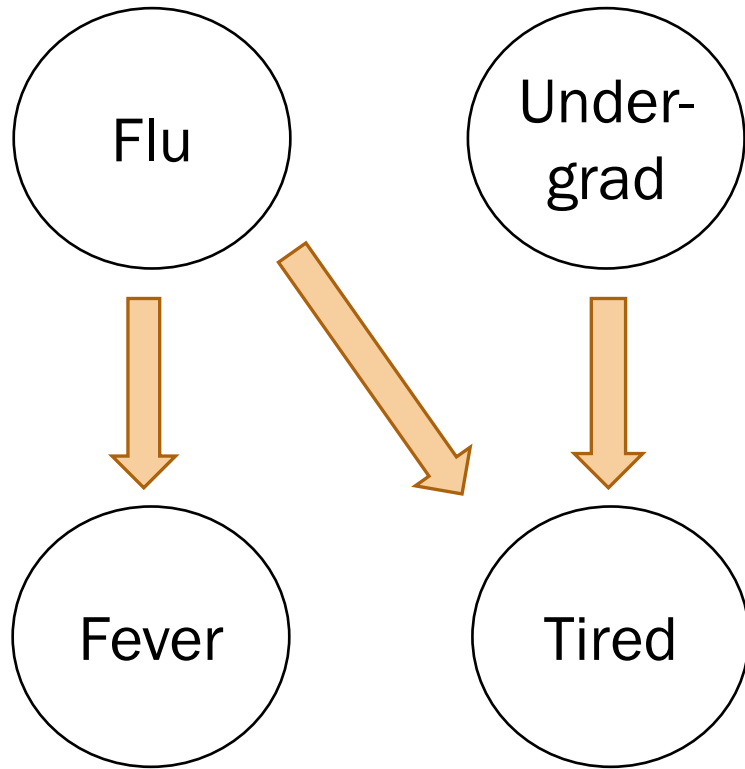
$$\frac{P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)}{\sum_x P(F_{lu} = x, F_{ev} = 0, U = 0, T = 1)}$$

$$= 0.095$$

# Inference via math

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

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



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

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

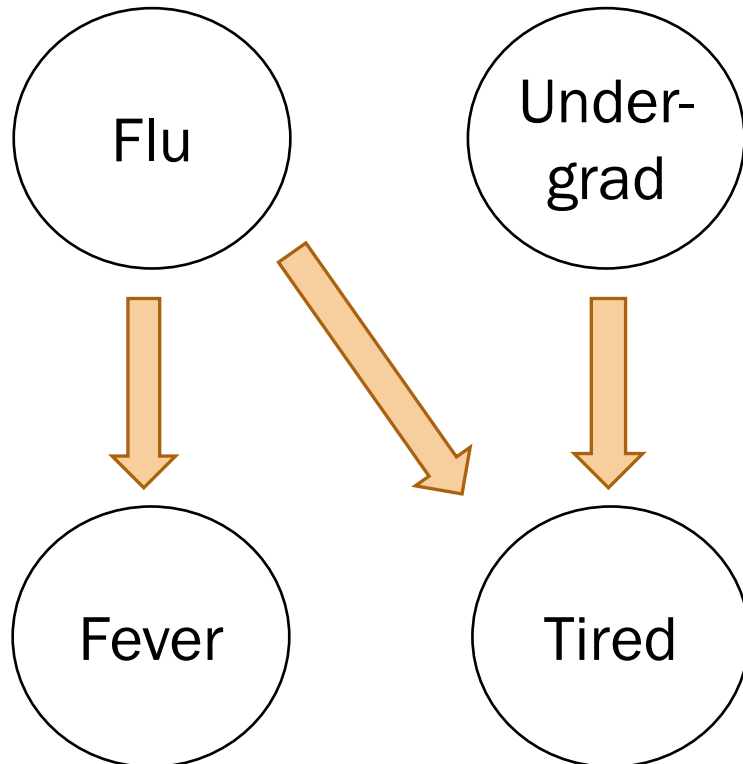
$$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$$
$$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$$
$$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$$
$$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$$



# Inference via math

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

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



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

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

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

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

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

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

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

1. Compute joint probabilities

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

...

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

2. Definition of conditional probability

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

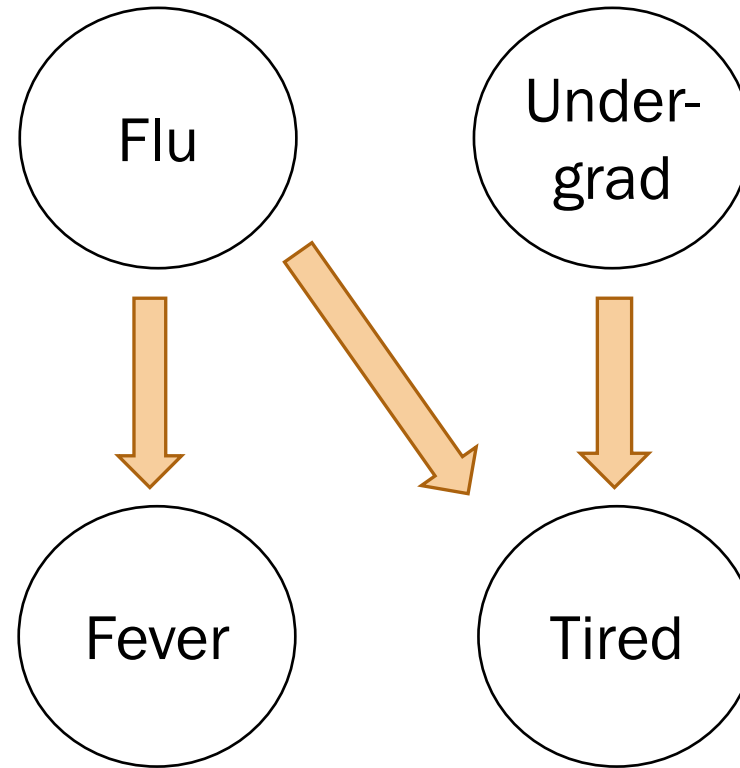
$$= 0.122$$

# Rejection sampling algorithm

Step 0:  
Have a fully specified  
Bayesian Network

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

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



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

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

# Alg #0: Straight Math

---

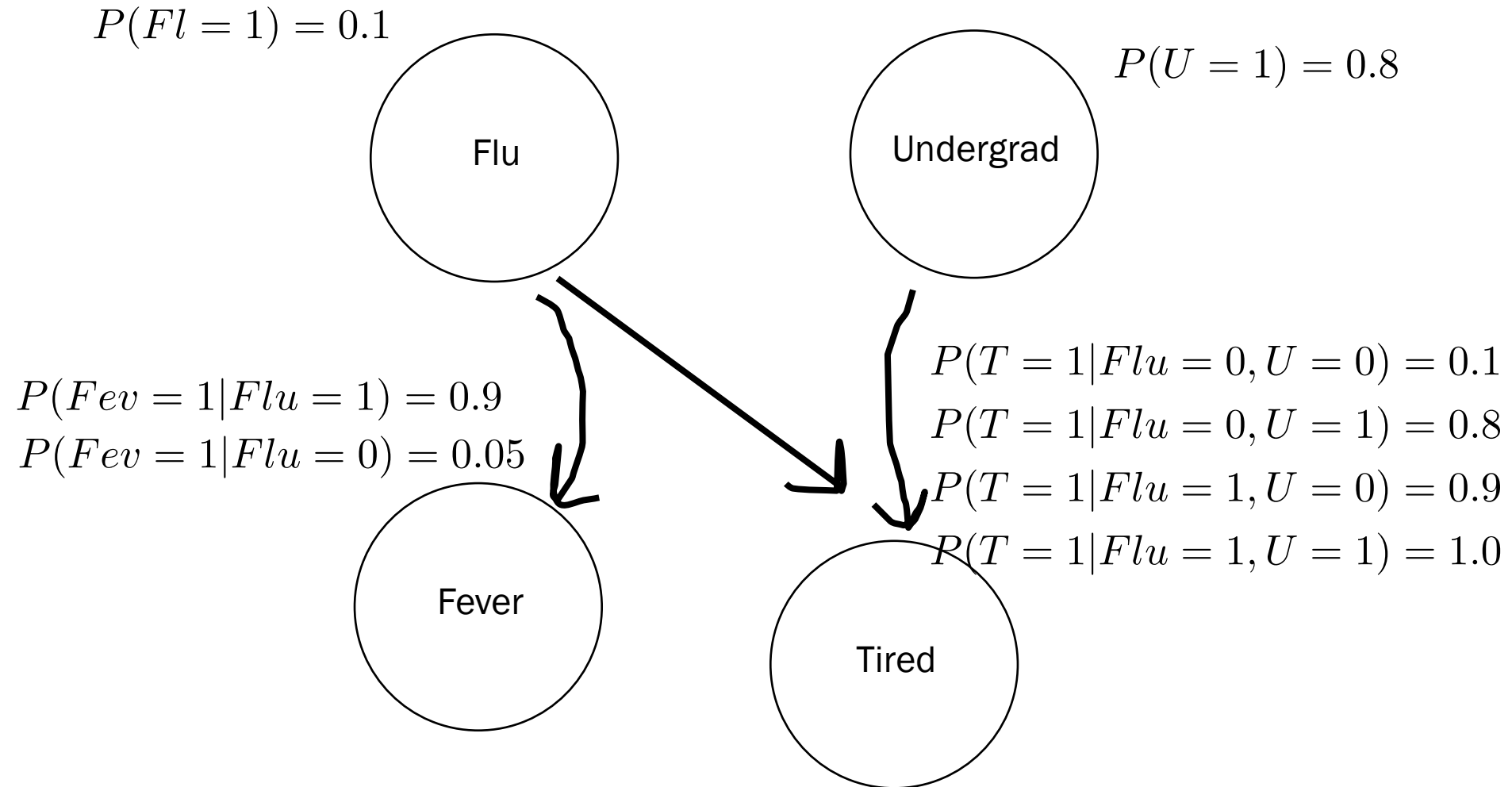
Too many possible **inference** questions one could ask...

# Alg #1: Rejection Sampling

```
3 N_SAMPLES = 100000
4
5 # Program: Joint Sample
6 # -----
7 # we can answer any probability question
8 # with multivariate samples from the joint,
9 # where conditioned variables match
10 def main():
11     obs = getObservation()
12     print 'Observation = ', obs
13
14     samples = sampleATon()
15     prob = probFluGivenObs(samples, obs)
16     print 'Pr(Flu) = ', prob
```

```
71 # Method: Sample A Ton
72 # -----
73 # chose N_SAMPLES with likelihood proportional
74 # to the joint distribution
75 def sampleATon():
76     samples = []
77     for i in range(N_SAMPLES):
78         sample = makeSample()
79         samples.append(sample)
80     return samples
```

# Recall: Probabilistic Model



```
82 # Method: Make Sample
83 # -----
84 # chose a single sample from the joint distribut
85 # based on the medical "Probabilistic Graphical
86 def makeSample():
87     # prior on causal factors
88     flu = bern(0.1)
89     und = bern(0.8)
90
91     # choose fever based on flue
92     if flu == 1: fev = bern(0.9)
93     else:       fev = bern(0.05)
94
95     # choose tired based on (undergrade and flu)
96     if und == 1 and flu == 1:   tir = bern(1.0)
97     elif und == 1 and flu == 0: tir = bern(0.8)
98     elif und == 0 and flu == 1: tir = bern(0.9)
99     else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82 # Method: Make Sample
83 # -----
84 # chose a single sample from the joint distribut
85 # based on the medical "Probabilistic Graphical
86 def makeSample():
87     # prior on causal factors
88     flu = bern(0.1)
89     und = bern(0.8)
90
91     # choose fever based on flue
92     if flu == 1: fev = bern(0.9)
93     else:       fev = bern(0.05)
94
95     # choose tired based on (undergrade and flu)
96     if und == 1 and flu == 1:   tir = bern(1.0)
97     elif und == 1 and flu == 0: tir = bern(0.8)
98     elif und == 0 and flu == 1: tir = bern(0.9)
99     else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82 # Method: Make Sample
83 # -----
84 # chose a single sample from the joint distribut
85 # based on the medical "Probabilistic Graphical
86 def makeSample():
87     # prior on causal factors
88     flu = bern(0.1)
89     und = bern(0.8)
90
91     # choose fever based on flue
92     if flu == 1: fev = bern(0.9)
93     else:       fev = bern(0.05)
94
95     # choose tired based on (undergrade and flu)
96     if und == 1 and flu == 1:   tir = bern(1.0)
97     elif und == 1 and flu == 0: tir = bern(0.8)
98     elif und == 0 and flu == 1: tir = bern(0.9)
99     else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82 # Method: Make Sample
83 # -----
84 # chose a single sample from the joint distribut
85 # based on the medical "Probabilistic Graphical
86 def makeSample():
87     # prior on causal factors
88     flu = bern(0.1)
89     und = bern(0.8)
90
91     # choose fever based on flue
92     if flu == 1: fev = bern(0.9)
93     else:         fev = bern(0.05)
94
95     # choose tired based on (undergrade and flu)
96     if und == 1 and flu == 1:   tir = bern(1.0)
97     elif und == 1 and flu == 0: tir = bern(0.8)
98     elif und == 0 and flu == 1: tir = bern(0.9)
99     else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

# Alg #1: Rejection Sampling

```
1
2
3 N_SAMPLES = 100000
4
5 # Program: Joint Sample
6 # -----
7 # we can answer any pro
8 # with multivariate sam
9 # where conditioned var
10 def main():
11     obs = getObservatio
12     print 'Observation
13
14     samples = sampleATo
15     prob = probFluGiven
16     print 'Pr(Flu) = ',
17
```

```
webMd — -bash — 30x20
[0, 1, 0, 1]
[1, 1, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[1, 1, 0, 1]
```

# Alg #1: Rejection Sampling

```
3 N_SAMPLES = 100000
4
5 # Program: Joint Sample
6 # -----
7 # we can answer any probability question
8 # with multivariate samples from the joint,
9 # where conditioned variables match
10 def main():
11     obs = getObservation()
12     print 'Observation = ', obs
13
14     samples = sampleATon()
15     prob = probFluGivenObs(samples, obs)
16     print 'Pr(Flu) = ', prob
```

```
25 # Method: Probability of Flu Given Observation
26 # -----
27 # Calculate the probability of flu given many
28 # samples from the joint distribution and a set
29 # of observations to condition on.
30 def probFluGivenObs(samples, obs):
31     # reject all samples which don't align
32     # with condition
33     keepSamples = []
34     for sample in samples:
35         if checkObsMatch(sample, obs):
36             keepSamples.append(sample)
37
38     # from remaining, simply count...
39     fluCount = 0
40     for sample in keepSamples:
41         [flu, und, fev, tir] = sample
42         if flu == 1:
43             fluCount += 1
44
45     # counting can be so sweet...
46     return float(fluCount) / len(keepSamples)
```

```

25 # Method: Probability of Flu Given Observation
26 # -----
27 # Calculate the probability of flu given many
28 # samples from the joint distribution and a set
29 # of observations to condition on.
30 def probFluGivenObs(samples, obs):
31     # reject all samples which don't align
32     # with condition
33     keepSamples = []
34     for sample in samples:
35         if checkObsMatch(sample, obs):
36             keepSamples.append(sample)
37
38     # from remaining, simply count...
39     fluCount = 0
40     for sample in keepSamples:
41         [flu, und, fev, tir] = sample
42         if flu == 1:
43             fluCount += 1
44
45     # counting can be so sweet...
46     return float(fluCount) / len(keepSamples)

```

```
25 # Method: Probability of Flu Given Observation
26 # -----
27 # Calculate the probability of flu given many
28 # samples from the joint distribution and a set
29 # of observations to condition on.
30 def probFluGivenObs(samples, obs):
31     # reject all samples which don't align
32     # with condition
33     keepSamples = []
34     for sample in samples:
35         if checkObsMatch(sample, obs):
36             keepSamples.append(sample)
37
38     # from remaining, simply count...
39     fluCount = 0
40     for sample in keepSamples:
41         [flu, und, fev, tir] = sample
42         if flu == 1:
43             fluCount += 1
44
45     # counting can be so sweet...
46     return float(fluCount) / len(keepSamples)
```



Lets try it!

**BACK** ←  
*TO* **CODE**  
*THE*

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:

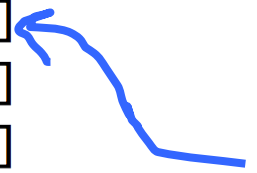
$$P(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n}$$

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

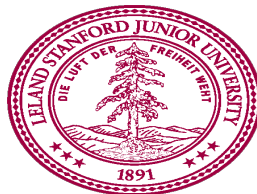


```
webMd — -bash — 39x20
[0, 1, 1, 0]
[1, 0, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 1, 0]
[1, 1, 1, 1]
[0, 1, 0, 0]
[0, 0, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 1]
Observation = [None, None, None, None]
Pr(Flu | Obs) = 0.10164
>
```

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



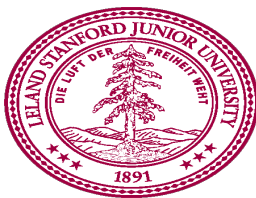
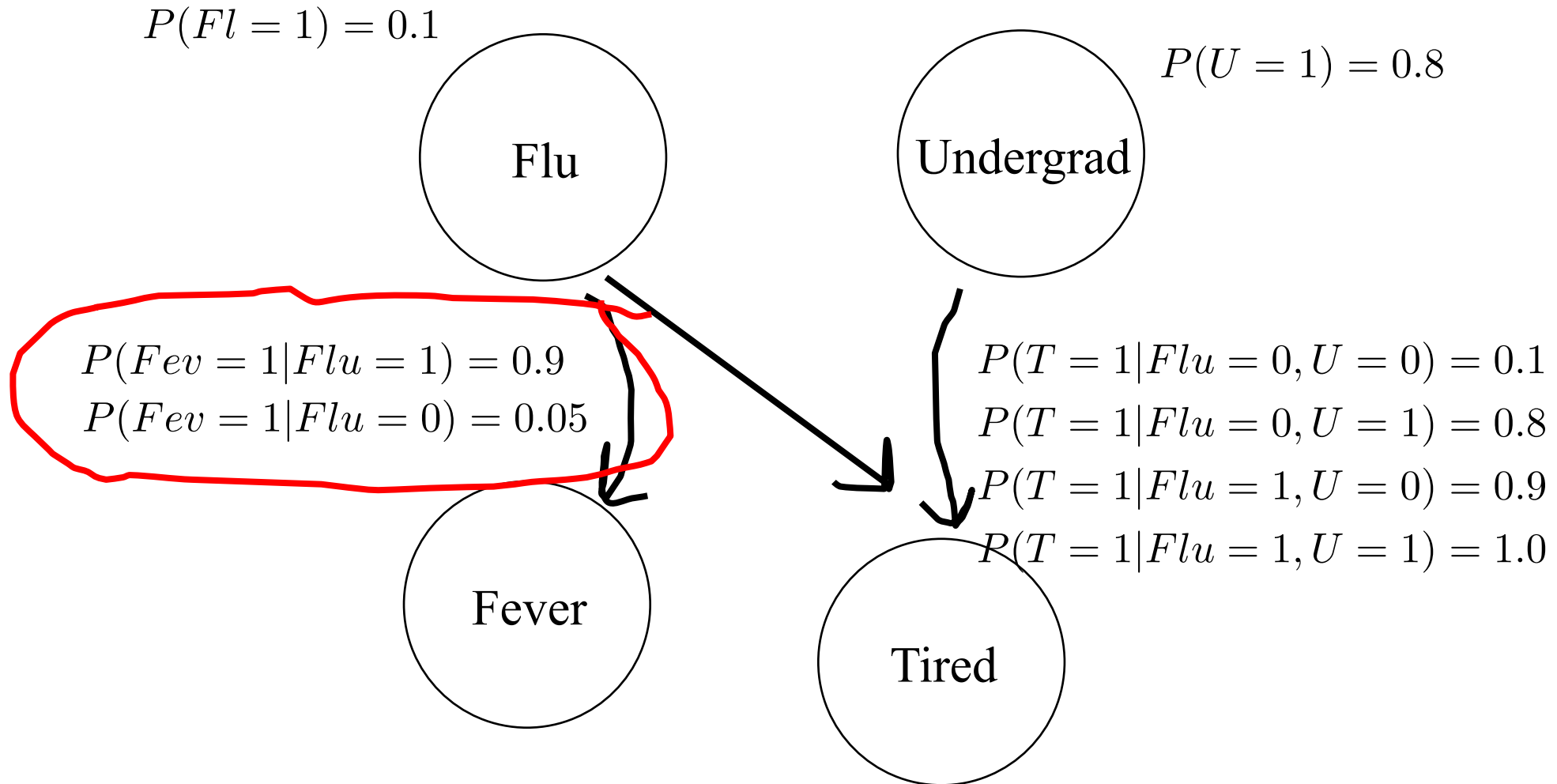
Each one of these is one joint sample:  
[Flu, Undergrad, Fever, Tired]



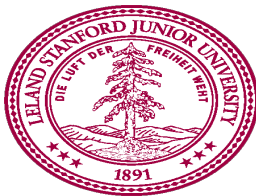
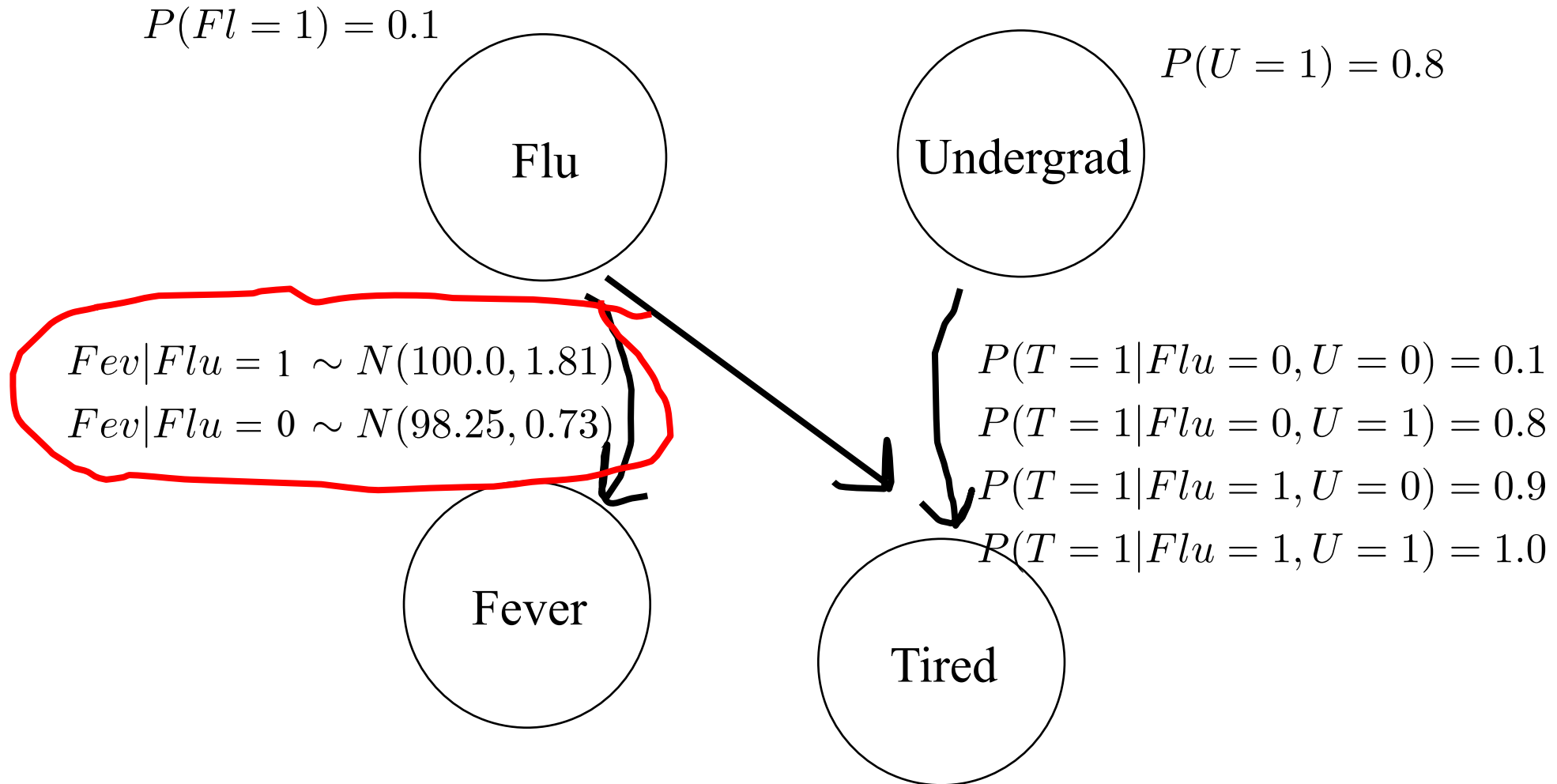
What's the matter with  
*rejection* sampling?



# Probabilistic Model



# Probabilistic Model



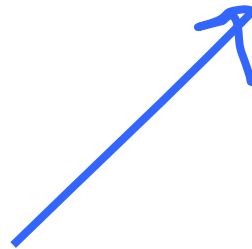
# The Magic School Bus™



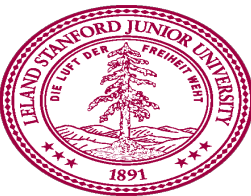
Markov Chain



**MCMC**



Monte Carlo



# Alg #2: MCMC

```
webmd -- -bash -- 10x20
[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, 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



What haven't we talked about?

Have a great weekend!