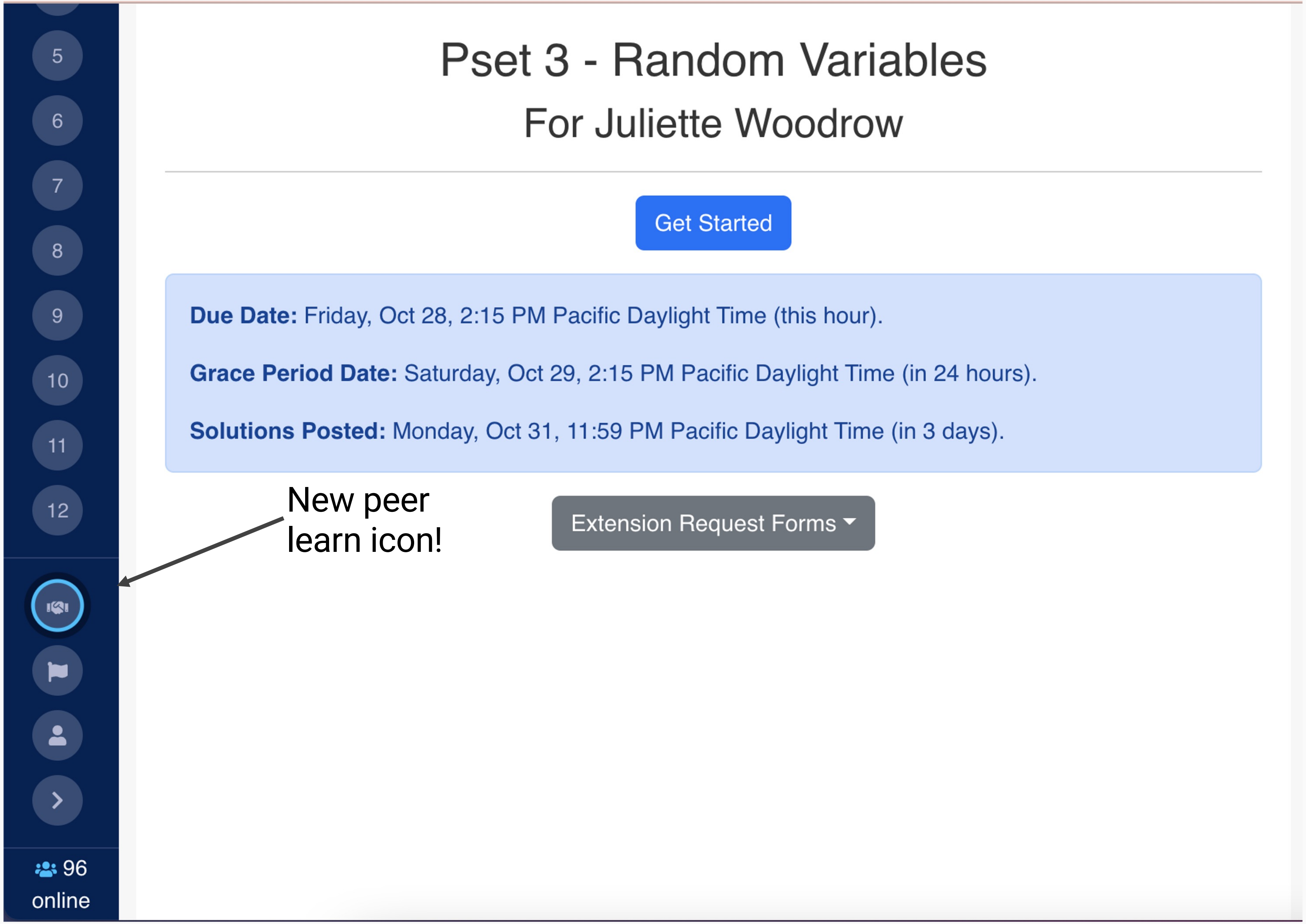
General Inference

Chris Piech
CS109, Stanford University

THE MOMENT YOU REALISE
IT'S FRIDAY
memegenerator.net

Midterm Review

(5pm today)



Pset 3 - Random Variables

For Juliette Woodrow

Get Started

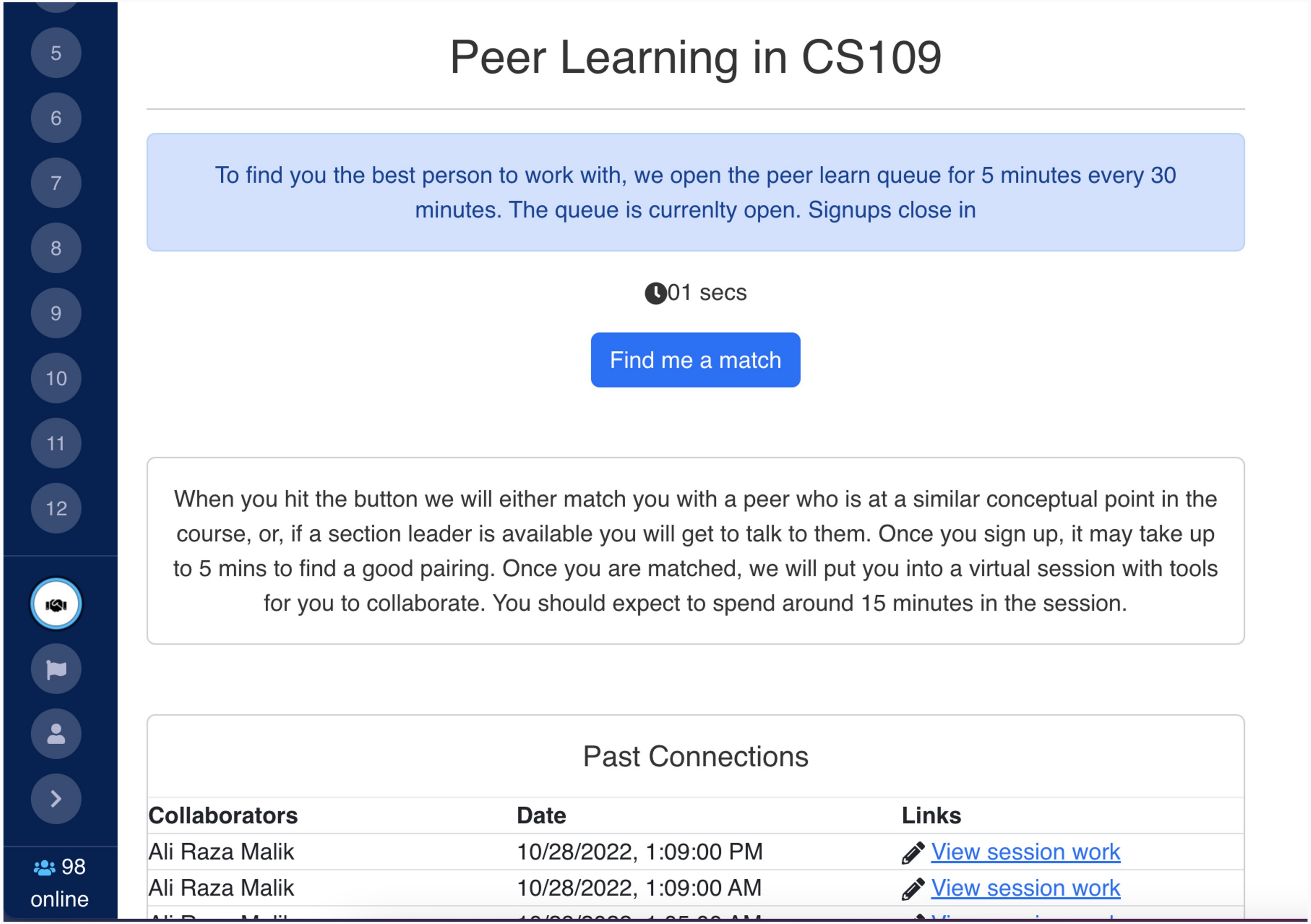
Due Date: Friday, Oct 28, 2:15 PM Pacific Daylight Time (this hour).

Grace Period Date: Saturday, Oct 29, 2:15 PM Pacific Daylight Time (in 24 hours).

Solutions Posted: Monday, Oct 31, 11:59 PM Pacific Daylight Time (in 3 days).

Extension Request Forms

New peer learn icon!



Peer Learning in CS109

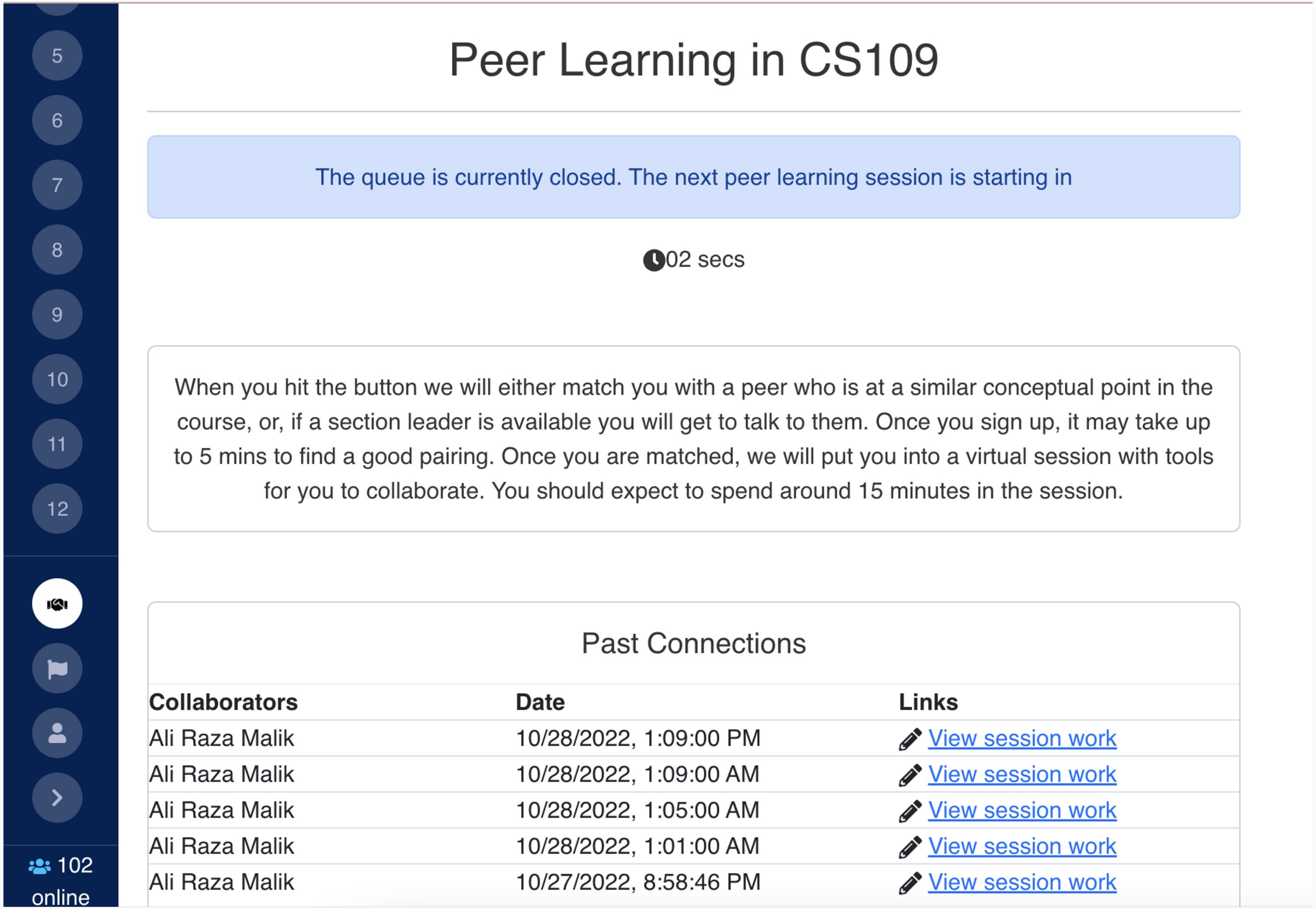
To find you the best person to work with, we open the peer learn queue for 5 minutes every 30 minutes. The queue is currenlty open. Signups close in

01 secs

When you hit the button we will either match you with a peer who is at a similar conceptual point in the course, or, if a section leader is available you will get to talk to them. Once you sign up, it may take up to 5 mins to find a good pairing. Once you are matched, we will put you into a virtual session with tools for you to collaborate. You should expect to spend around 15 minutes in the session.

Past Connections

Collaborators	Date	Links
Ali Raza Malik	10/28/2022, 1:09:00 PM	View session work
Ali Raza Malik	10/28/2022, 1:09:00 AM	View session work



Peer Learning in CS109

The queue is currently closed. The next peer learning session is starting in

02 secs

When you hit the button we will either match you with a peer who is at a similar conceptual point in the course, or, if a section leader is available you will get to talk to them. Once you sign up, it may take up to 5 mins to find a good pairing. Once you are matched, we will put you into a virtual session with tools for you to collaborate. You should expect to spend around 15 minutes in the session.

Past Connections

Collaborators	Date	Links
Ali Raza Malik	10/28/2022, 1:09:00 PM	View session work
Ali Raza Malik	10/28/2022, 1:09:00 AM	View session work
Ali Raza Malik	10/28/2022, 1:05:00 AM	View session work
Ali Raza Malik	10/28/2022, 1:01:00 AM	View session work
Ali Raza Malik	10/27/2022, 8:58:46 PM	View session work

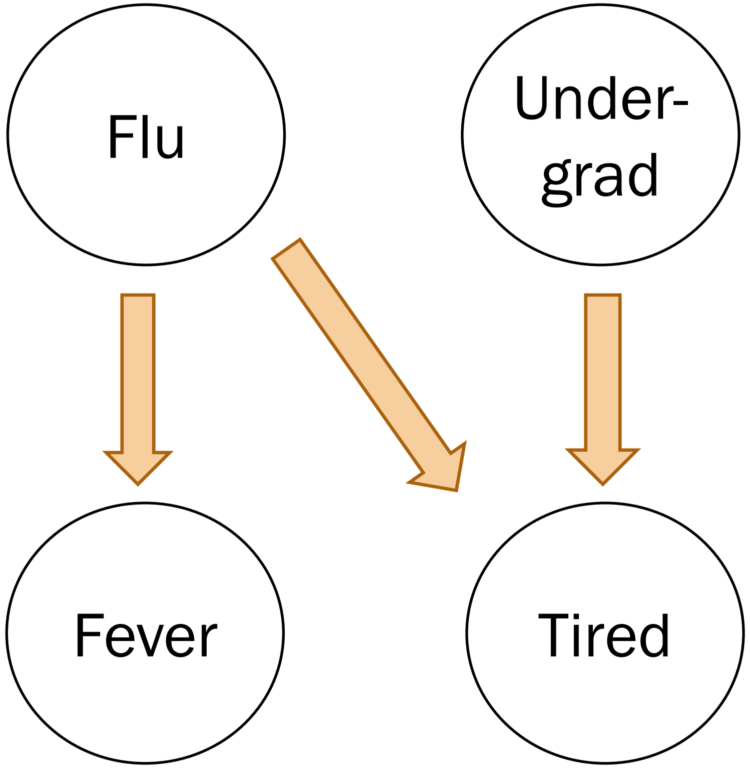
Learning Goals

1. Finish conversation on correlations
2. Learn rejection sampling

BAYES NETS!

Where do models come from?

Constructing a Bayesian Network



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

In a Bayesian Network,

Each random variable is caused by its parents. Def P(node | parents)

- Node: random variable
- Directed edge: causality

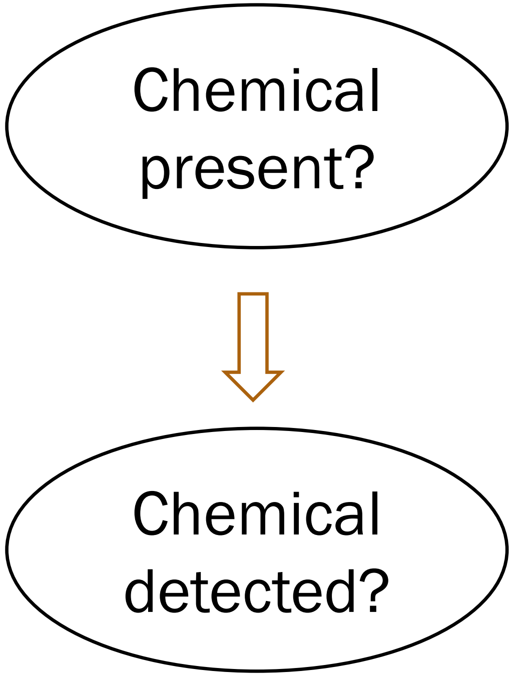
Examples:

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

Make a *Generative* Model

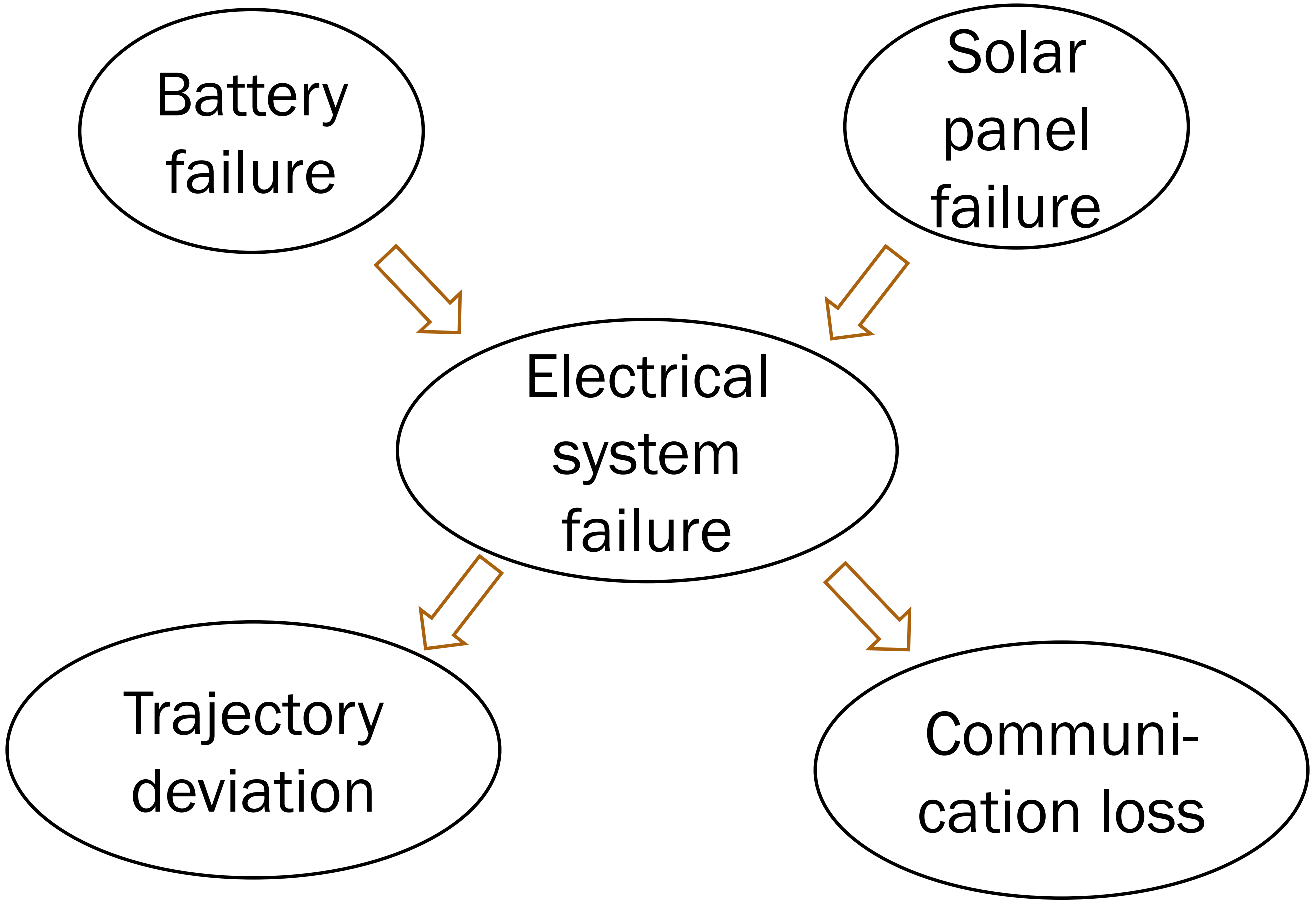
A good probabilistic model is generative. It explains the process through which the joint is created.

Other applications



Chemical present?

Chemical detected?



Battery failure

Solar panel failure

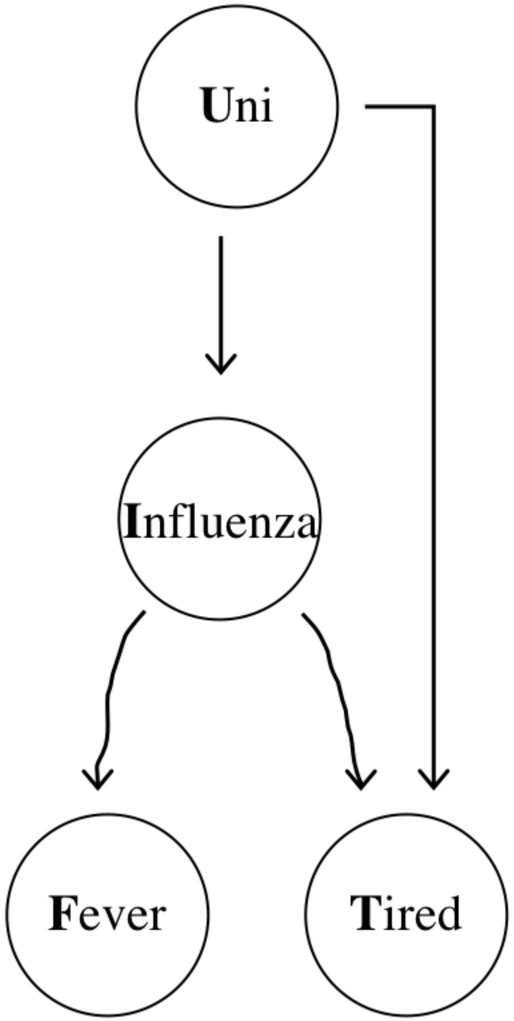
Electrical system failure

Trajectory deviation

Communi-cation loss

Bayesian Network

Simple Disease Model



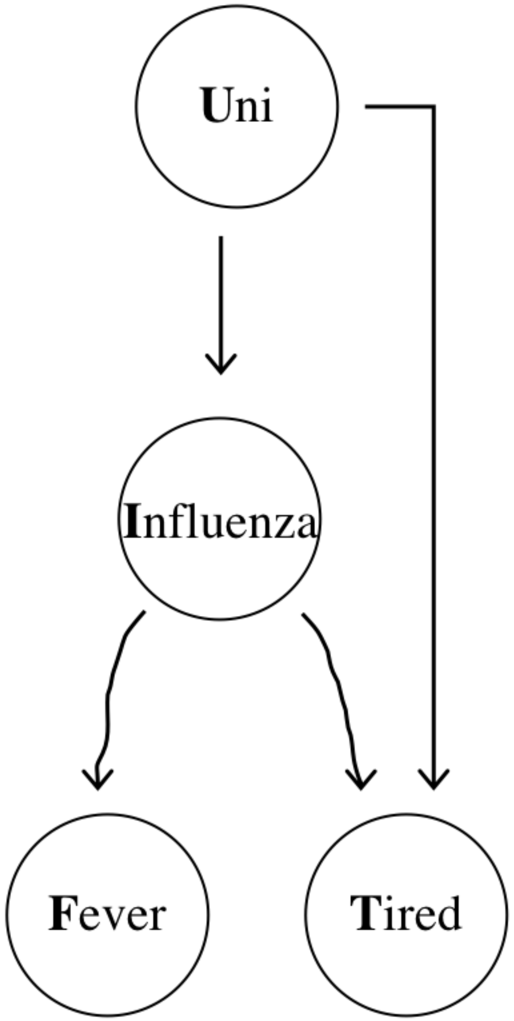
```
def get_prob_Xi(x, parents):
    # what is the probability that Xi = x
    # given the list parents of assignments to
    # the parents variables Xi
```

$$P(\text{Uni} = 1) = 0.8$$
$$P(\text{Influenza} = 1|\text{Uni} = 1) = 0.2$$
$$P(\text{Influenza} = 1|\text{Uni} = 0) = 0.1$$
$$P(\text{Tired} = 1|\text{Uni} = 0, \text{Influenza} = 0) = 0.1$$
$$P(\text{Tired} = 1|\text{Uni} = 1, \text{Influenza} = 0) = 0.8$$

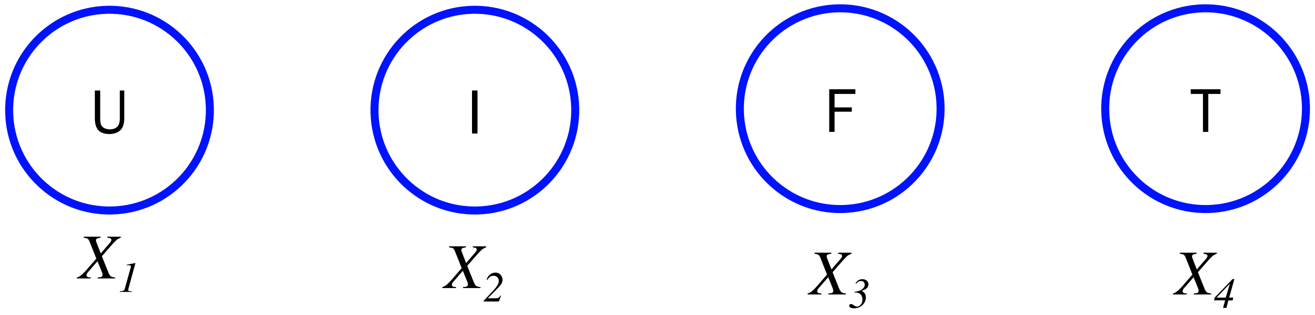
$$P(\text{Fever} = 1|\text{Influenza} = 1) = 0.9$$
$$P(\text{Fever} = 1|\text{Influenza} = 0) = 0.05$$
$$P(\text{Tired} = 1|\text{Uni} = 0, \text{Influenza} = 1) = 0.9$$
$$P(\text{Tired} = 1|\text{Uni} = 1, \text{Influenza} = 1) = 1.0$$

Bayesian Network Assumption

Simple Disease Model



Order nodes by ancestry



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

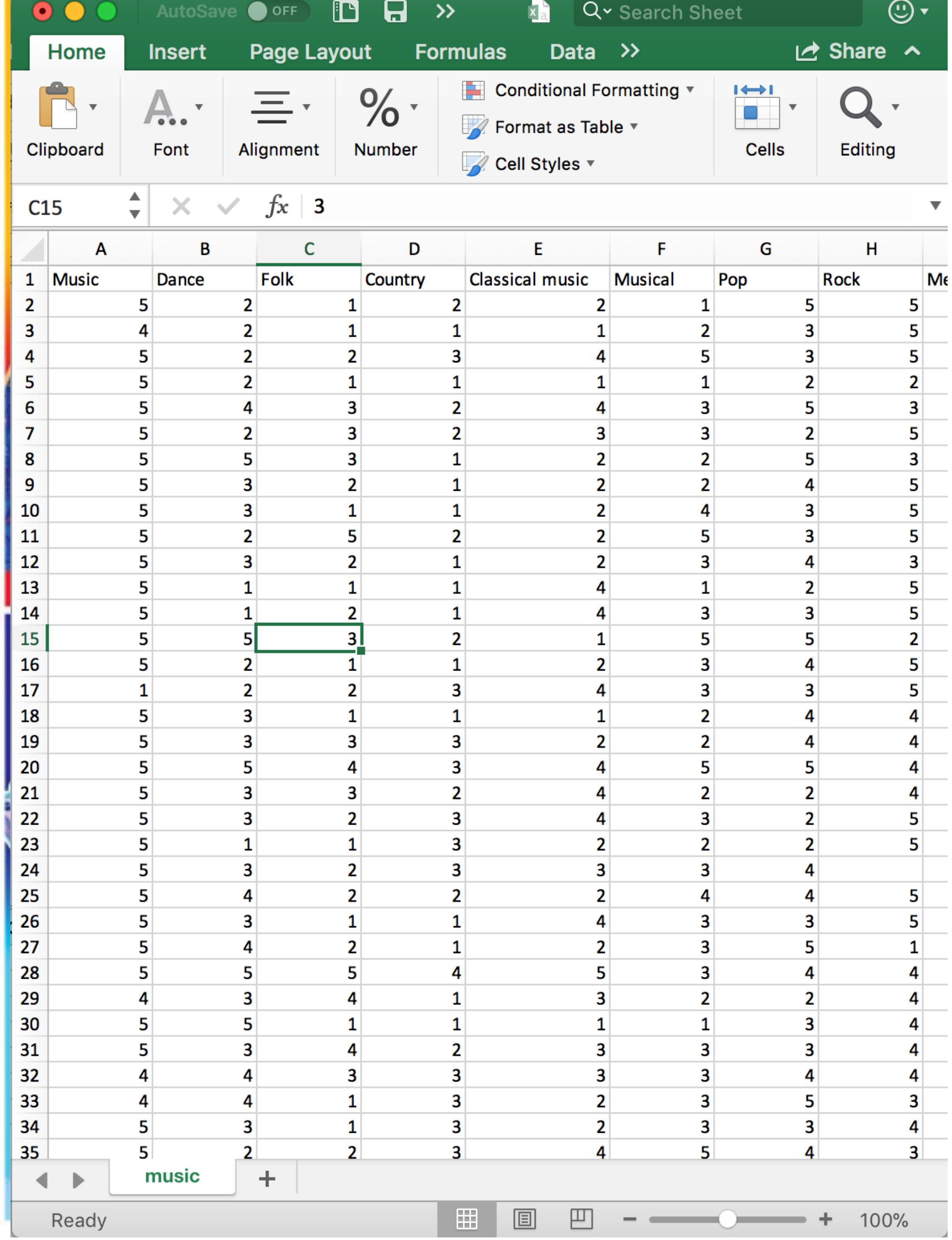
$$= \prod_i P(x_i | \text{Values of parents of } X_i)$$

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

End Review

How do people design these?

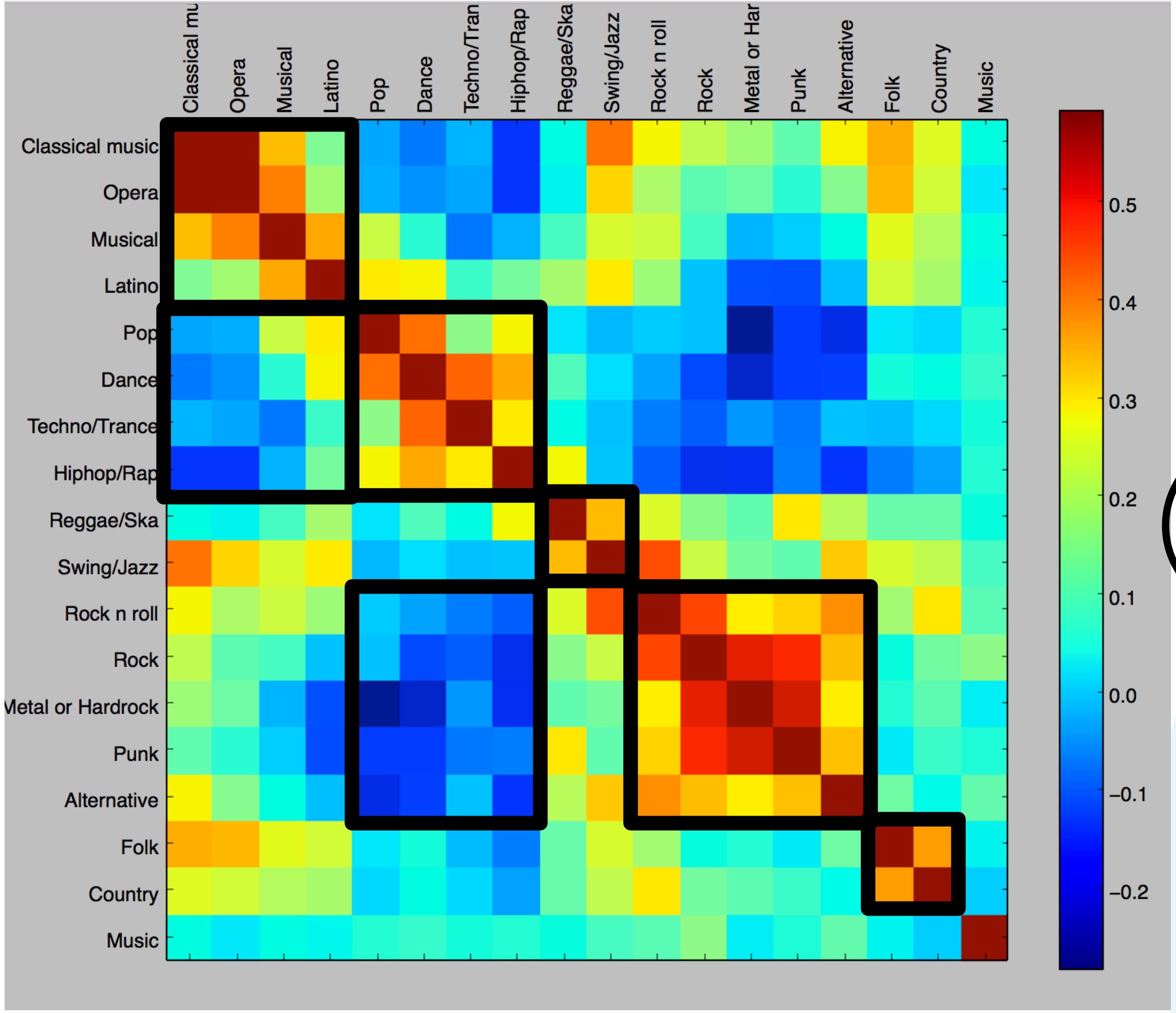
ROCK
The Sound: Vigorous, defiant, energetic, inventive
The Roots: Rhythm & blues, country
The Pioneers: Bill Haley, Chuck Berry, Fats Domino, Little Richard, Buddy Holly, Elvis Presley
The Places: Cleveland, New Orleans, Detroit, New York City
The Ensemble: Electric guitar, bass, drums, keyboard, vocals
"We're a rock group. We're noisy, rowdy, sensational and weird."
– Angus Young (b. 1955)
Lead guitarist of rock band AC/DC
HIP-HOP RAP
The Sound: Rhythmic, unvarnished, adaptable, streetwise
The Roots: Rhythm & blues, soul, funk, reggae
The Pioneers: Afrika Bambaataa, Kool Herc, DJ Hollywood, Grandmaster Flash, Kurtis Blow, Grandmaster Caz
The Places: New York City (South Bronx)
The Ensemble: Vinyl, turntable, vocals
"The beautiful thing about hip-hop is it's like 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."
– Talib Kweli (b. 1975)
Hip-hop artist
The Sound: Syncopated, enthusiastic, diverse, vibrant
The Roots: Spain, Africa, Caribbean, South America
The Pioneers: Arsenio Rodríguez, 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
LATIN American
"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."
– Rubén Blades (b. 1948)
Salsa singer and songwriter
Folk
The Sound: Grassroots, narrative, sincere, lyrical
The Roots: Ballads, immigrant folklore, spirituals, cowboy songs
The Pioneers: Lead Belly, Odetta, Woody Guthrie, Pete Seeger, Bob Dylan, Joan Baez
The Places: Appalachia, Deep South, Western frontier
The Ensemble: Guitar, banjo, fiddle, accordion, vocals
"I liked the rhythms [of folk music]. I liked the melodies, time-tested by generations of singers. Above all, I liked the words . . . they seemed frank, straightforward, honest."
– Pete Seeger (b. 1919)
Folk musician
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 Western
"Country music is three chords and the truth."
– Harlan Howard (1927–2002)
Country music songwriter
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

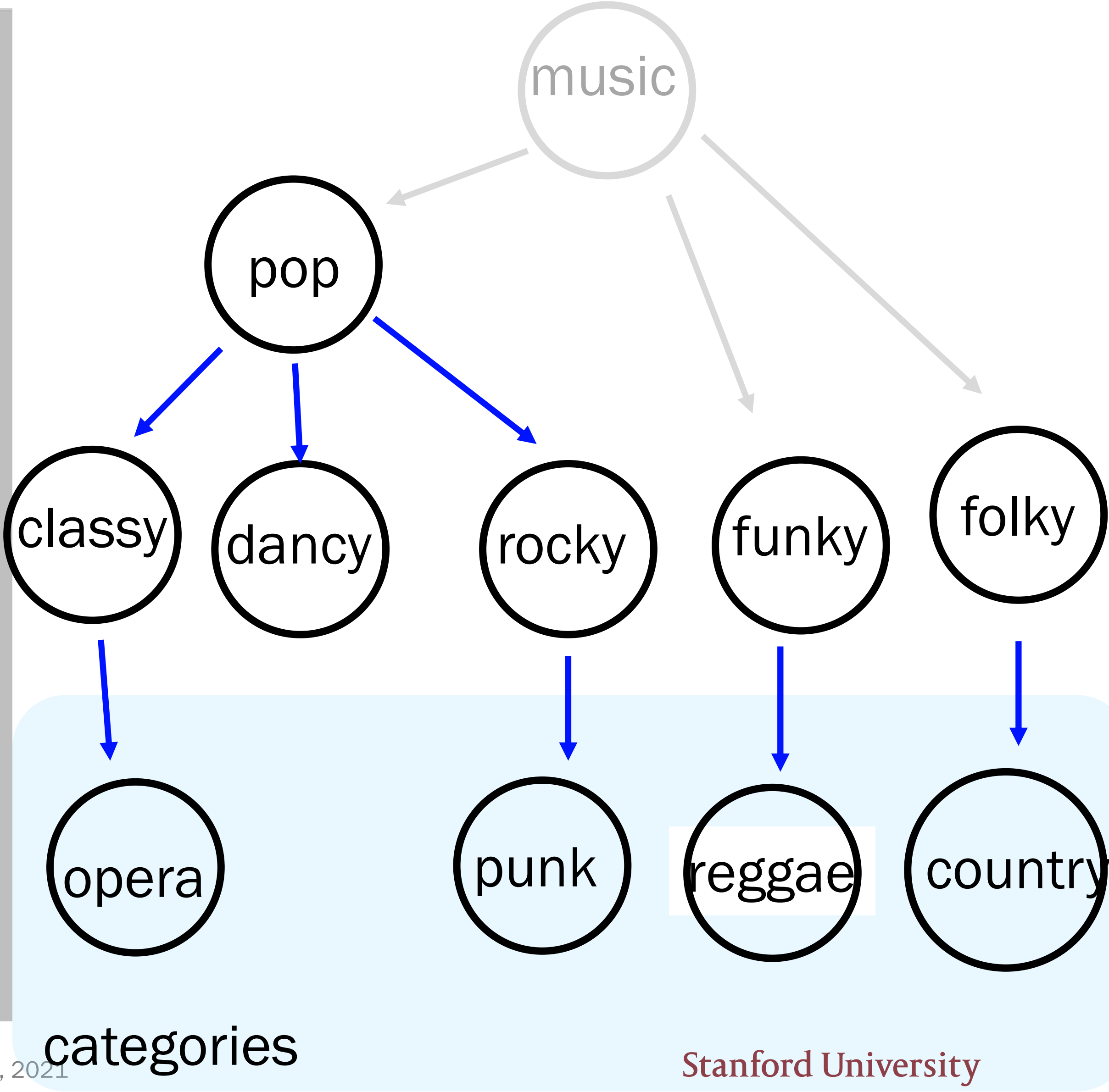


C15 | fx | 3

	A	B	C	D	E	F	G	H	
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	

From Correlation to Bayes Net. Alternative!





music

pop

classy

dancy

rocky

funky

folky

opera

punk

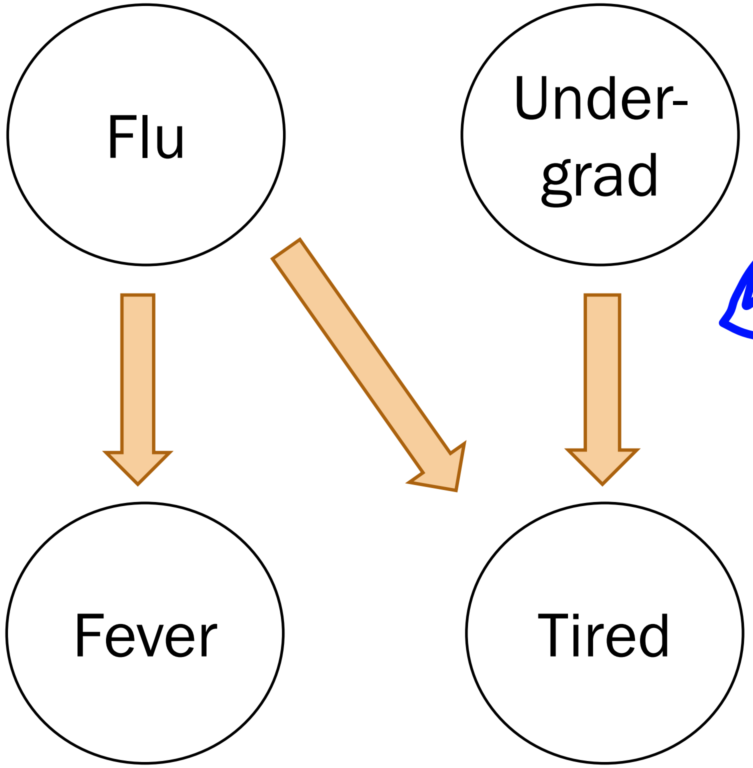
reggae

country

categories

The art of modelling

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



1. Design this

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

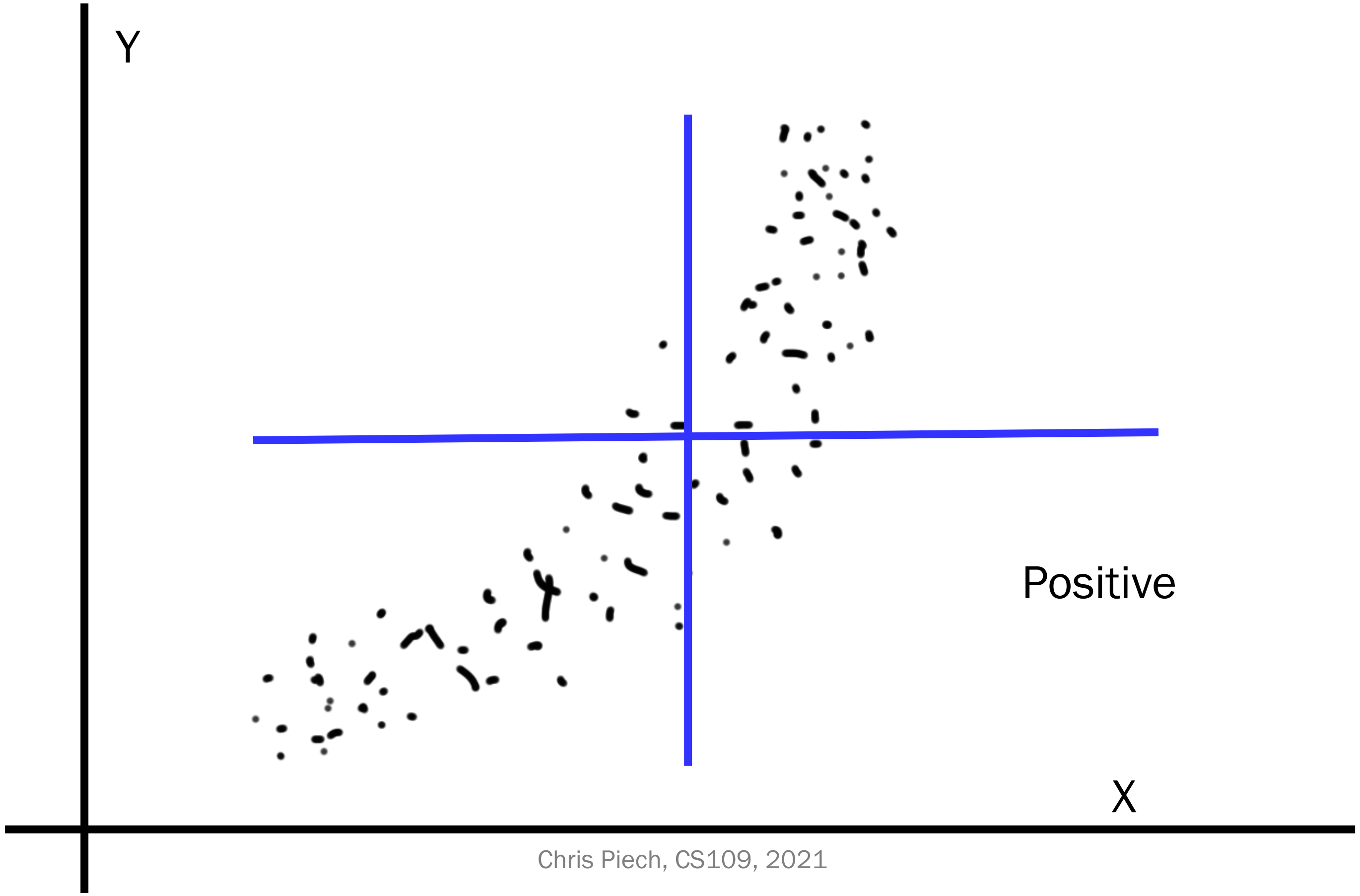
$$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$$
$$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$$

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

Calculate the Covariance / Correlation (new stat!)

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

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



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 \ 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

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

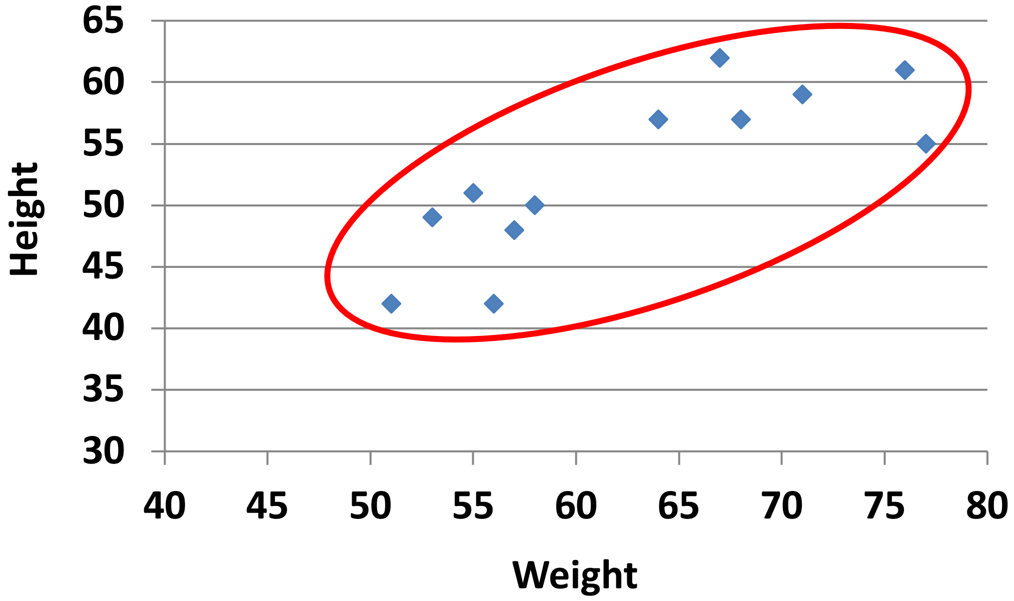
- E[X] = -1(1/3) + 0(1/3) + 1(1/3) = 0
- E[Y] = 0(2/3) + 1(1/3) = 1/3
- Since XY = 0, E[XY] = 0
- Cov(X, Y) = E[XY] – E[X]E[Y] = 0 – 0 = 0

But, X and Y are clearly dependent!

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

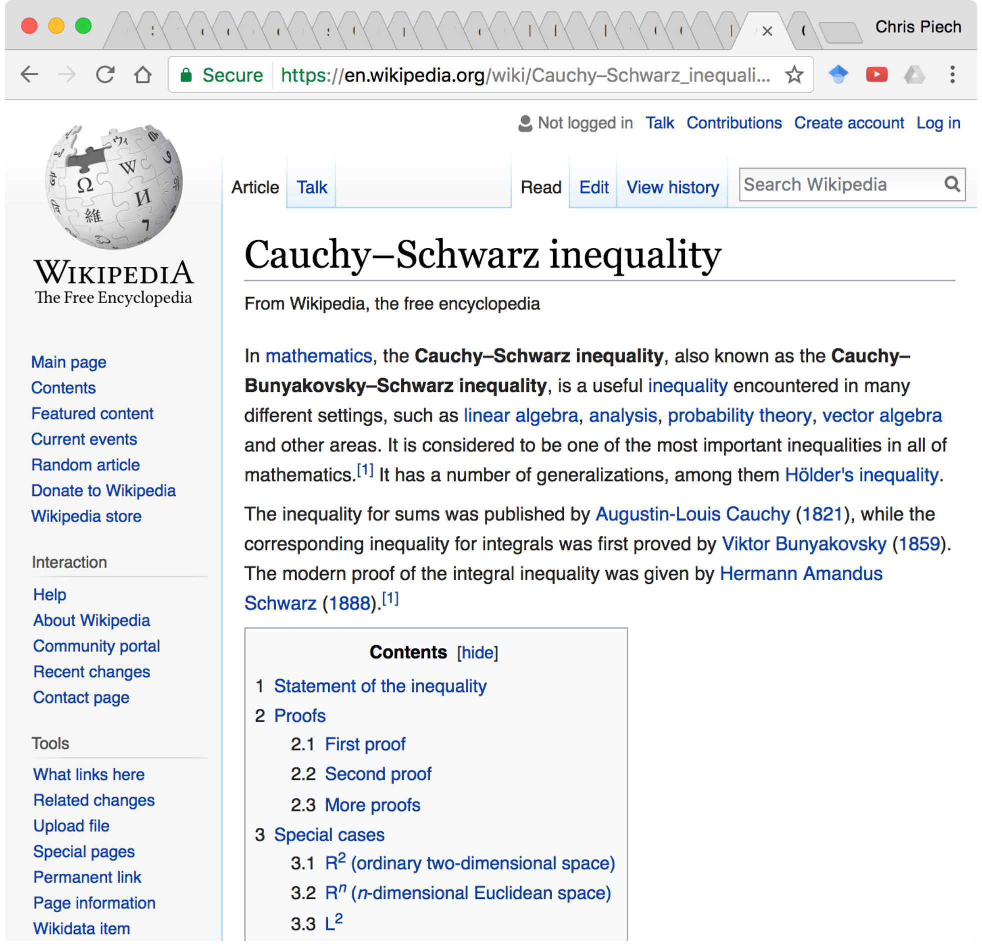


Cov(W, H) = E[W*H] – E[W]E[H]

= 3355.83 – (62.75)(52.75)

= 45.77

Cauchy Schwarz, a great way to normalize!



Cauchy–Schwarz inequality

From Wikipedia, the free encyclopedia

In mathematics, the **Cauchy–Schwarz inequality**, also known as the **Cauchy–Bunyakovsky–Schwarz inequality**, is a useful inequality encountered in many different settings, such as linear algebra, analysis, probability theory, vector algebra and other areas. It is considered to be one of the most important inequalities in all of mathematics.[1] It has a number of generalizations, among them Hölder's inequality.

The inequality for sums was published by Augustin-Louis Cauchy (1821), while the corresponding inequality for integrals was first proved by Viktor Bunyakovsky (1859). The modern proof of the integral inequality was given by Hermann Amandus Schwarz (1888).[1]

Contents [hide]

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

Correlation is just normalized Covariance

Correlation

Covariance

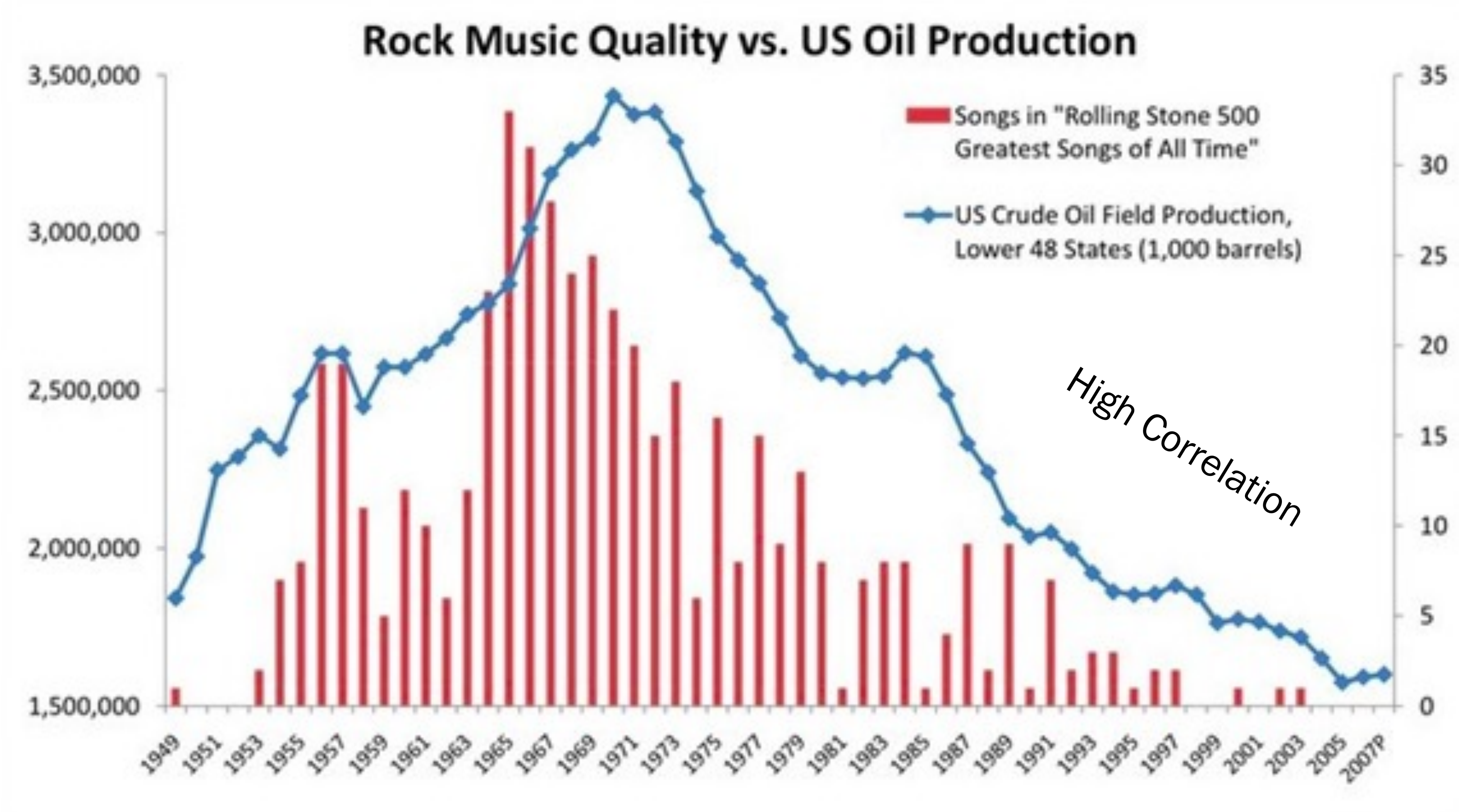
$$\rho(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

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?

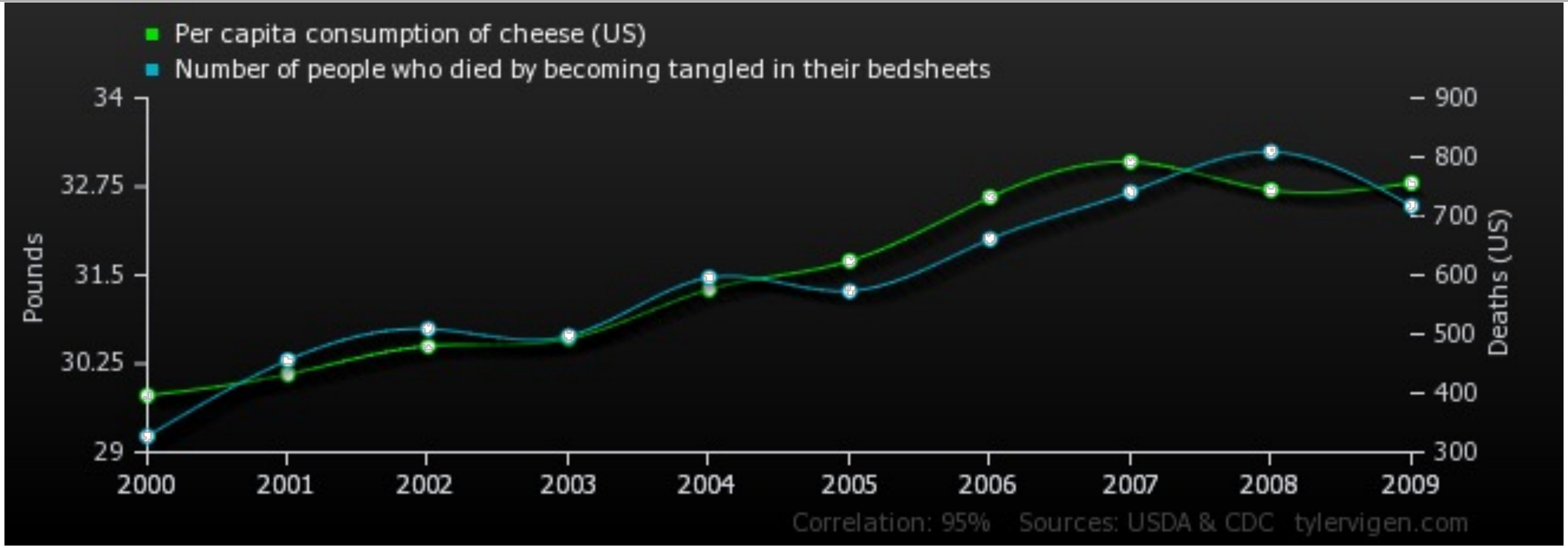


High Correlation

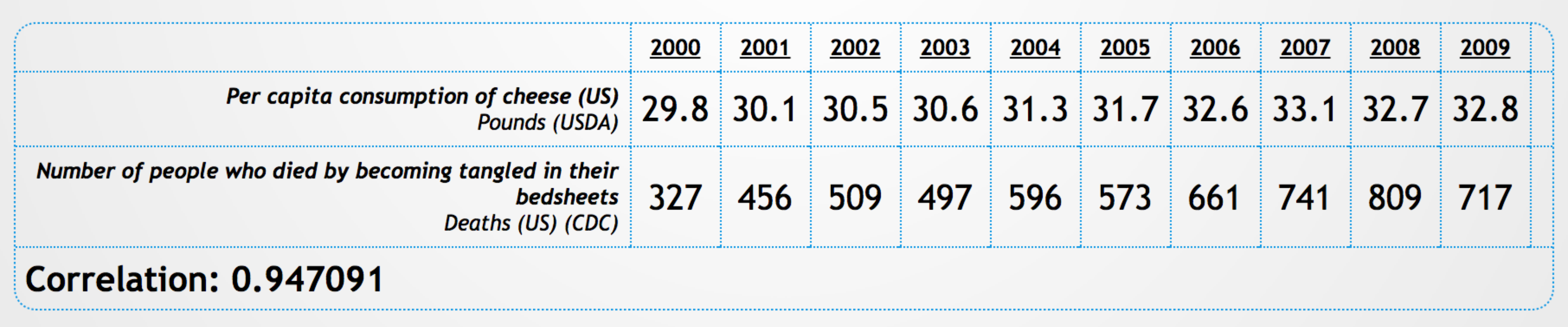
Hubbert Peak Theory

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

Tell your friends!



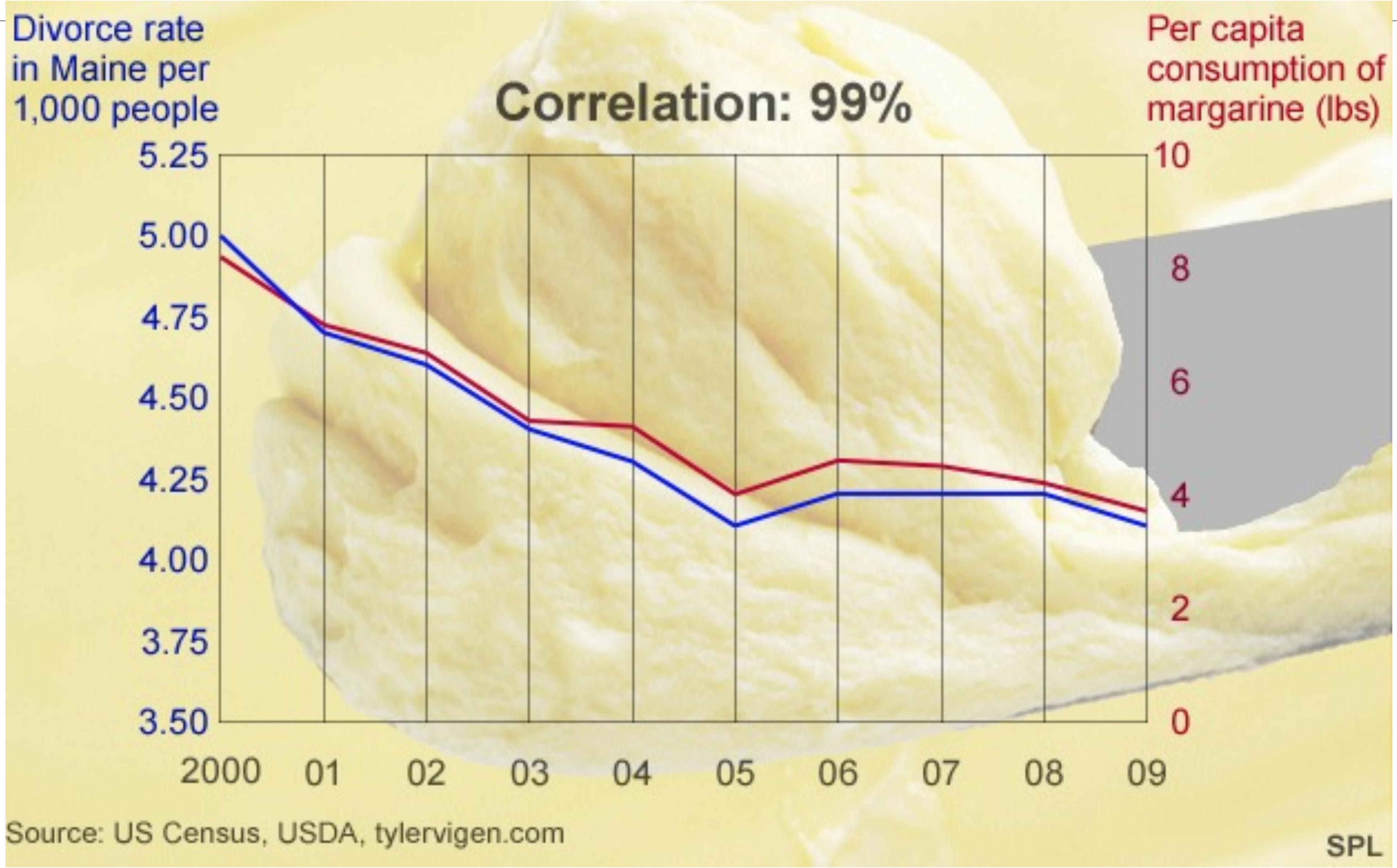
Correlation: 95% Sources: USDA & CDC tylervigen.com



	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

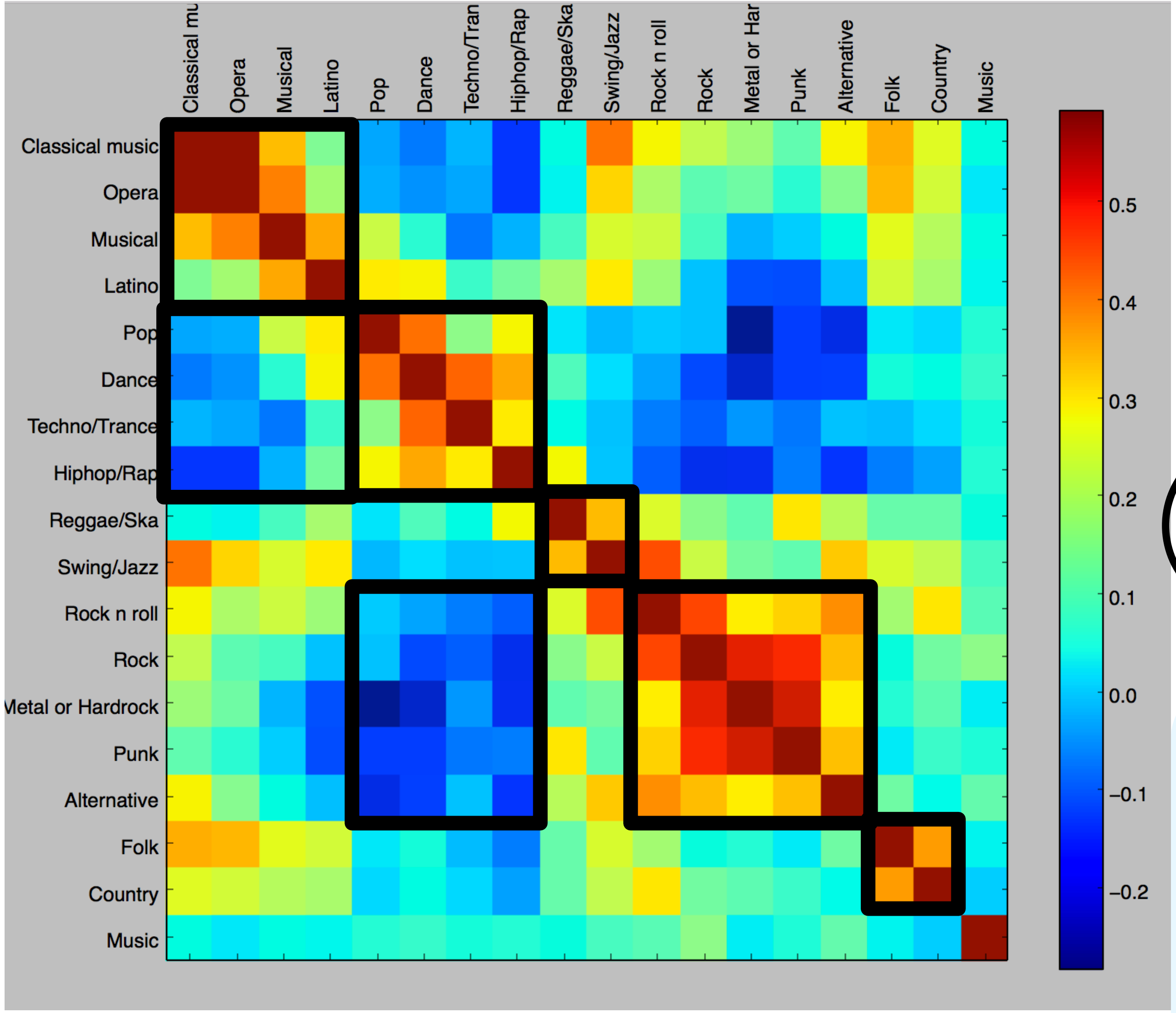
Divorce Vs Butter?

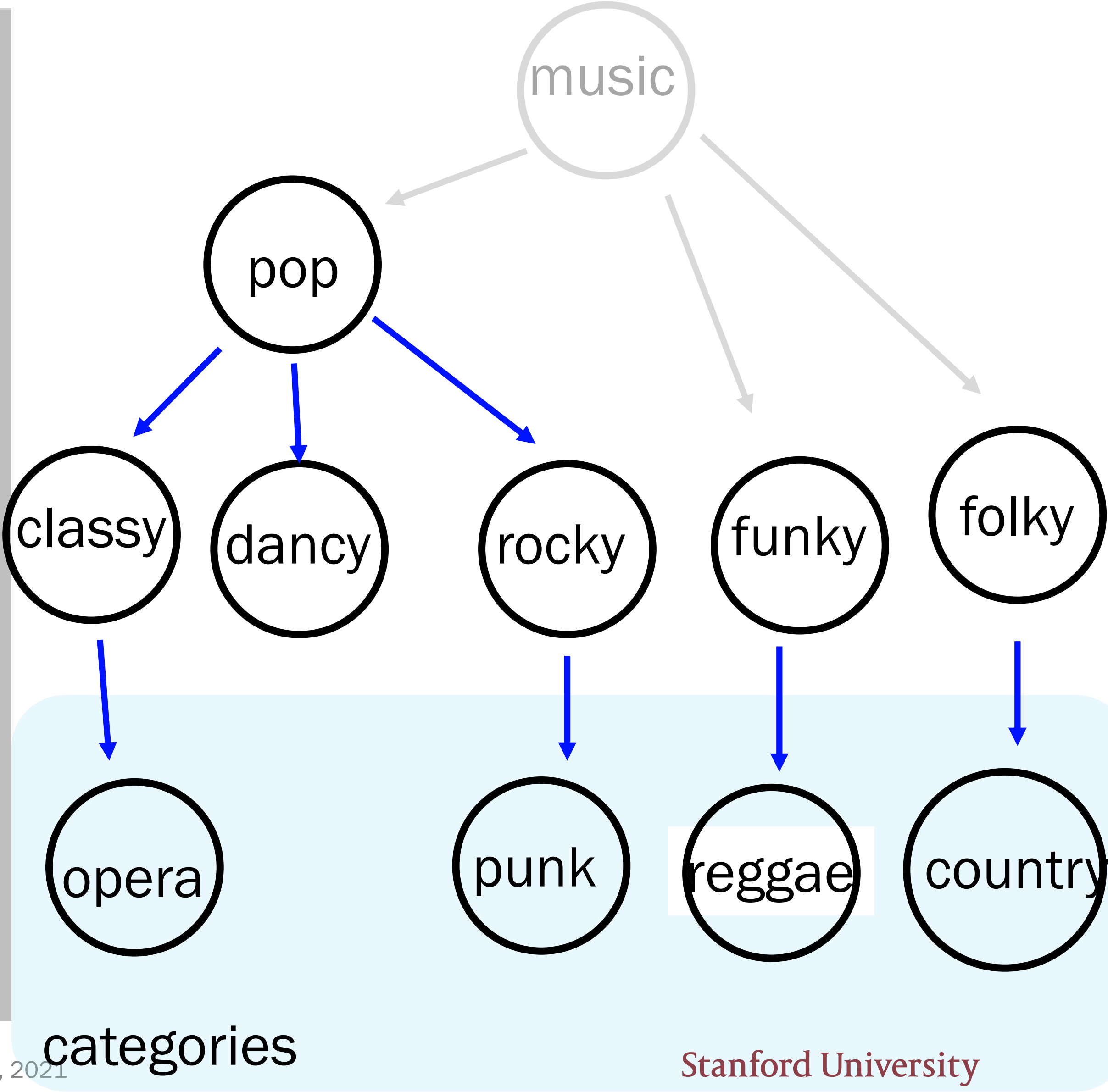


Source: US Census, USDA, tylervigen.com

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

Recall: It is a useful starting point

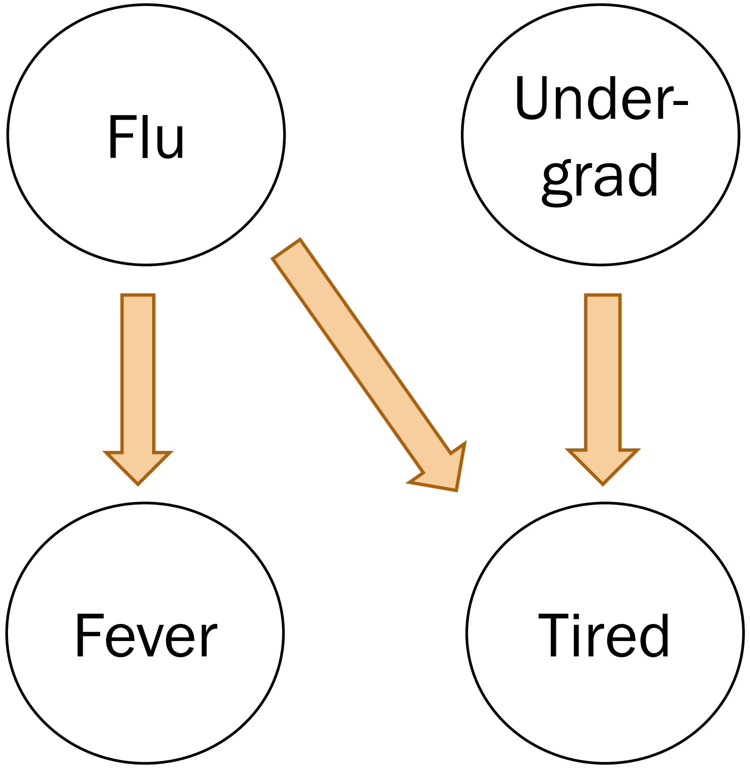




We have models. Need to solve problems

Inference: Algebra

Bayes Nets: Conditional independence



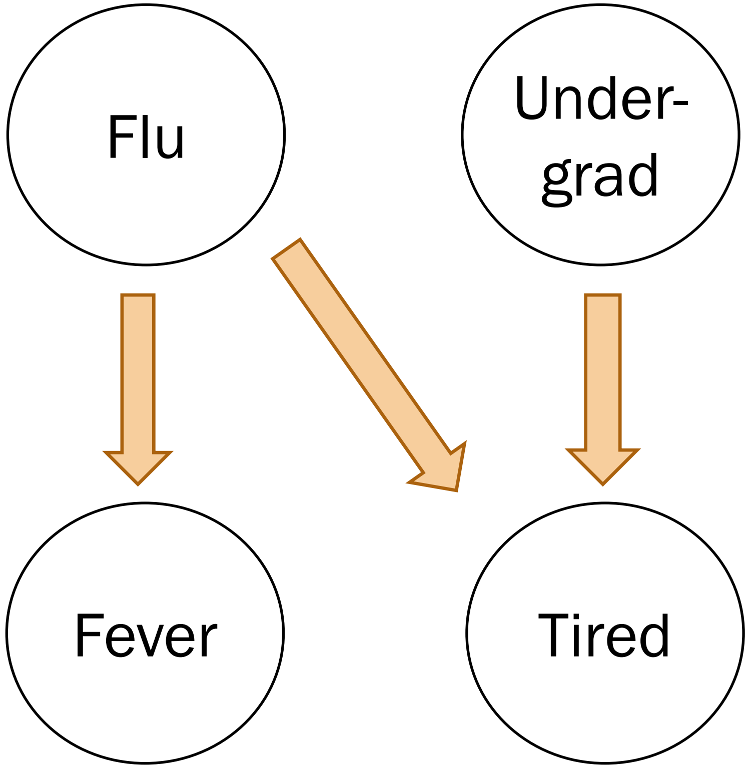
In a Bayesian Network,

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

- Node: random variable
- Directed edge: conditional dependency

Inference via math

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



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

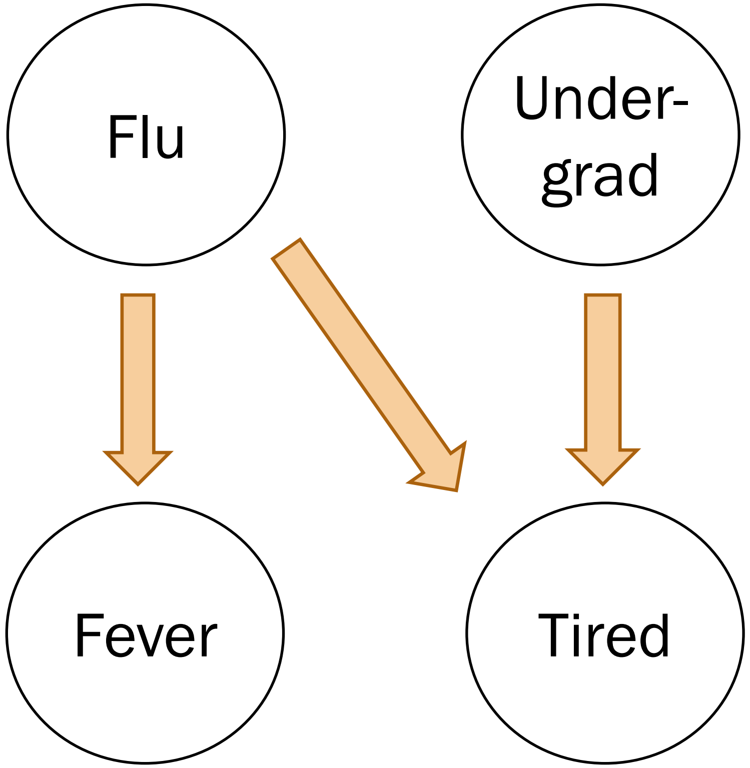
$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

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

   Compute joint probabilities using chain rule.

Inference via math

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



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

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

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

   1. Compute joint probabilities

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

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

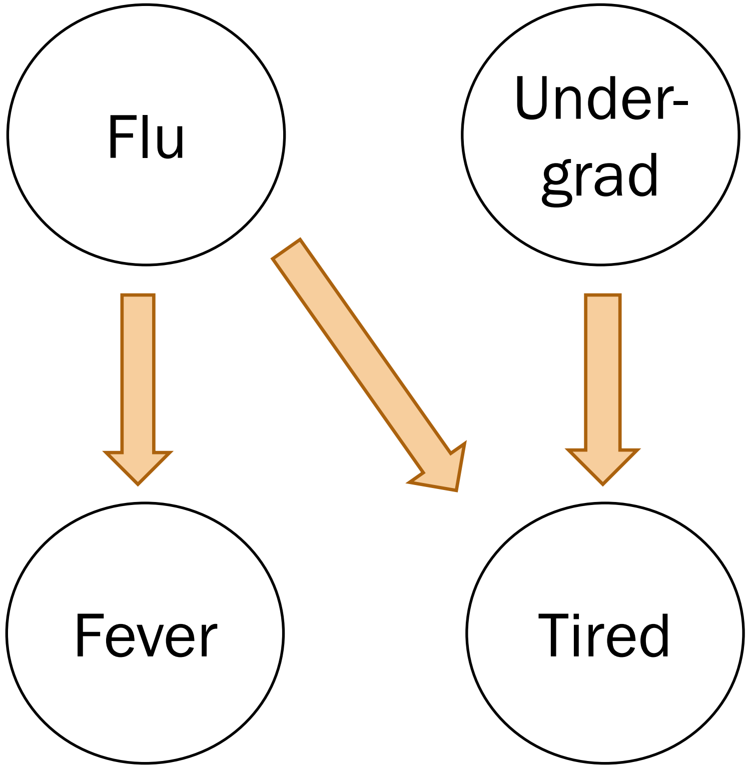
   2. Definition of conditional probability

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

      $$= 0.095$$

Inference via math

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



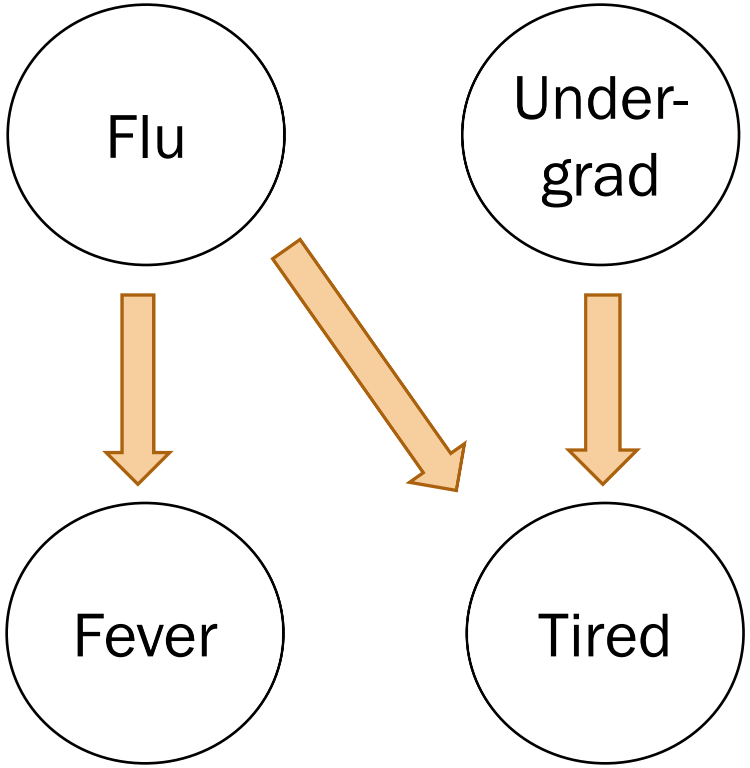
$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

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

Inference via math

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



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

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

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

   1. Compute joint probabilities

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

   2. Definition of conditional probability

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

$$= 0.122$$

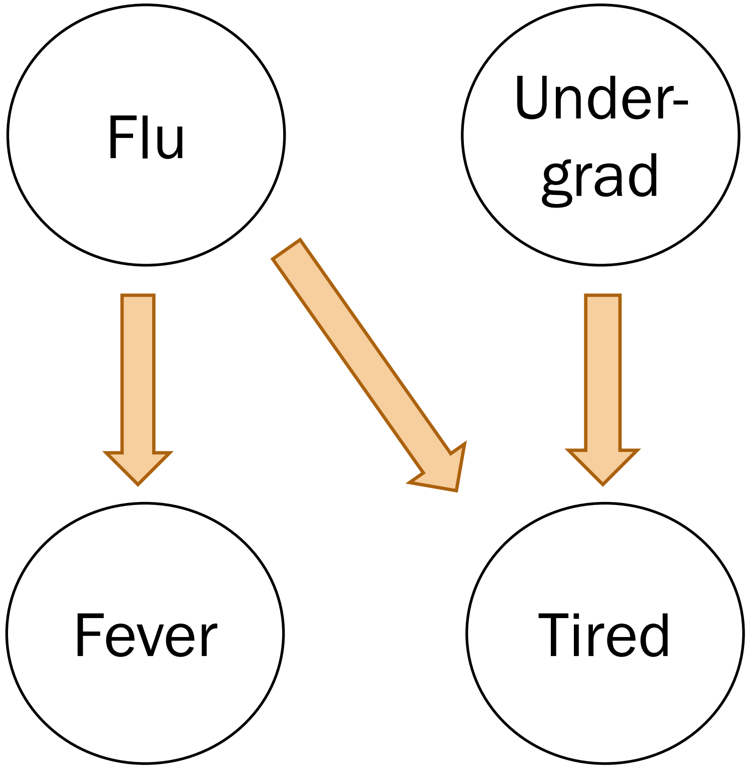
Rejection sampling algorithm

Step 0:

Have a fully specified Bayesian Network

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

$P(U = 1) = 0.8$



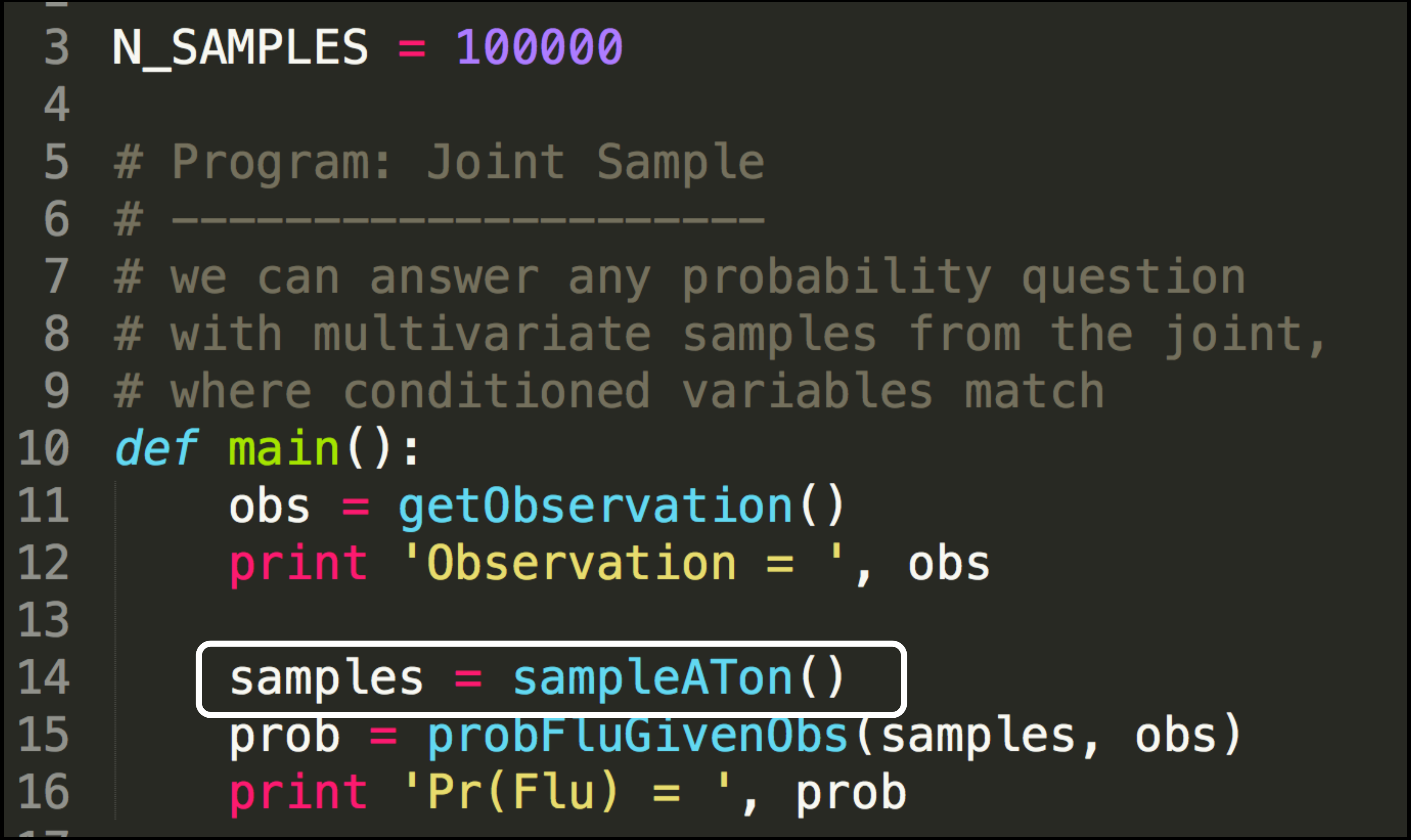
$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

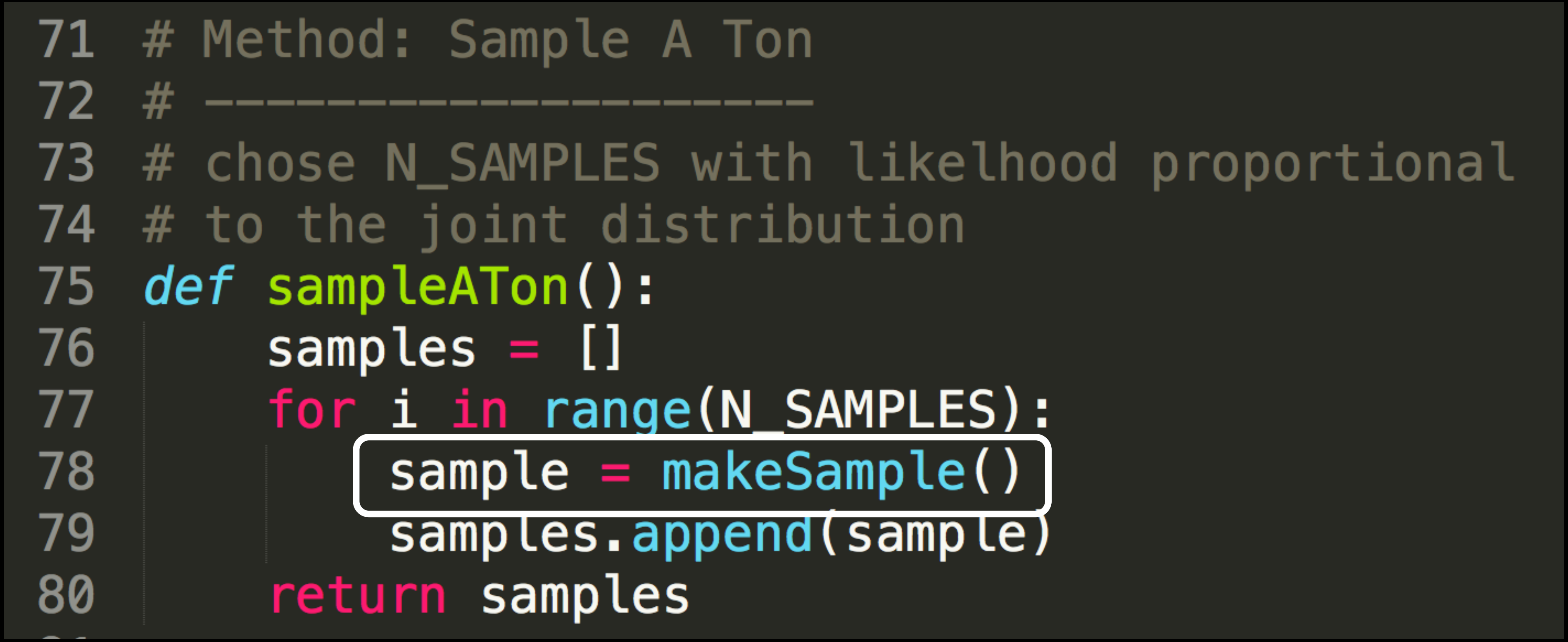
Alg #0: Straight Math

Too many possible inference questions one could ask…

Alg #1: Rejection Sampling



```
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any probability question
# with multivariate samples from the joint,
# where conditioned variables match
def main():
    obs = getObservation()
    print 'Observation = ', obs

    samples = sampleATon()
    prob = probFluGivenObs(samples, obs)
    print 'Pr(Flu) = ', prob
```

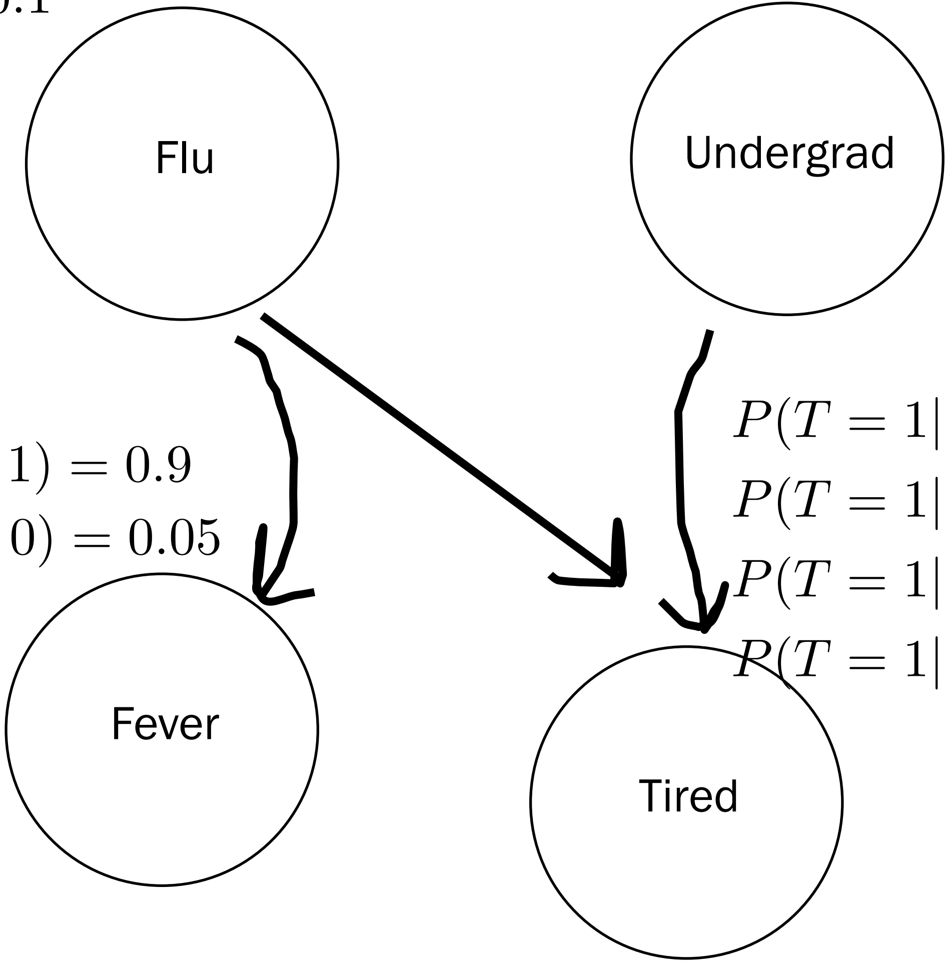


```
# Method: Sample A Ton
# --------------------
# chose N_SAMPLES with likelhood proportional
# to the joint distribution
def sampleATon():
    samples = []
    for i in range(N_SAMPLES):
        sample = makeSample()
        samples.append(sample)
    return samples
```

Recall: Probabilistic Model

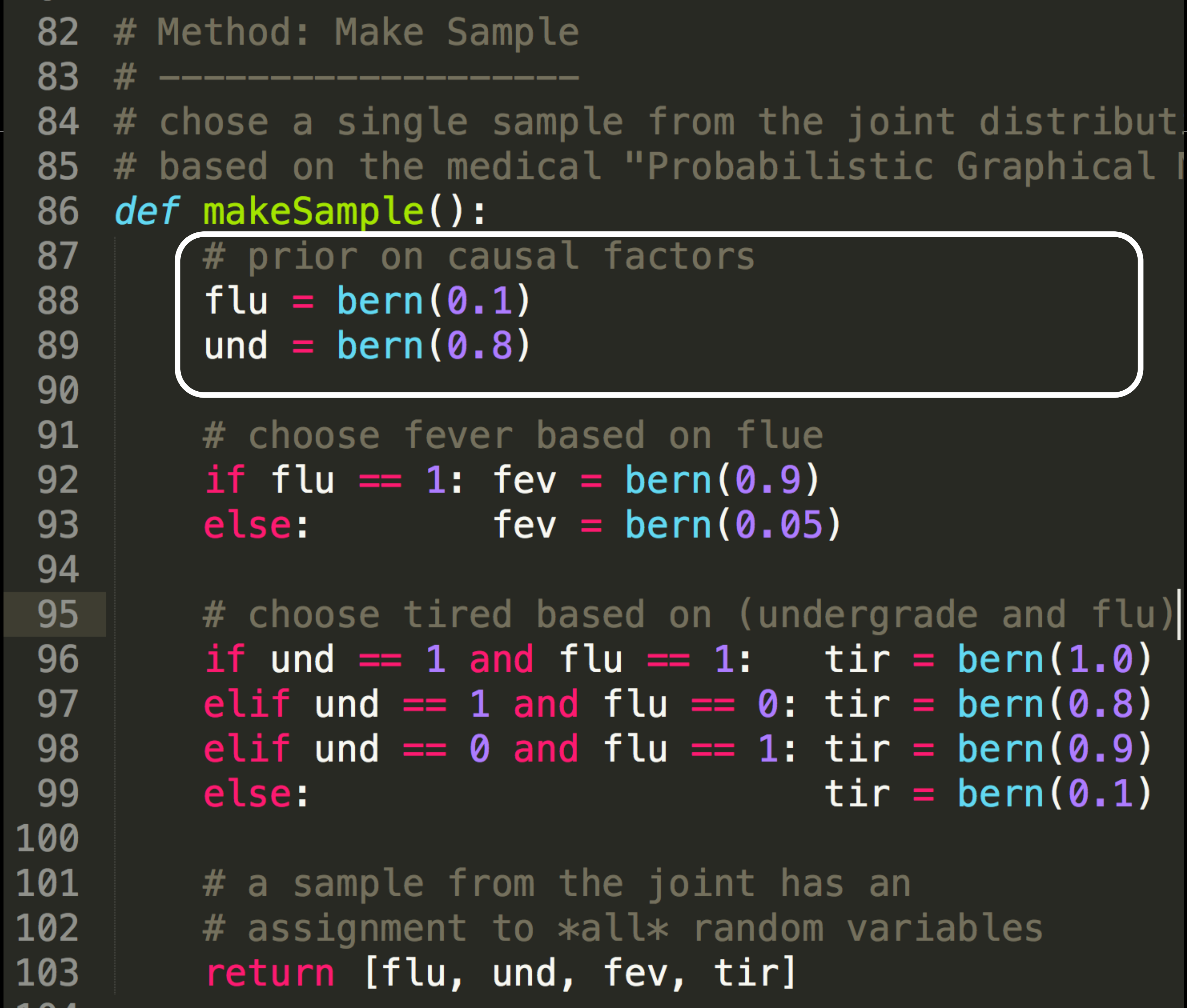
$P(Fl = 1) = 0.1$

$P(U = 1) = 0.8$

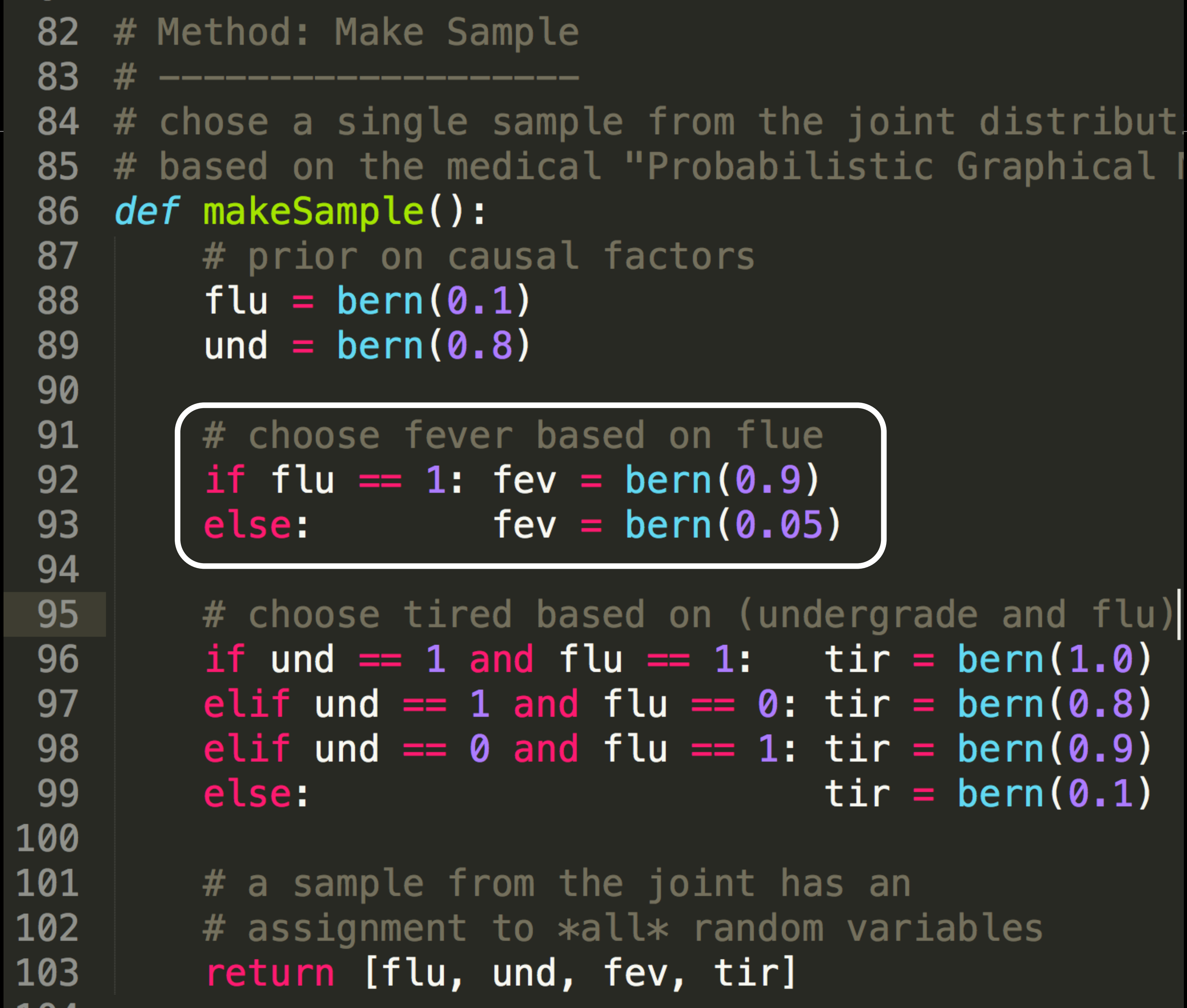


$P(Fev = 1|Flu = 1) = 0.9$
$P(Fev = 1|Flu = 0) = 0.05$

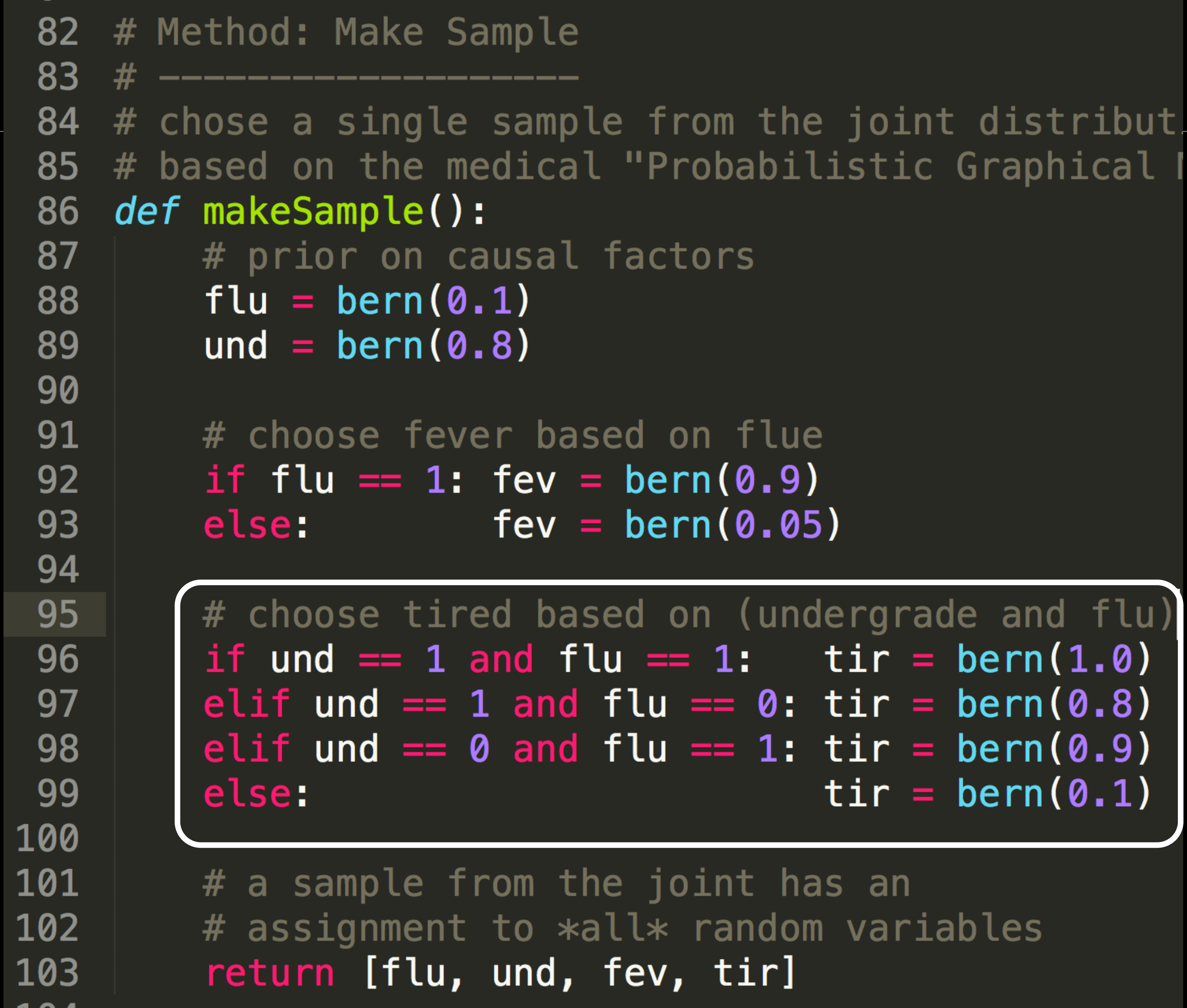
$P(T = 1|Flu = 0, U = 0) = 0.1$
$P(T = 1|Flu = 0, U = 1) = 0.8$
$P(T = 1|Flu = 1, U = 0) = 0.9$
$P(T = 1|Flu = 1, U = 1) = 1.0$



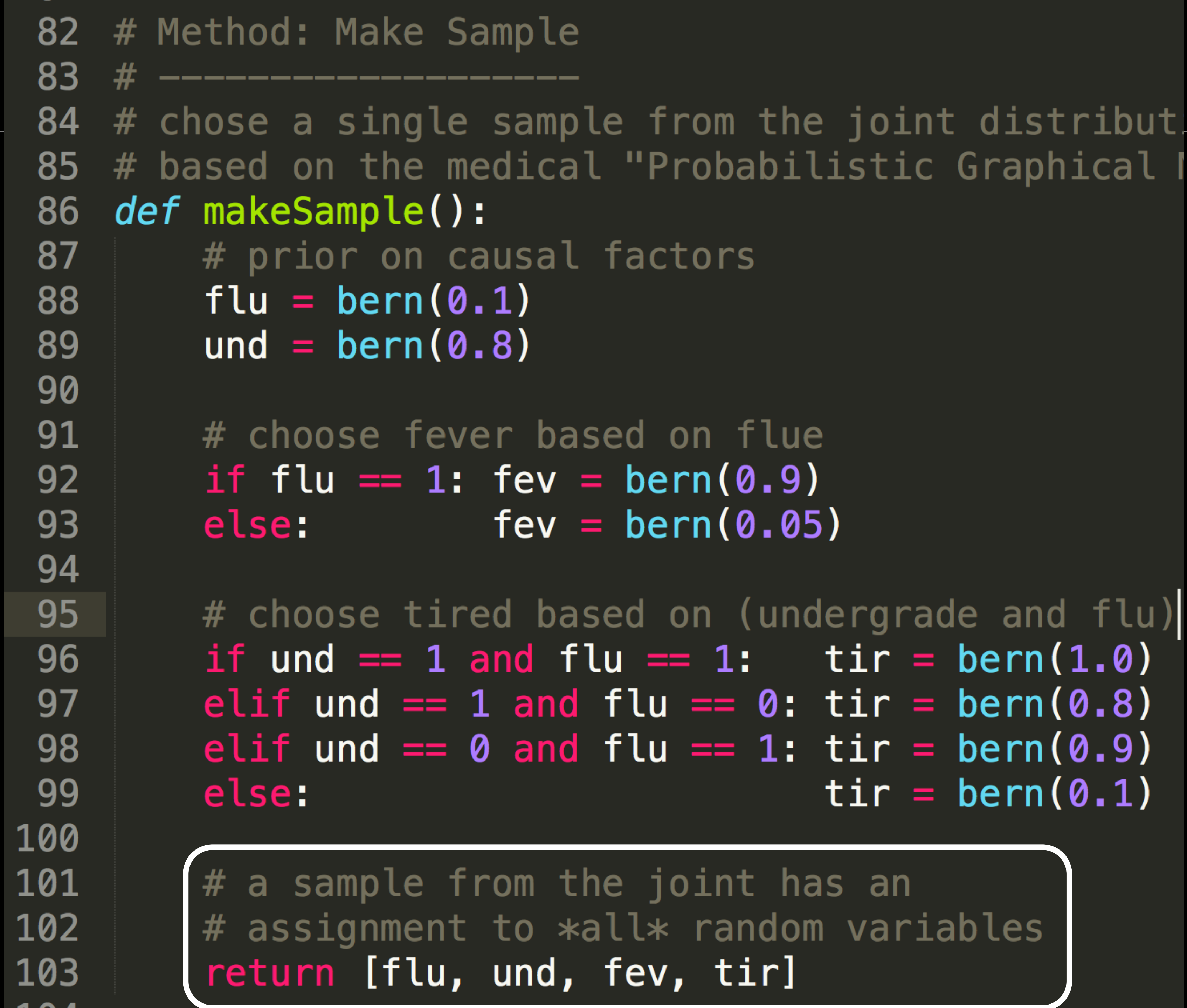
```python
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

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

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

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```



```
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

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

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

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

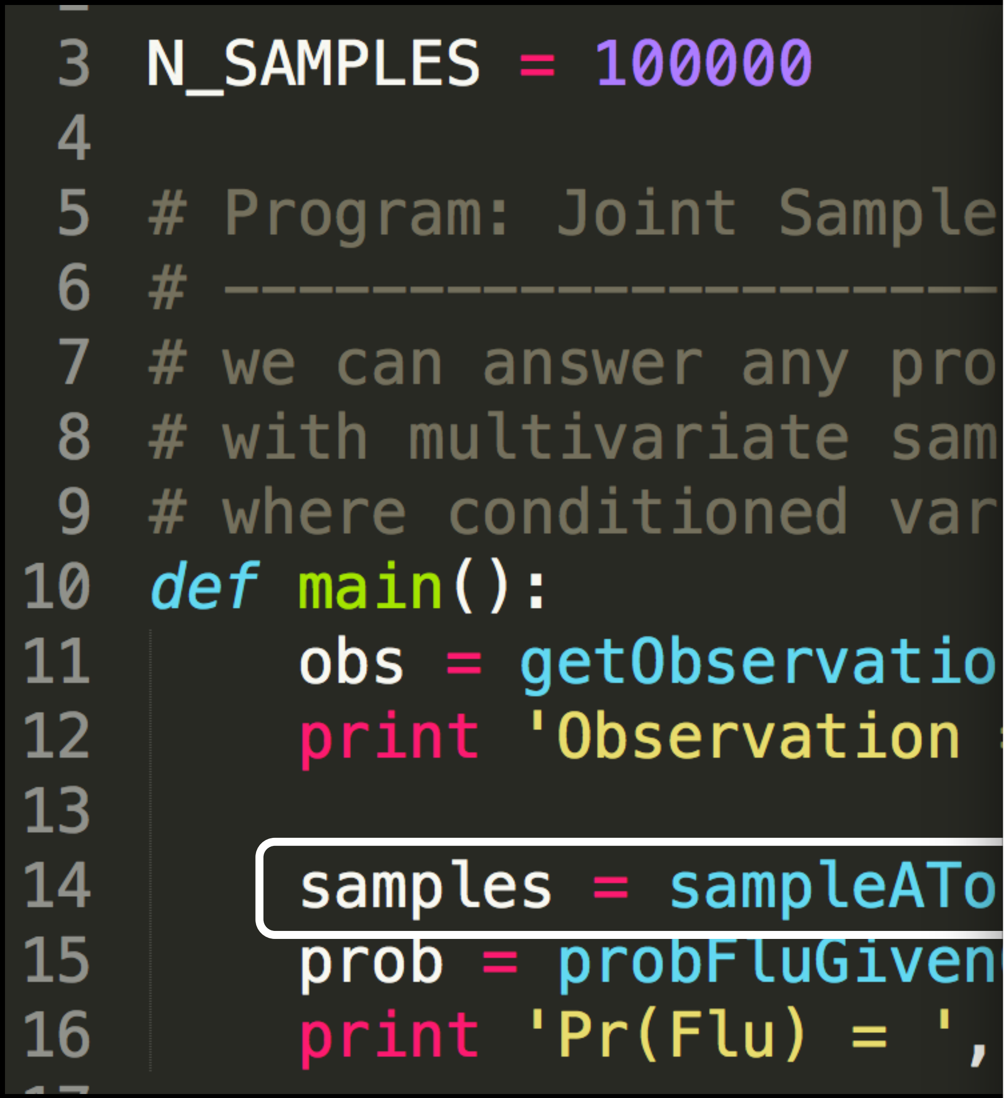


```python
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

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

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

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

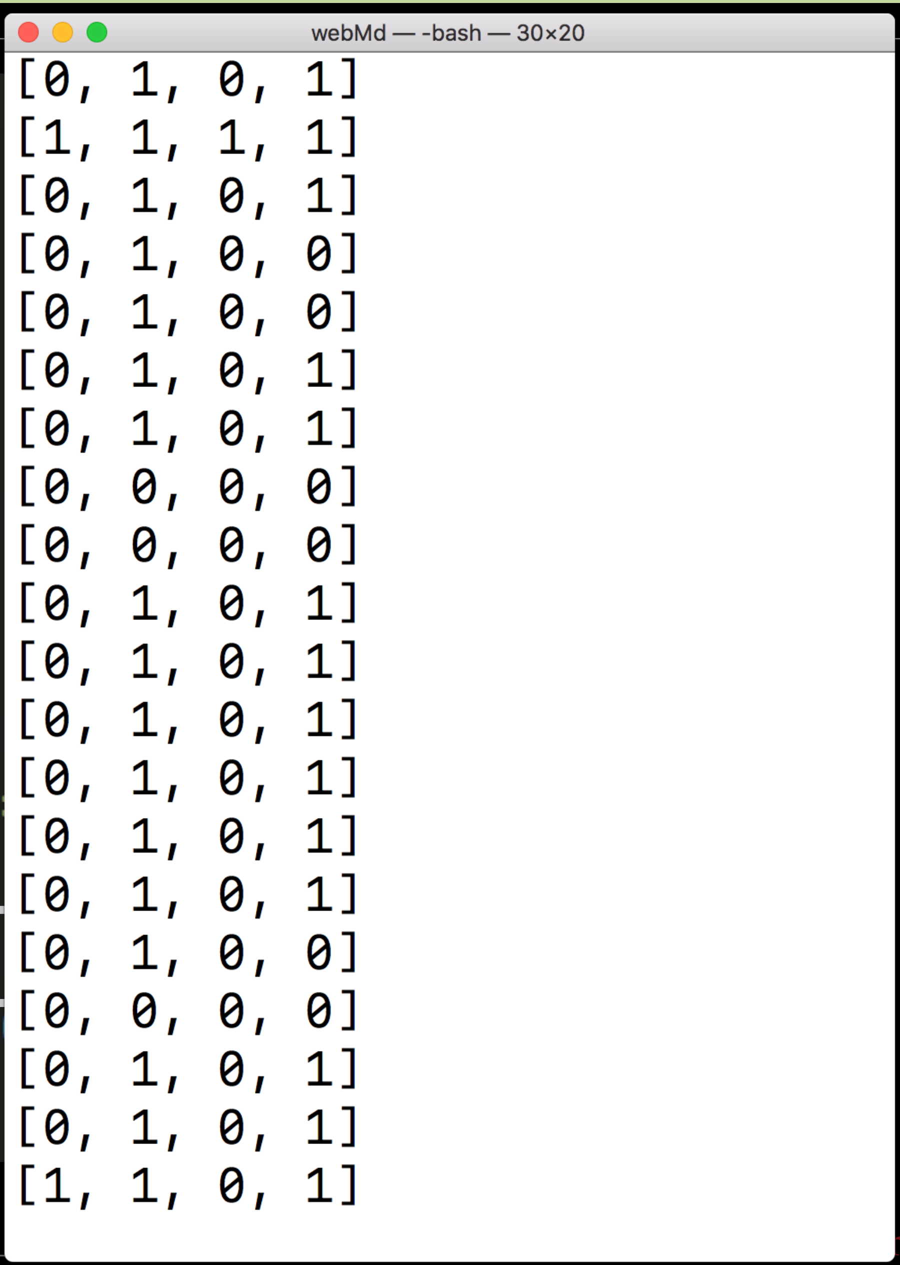


```
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

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

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

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

Alg #1: Rejection Sampling

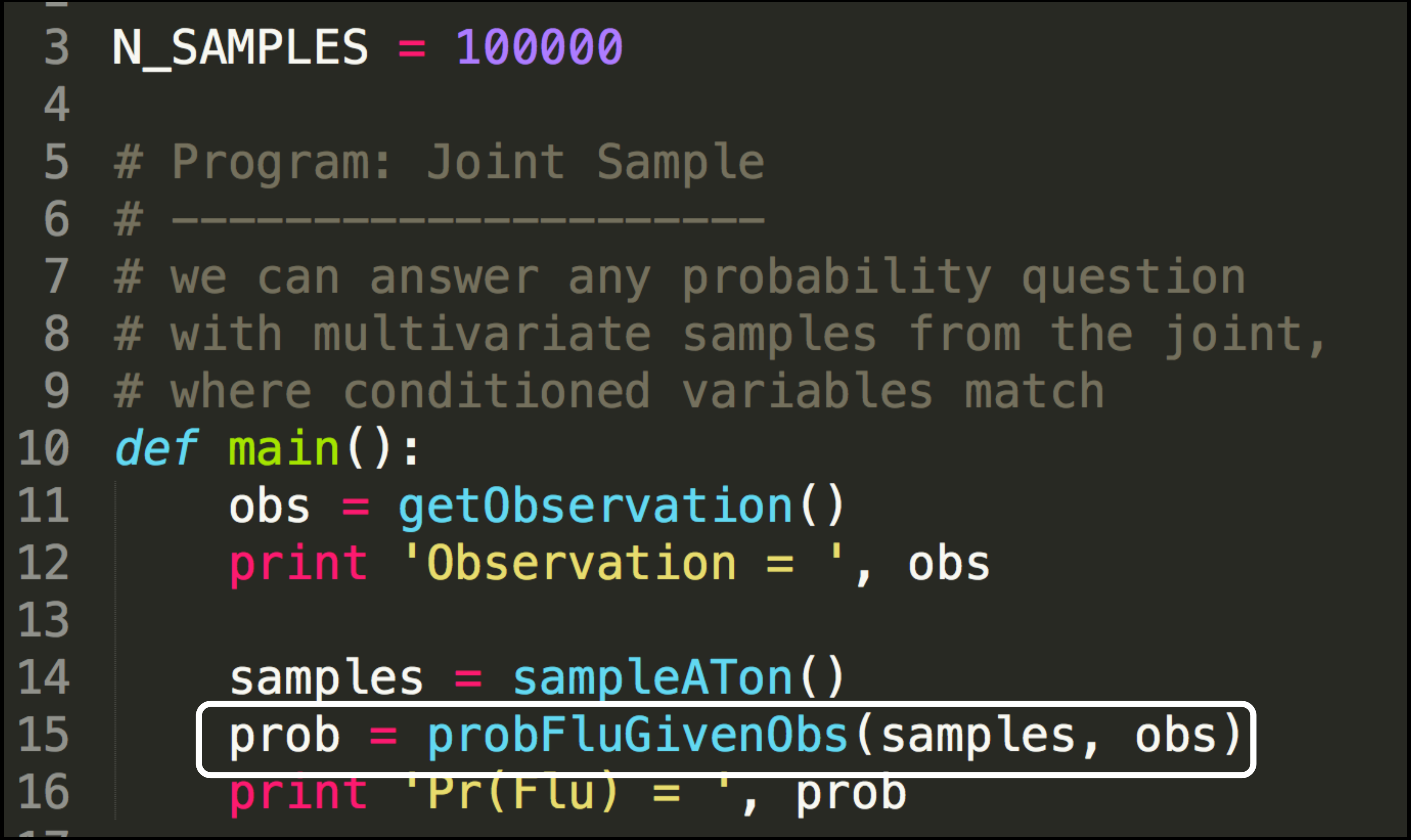


```
N_SAMPLES = 100000

# Program: Joint Sample
# ----------------------
# we can answer any pro
# with multivariate sam
# where conditioned var
def main():
    obs = getObservatio
    print 'Observation

    samples = sampleATo
    prob = probFluGiven
    print 'Pr(Flu) = ',
```

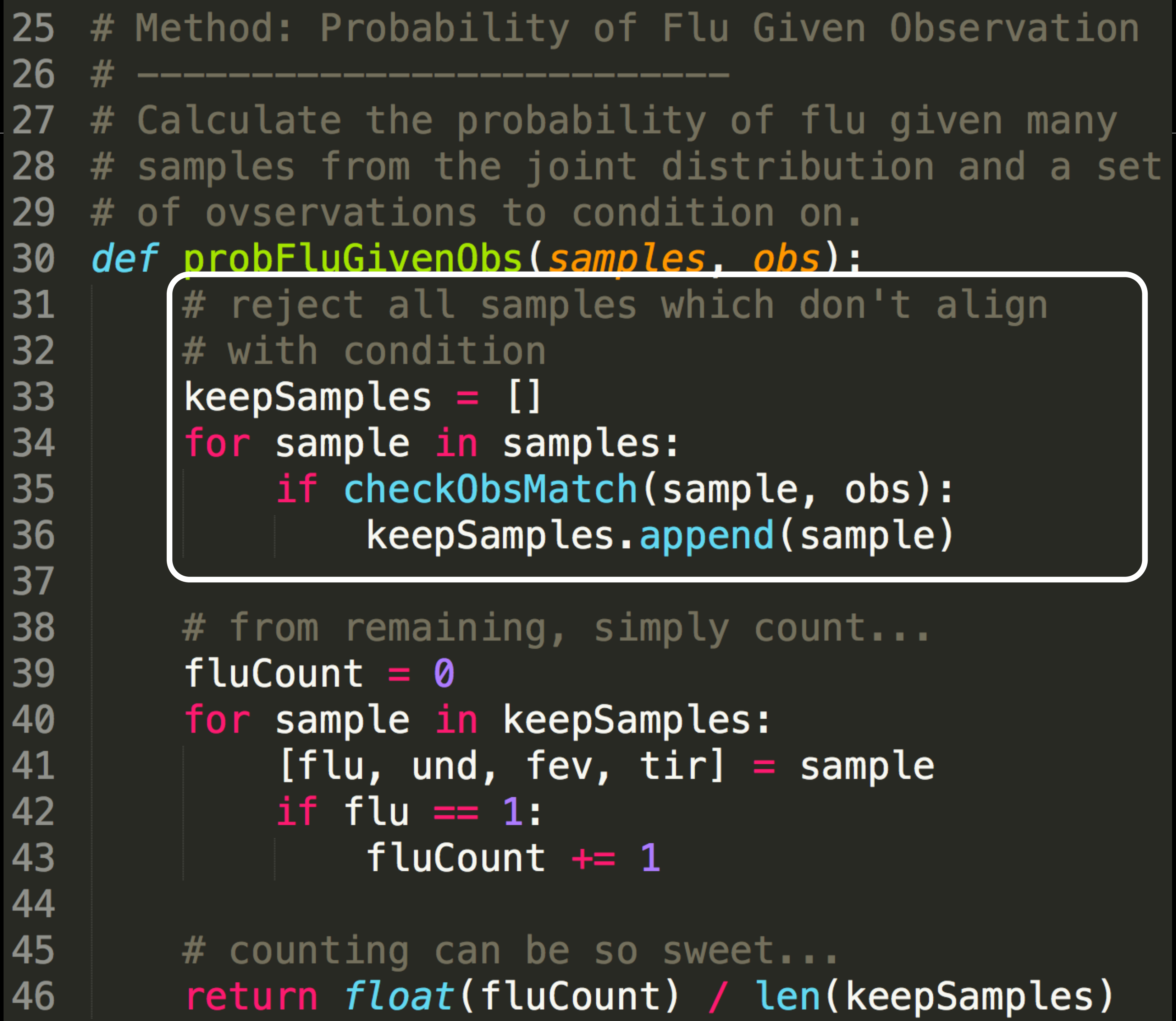


```
[0, 1, 0, 1]
[1, 1, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[1, 1, 0, 1]
```

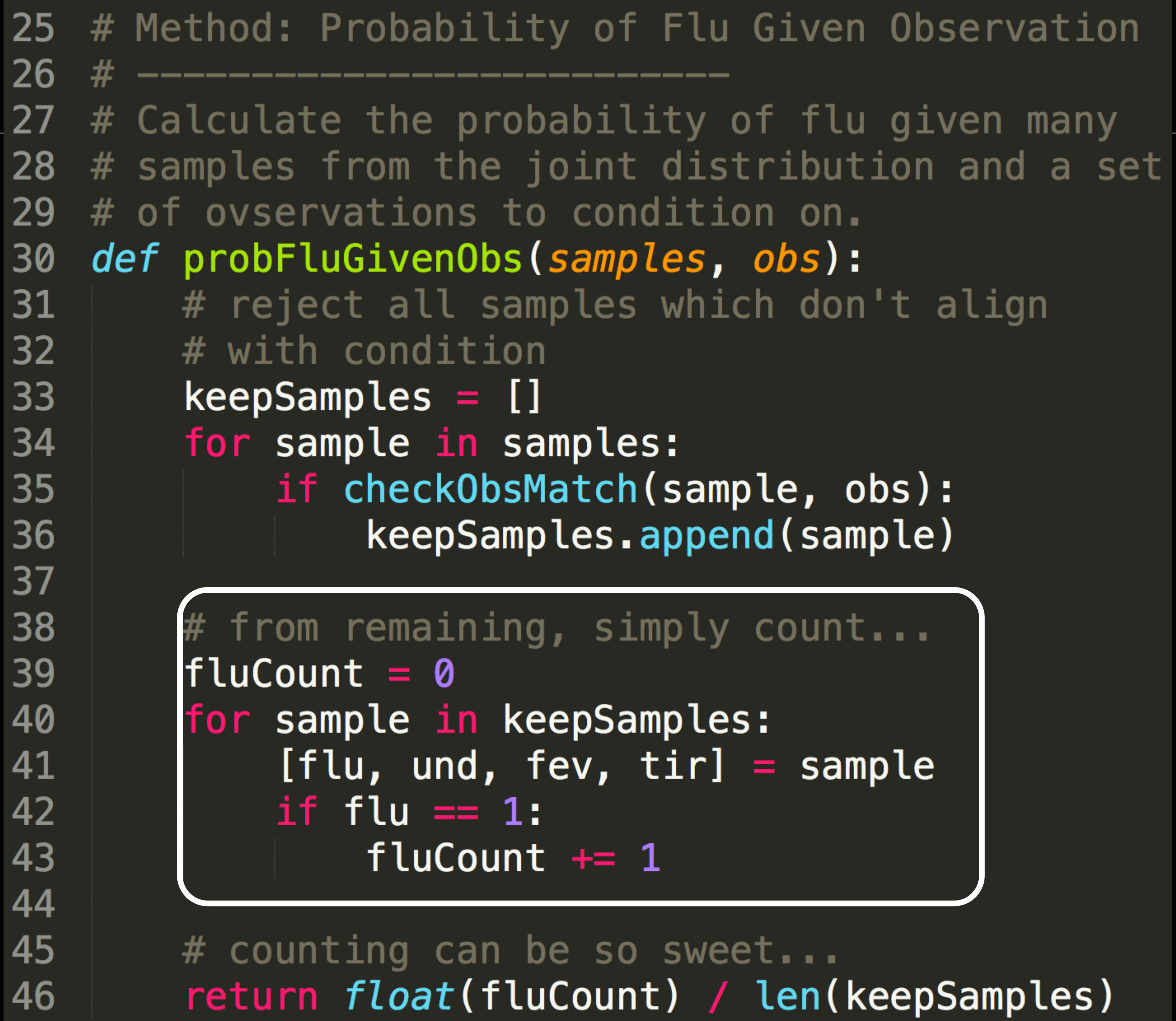
Alg #1: Rejection Sampling



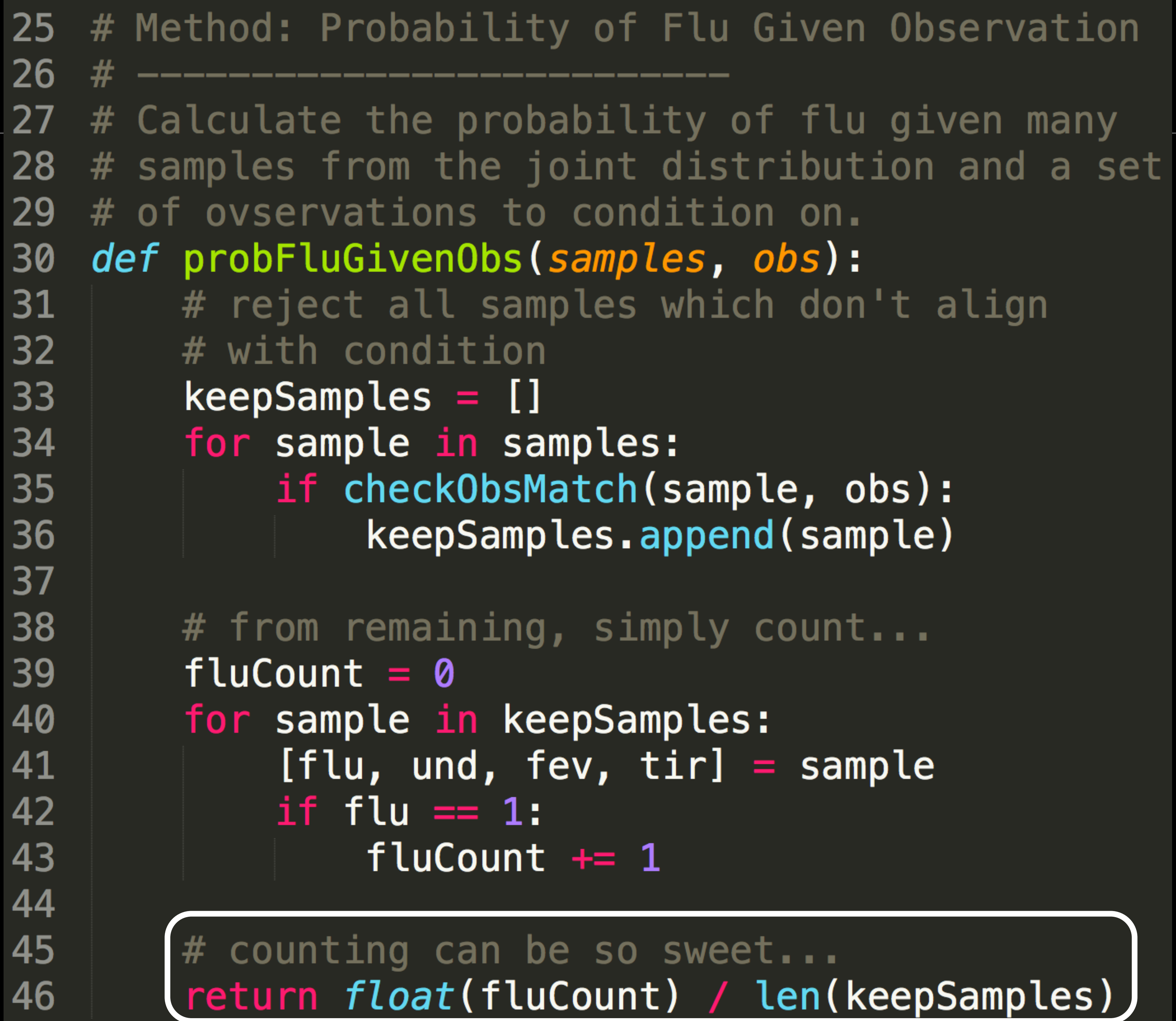
```
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any probability question
# with multivariate samples from the joint,
# where conditioned variables match
def main():
    obs = getObservation()
    print 'Observation = ', obs

    samples = sampleATon()
    prob = probFluGivenObs(samples, obs)
    print 'Pr(Flu) = ', prob
```



```python
# Method: Probability of Flu Given Observation
# ---------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

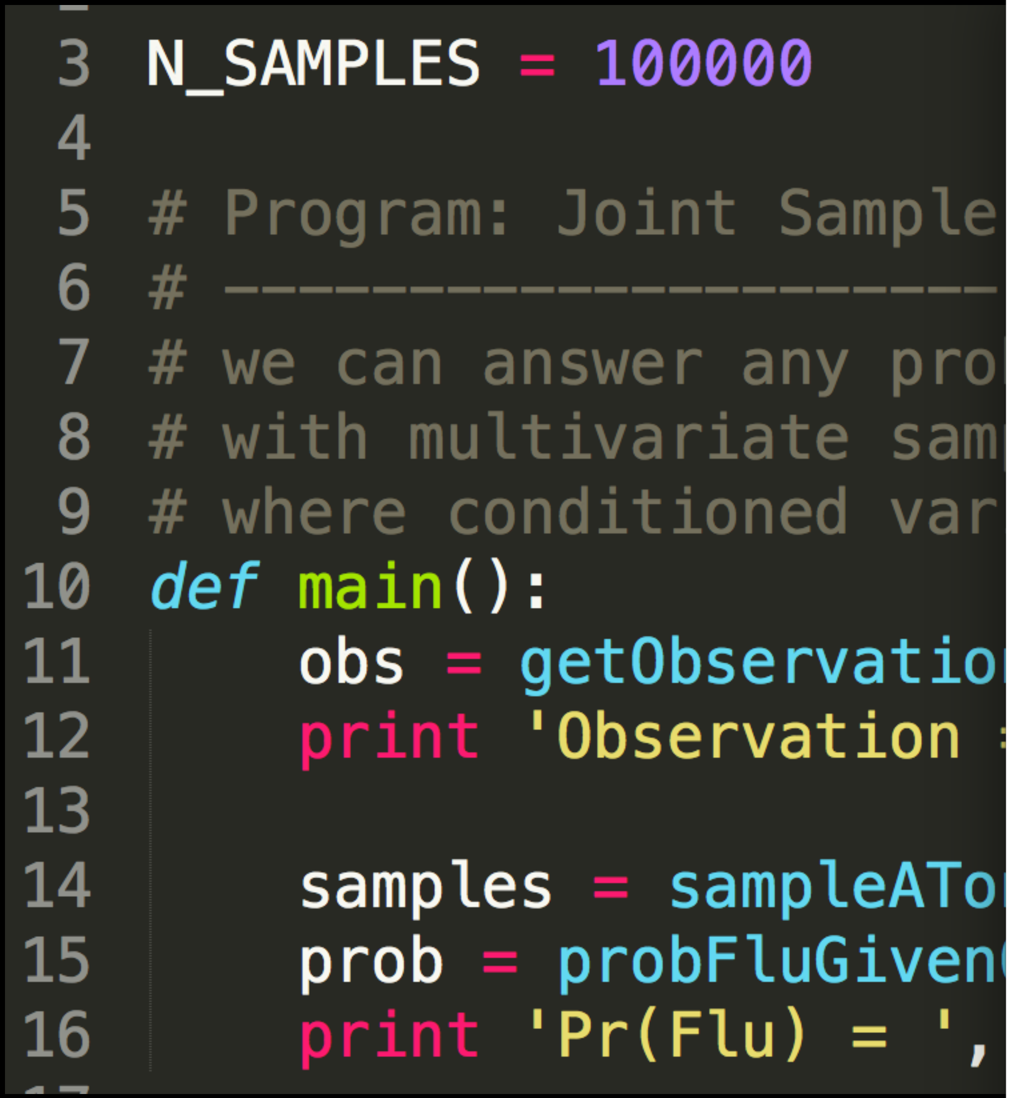


```python
# Method: Probability of Flu Given Observation
# -------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

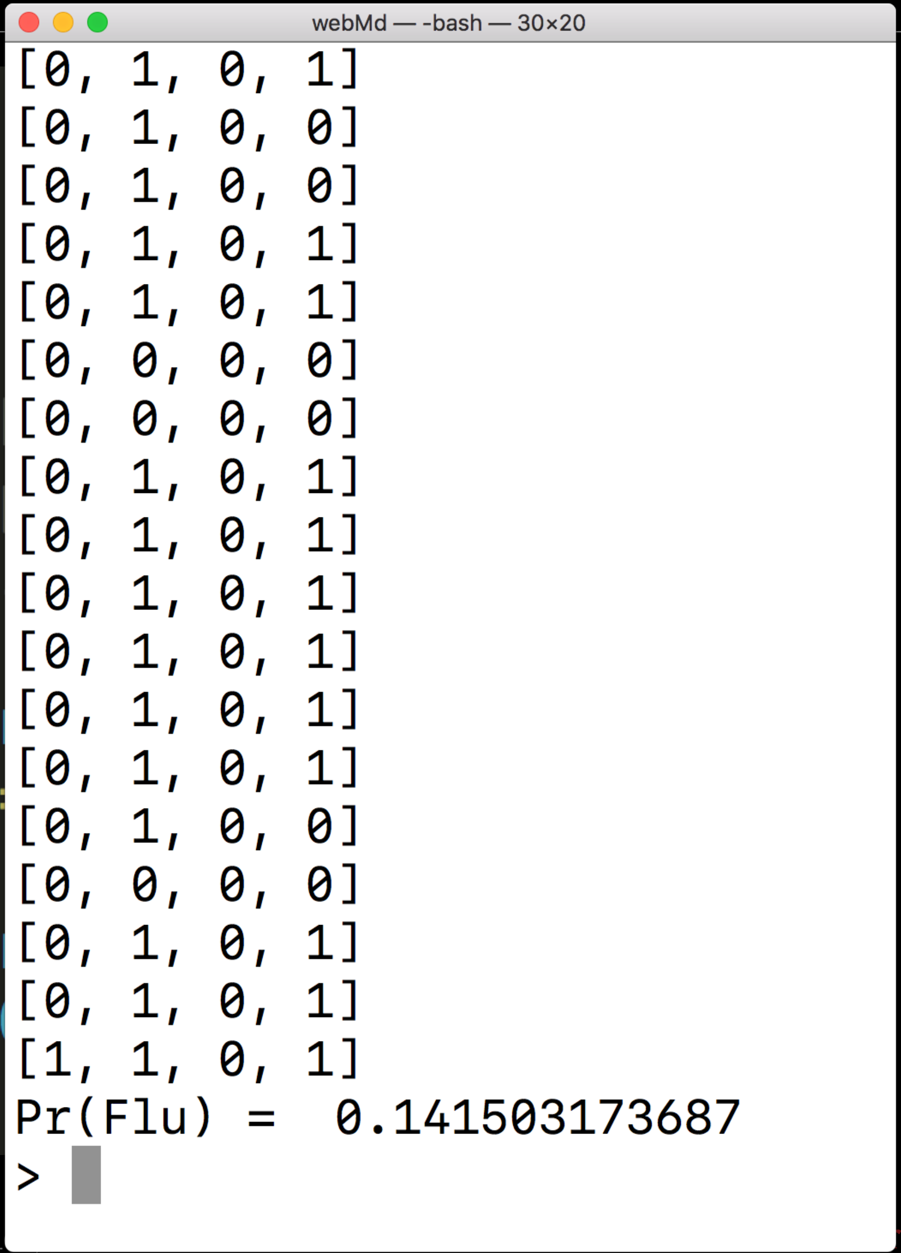


```python
# Method: Probability of Flu Given Observation
# ----------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

Alg #1: Rejection Sampling



```
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any pro
# with multivariate sam
# where conditioned var
def main():
    obs = getObservatio
    print 'Observation =

    samples = sampleATo
    prob = probFluGiven(
    print 'Pr(Flu) = ',
```



```
[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]
[1, 1, 0, 1]
Pr(Flu) =  0.141503173687
> 
```

Lets try it!

BACK TO THE CODE

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

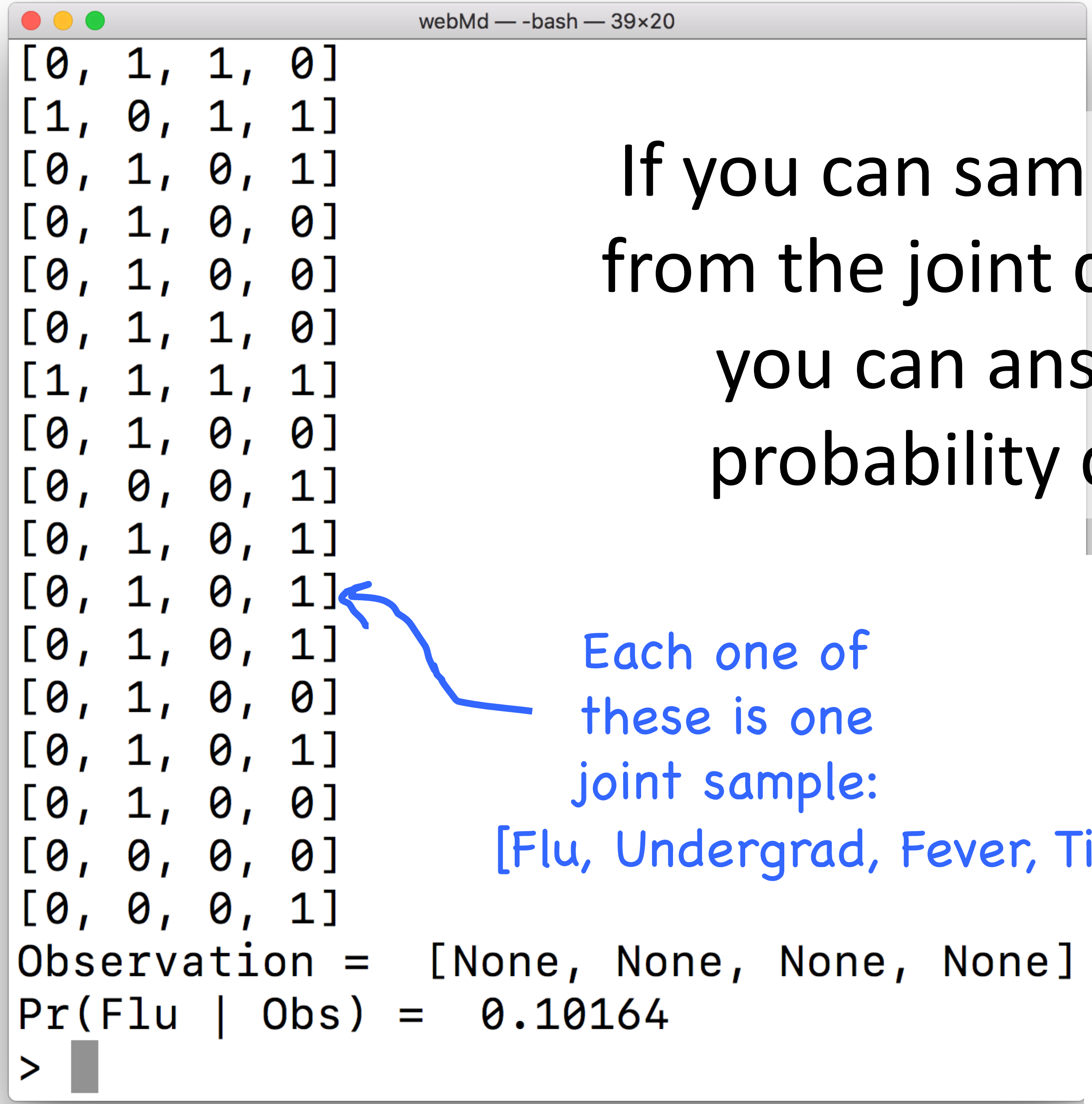
$$\text{probability} \approx \frac{\#\text{ samples with } (F_{lu} = 1, U = 1, T = 1)}{\#\text{ samples with } (U = 1, T = 1)}$$

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



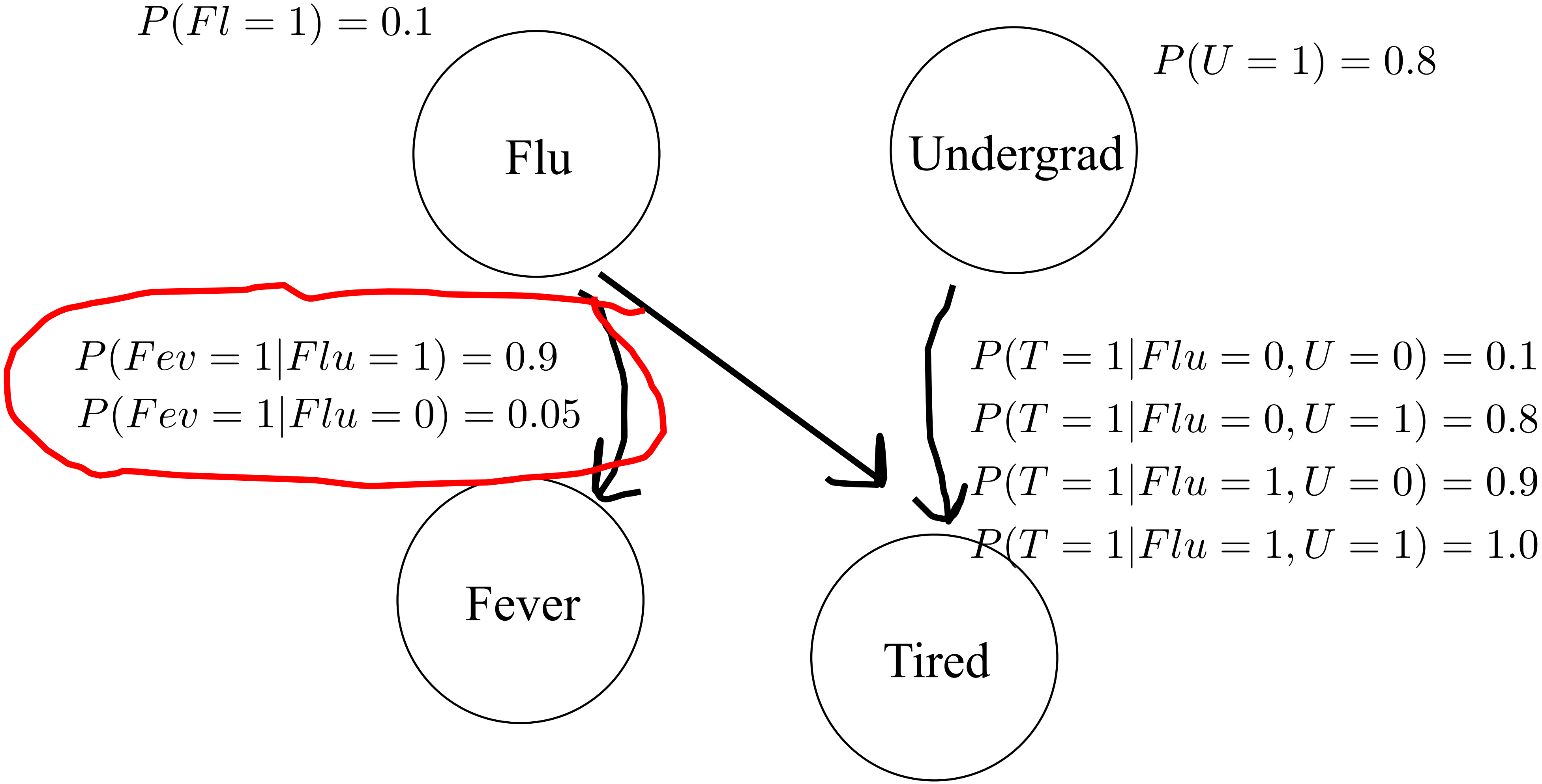
```
[0, 1, 1, 0]
[1, 0, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 1, 0]
[1, 1, 1, 1]
[0, 1, 0, 0]
[0, 0, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 1]
Observation =  [None, None, None, None]
Pr(Flu | Obs) =  0.10164
> 
```

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



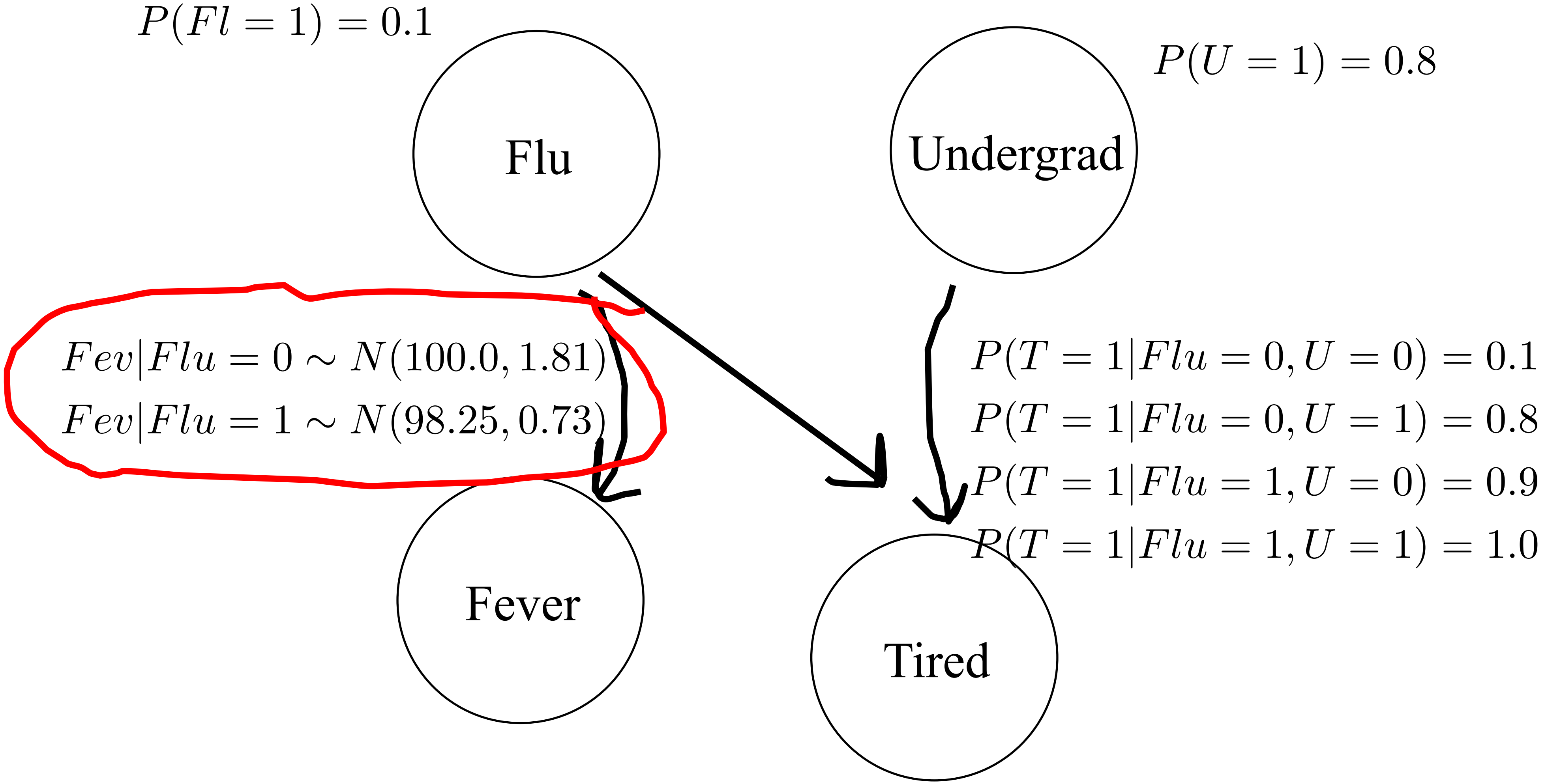
$P(Fl = 1) = 0.1$

$P(U = 1) = 0.8$

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

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

Probabilistic Model



$P(Fl = 1) = 0.1$

$P(U = 1) = 0.8$

$Fev|Flu = 0 \sim N(100.0, 1.81)$

$Fev|Flu = 1 \sim N(98.25, 0.73)$

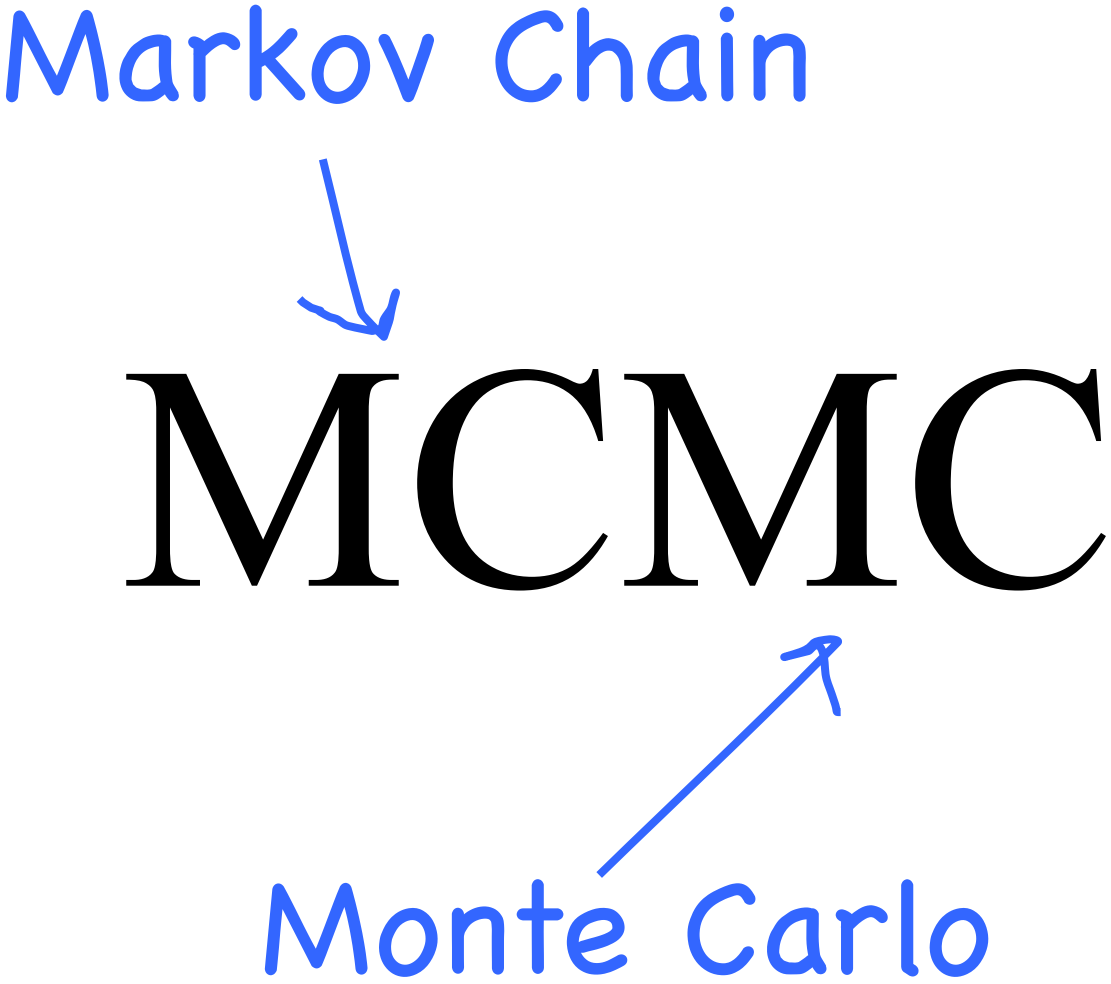
$P(T = 1|Flu = 0, U = 0) = 0.1$

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

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

$P(T = 1|Flu = 1, U = 1) = 1.0$

The Magic School Bus™
Blast Off!

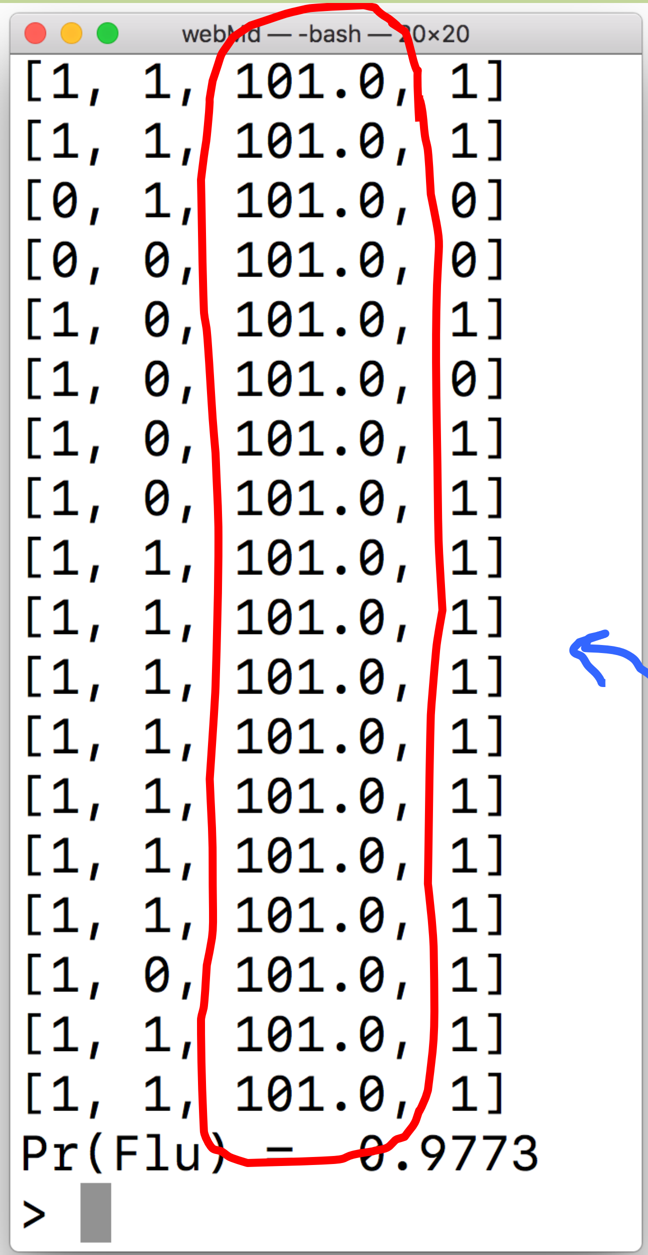


Markov Chain

MCMC

Monte Carlo

Alg #2: MCMC



```
[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

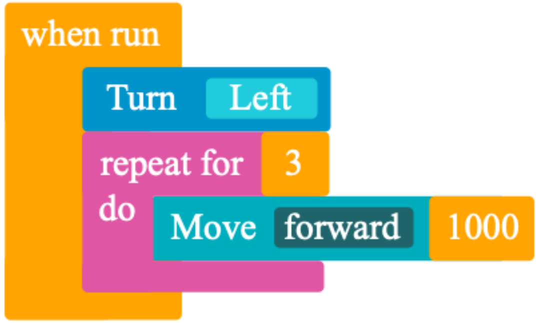
Idea2Text

What about Code.org?

Computers Couldn’t Understand Code

60,000 students attempted this problem
37,000 unique solutions

Challenge



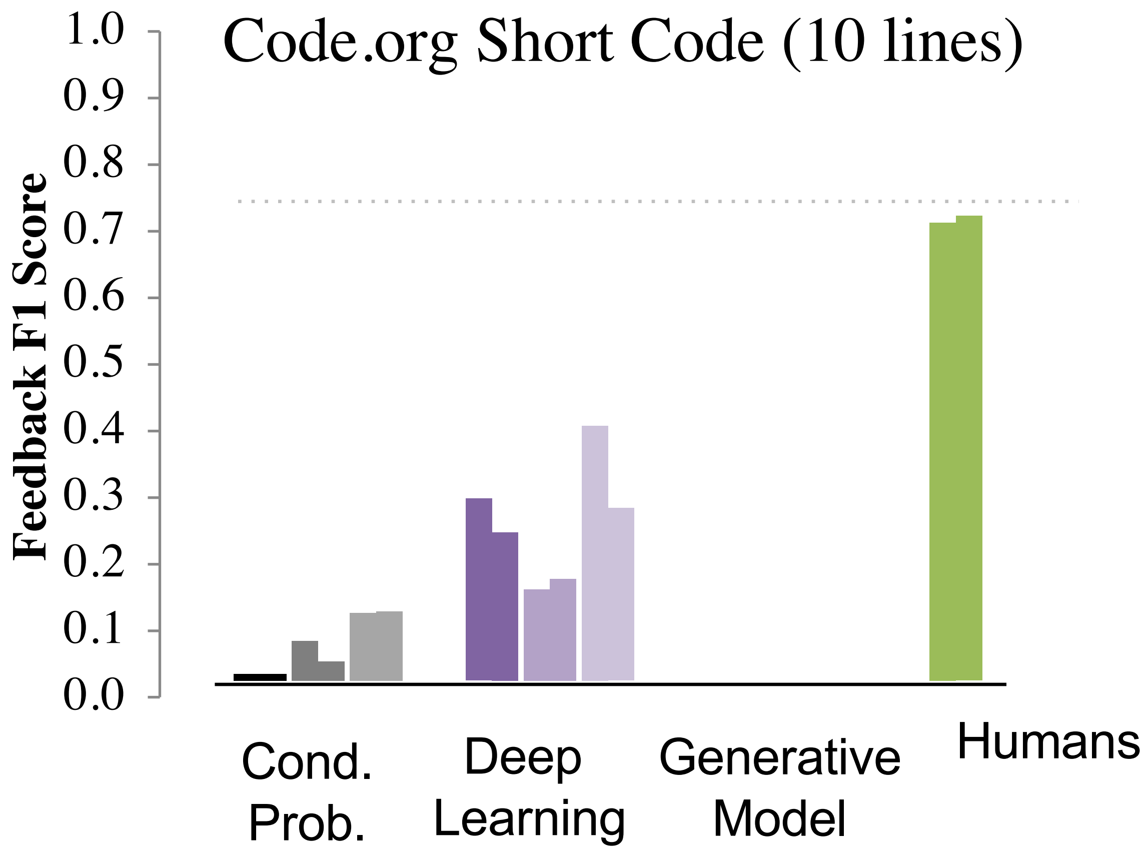
```
when run
    Turn Left
    repeat for 3
    do
        Move forward 1000
```

Student Code

You need to move and turn in your loop

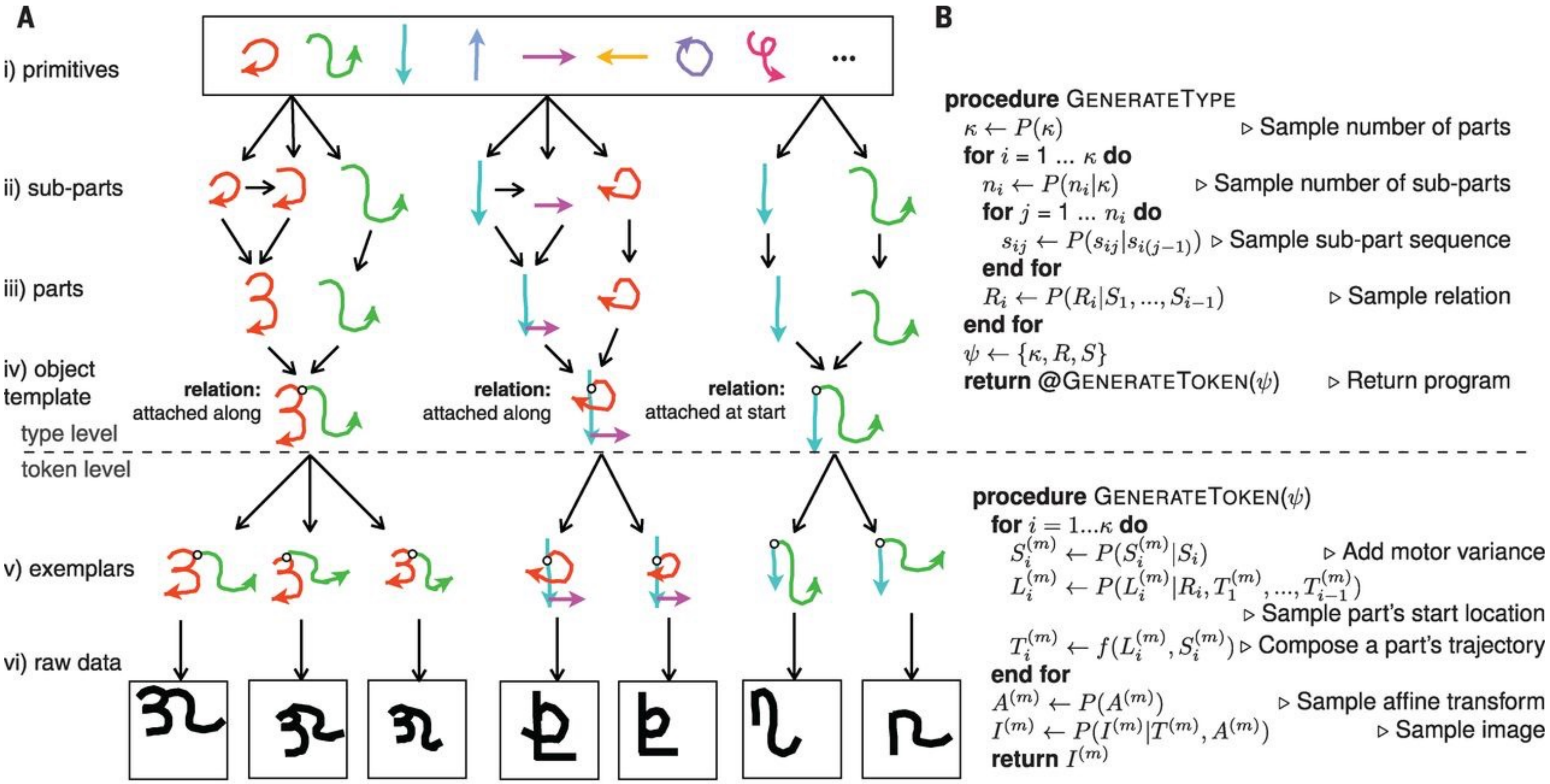
Insight

Computers Couldn't Understand Code



Generative Model of Handwritten Digits

Lake et al, 2015

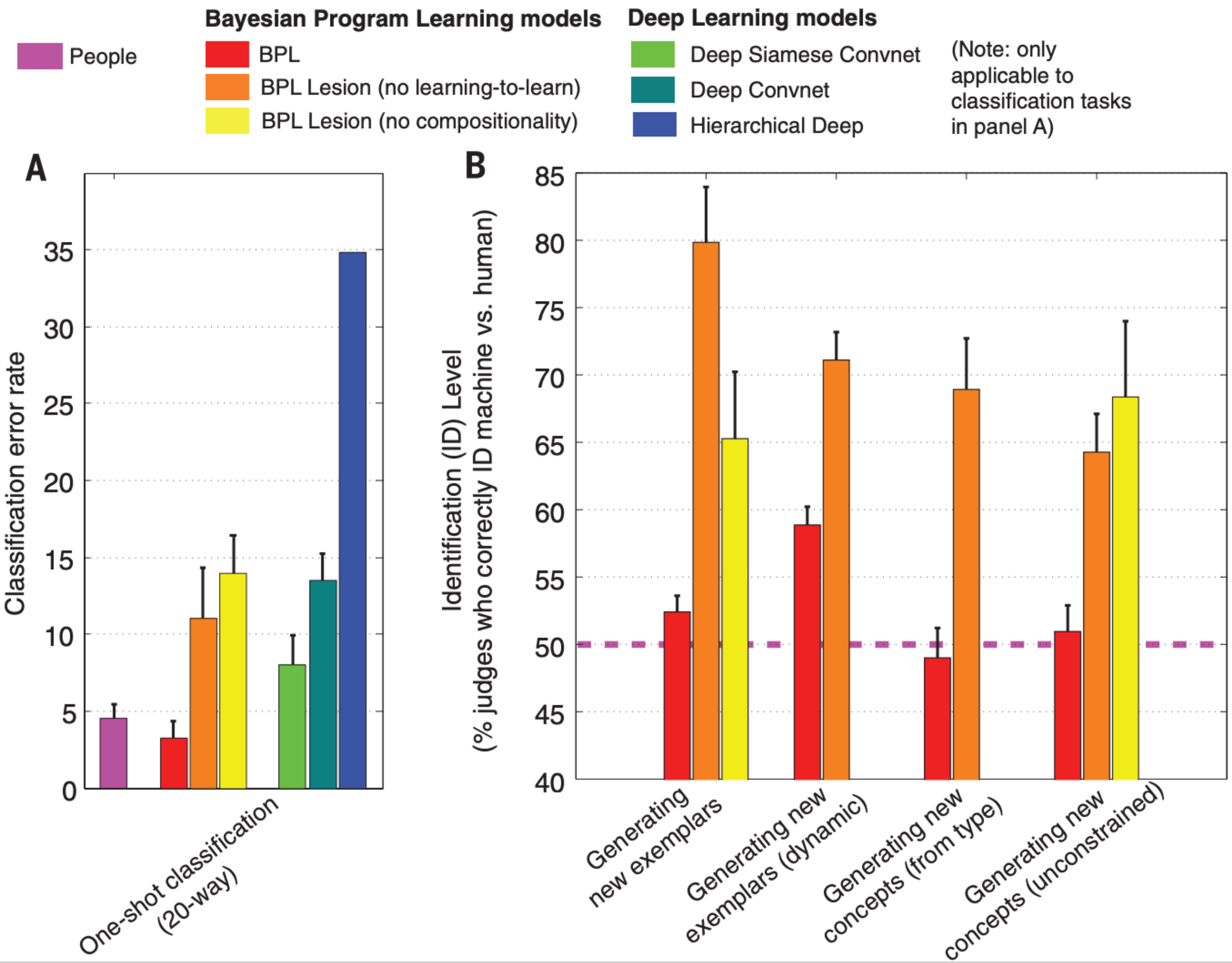


```
procedure GENERATETYPE
  $\kappa \leftarrow P(\kappa)$    ▷ Sample number of parts
  for $i = 1 \ldots \kappa$ do
    $n_i \leftarrow P(n_i|\kappa)$    ▷ Sample number of sub-parts
    for $j = 1 \ldots n_i$ do
      $s_{ij} \leftarrow P(s_{ij}|s_{i(j-1)})$    ▷ Sample sub-part sequence
    end for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

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

Inference. Given a character, infer generation

Human Level. And More!

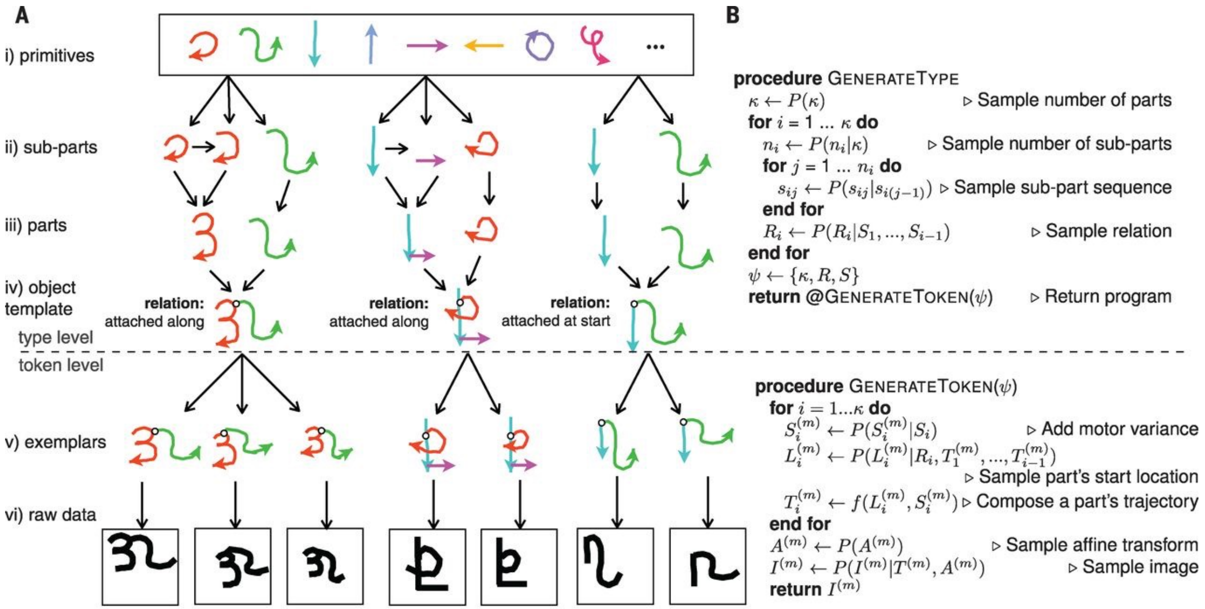


Chris Piech, CS109, 2021

Stanford University

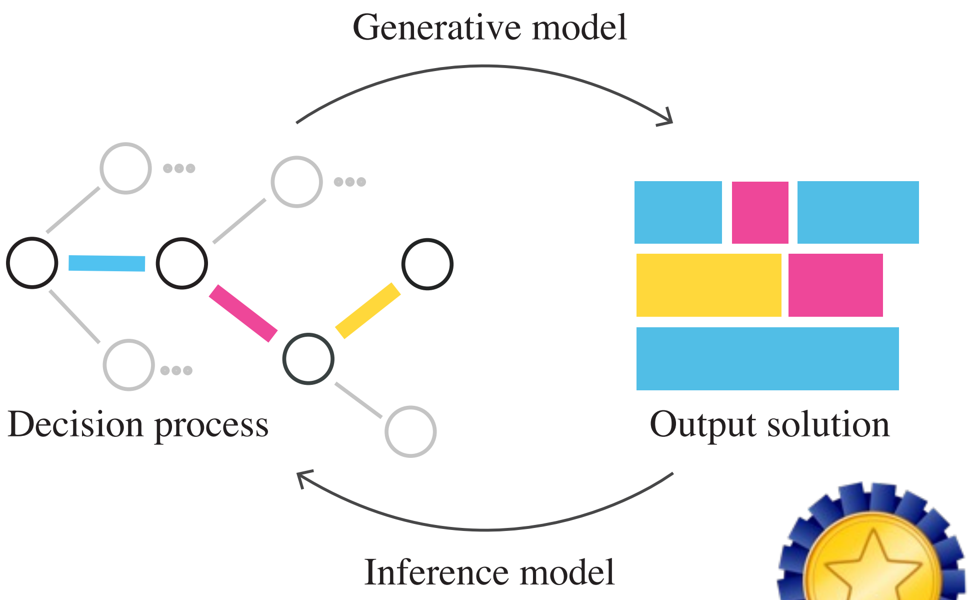
Generative Model of Grading

Lake et al, 2015



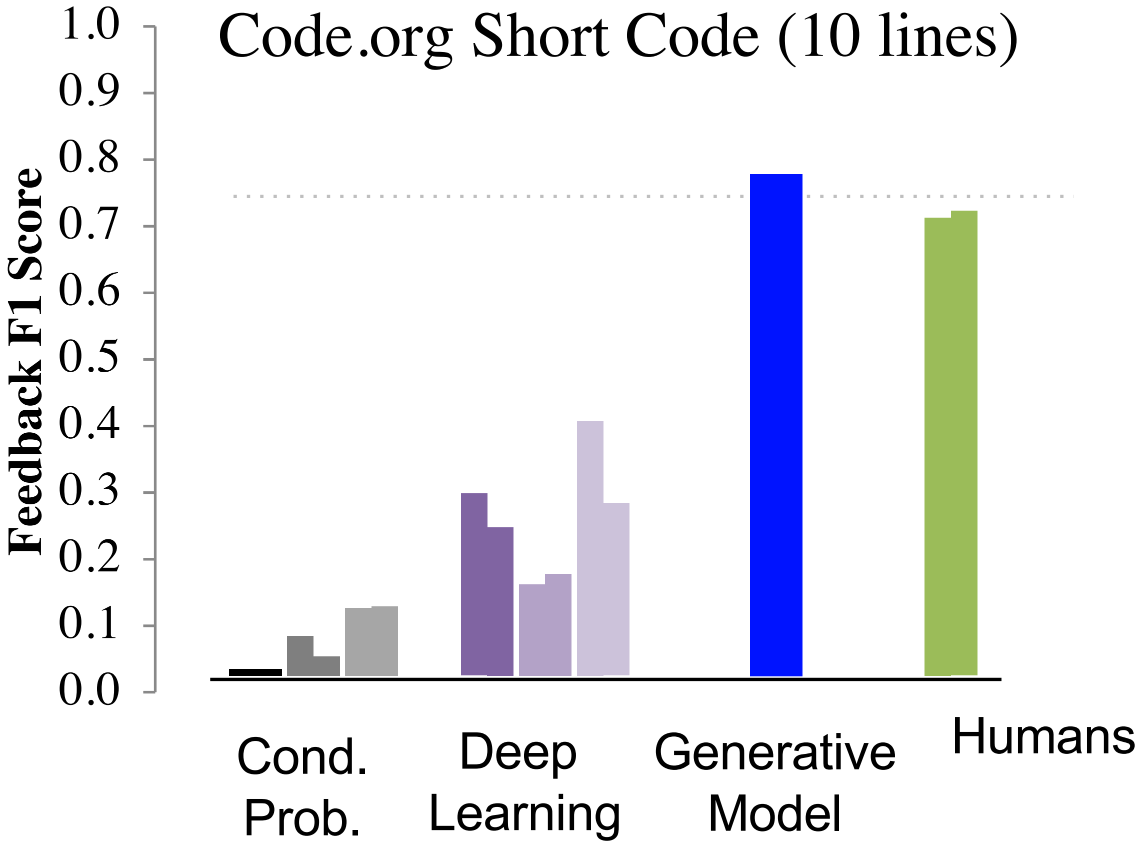
```
procedure GenerateType
  $\kappa \leftarrow P(\kappa)$                         ▷ Sample number of parts
  for $i$ = 1 ... $\kappa$ do
    $n_i \leftarrow P(n_i|\kappa)$                    ▷ Sample number of sub-parts
    for $j$ = 1 ... $n_i$ do
      $s_{ij} \leftarrow P(s_{ij}|s_{i(j-1)})$        ▷ 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($\psi$)                     ▷ Return program

procedure GenerateToken($\psi$)
  for $i = 1 ... \kappa$ 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 $I^{(m)}$
```

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

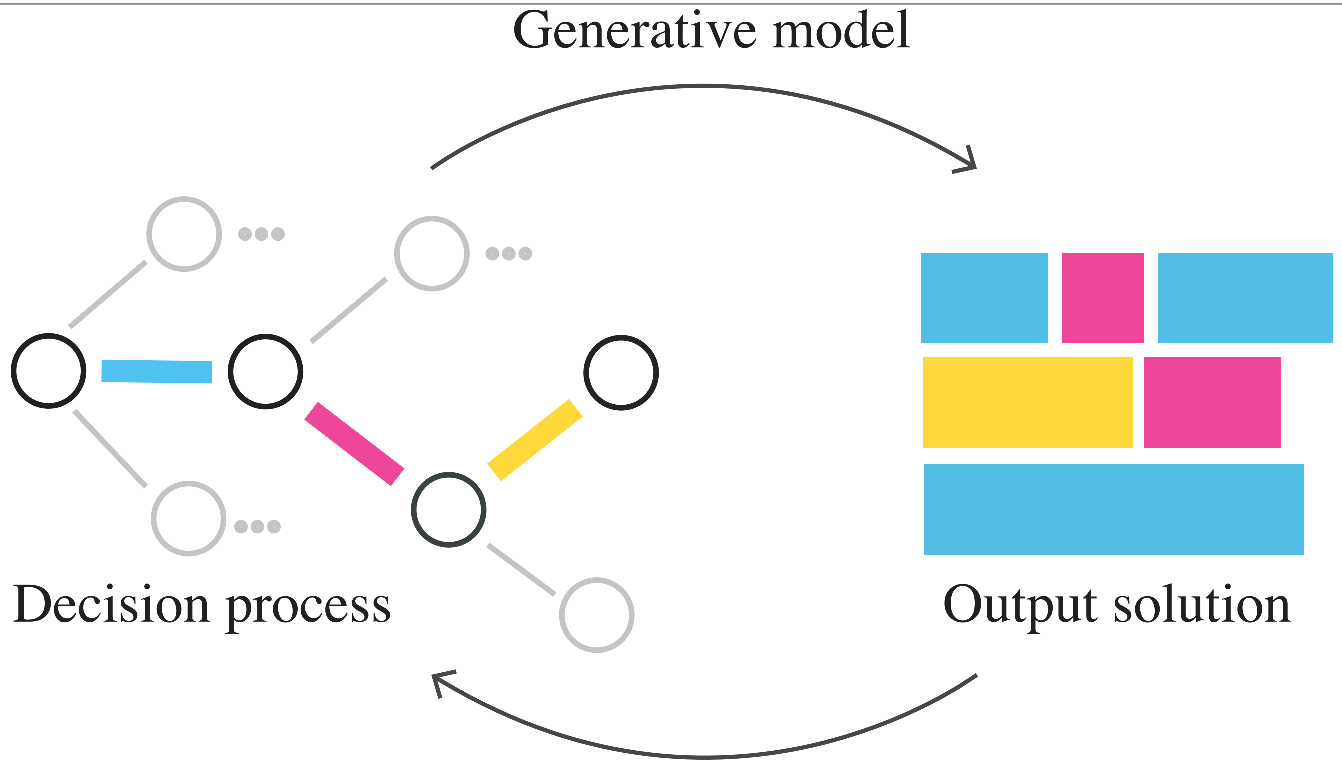


Outstanding Student paper award, AAAI 2019

Computers Couldn't Understand Code

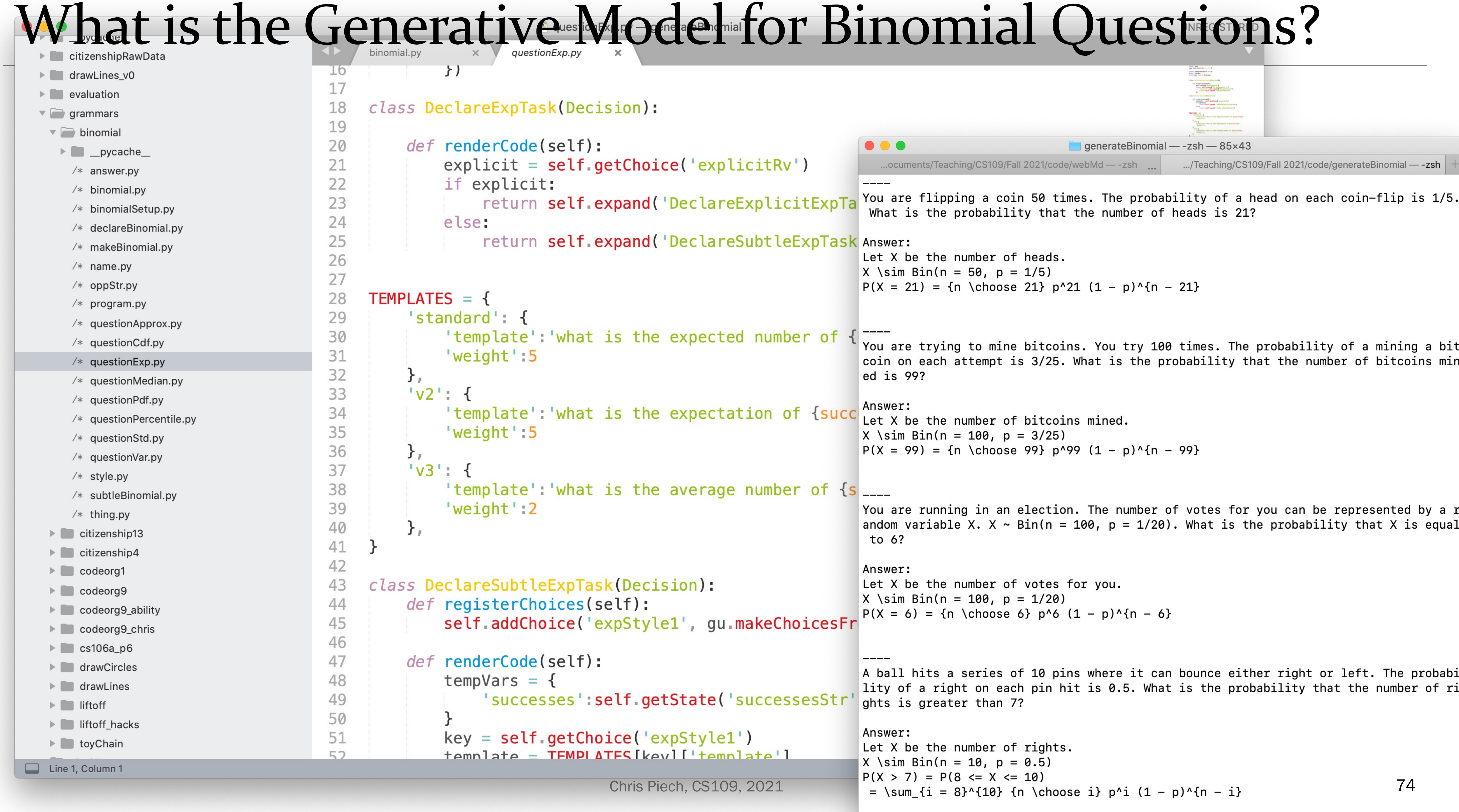


Idea 2 Text



Inference model

What is the Generative Model for Binomial Questions?


binomial.py
questionExp.py
generateBinomial — -zsh — 85×43
...ocuments/Teaching/CS109/Fall 2021/code/webMd — -zsh
.../Teaching/CS109/Fall 2021/code/generateBinomial — -zsh
Line 1, Column 1

```
citizenshipRawData
drawLines_v0
evaluation
grammars
  binomial
    __pycache__
    answer.py
    binomial.py
    binomialSetup.py
    declareBinomial.py
    makeBinomial.py
    name.py
    oppStr.py
    program.py
    questionApprox.py
    questionCdf.py
    questionExp.py
    questionMedian.py
    questionPdf.py
    questionPercentile.py
    questionStd.py
    questionVar.py
    style.py
    subtleBinomial.py
    thing.py
  citizenship13
  citizenship4
  codeorg1
  codeorg9
  codeorg9_ability
  codeorg9_chris
  cs106a_p6
  drawCircles
  drawLines
  liftoff
  liftoff_hacks
  toyChain
```

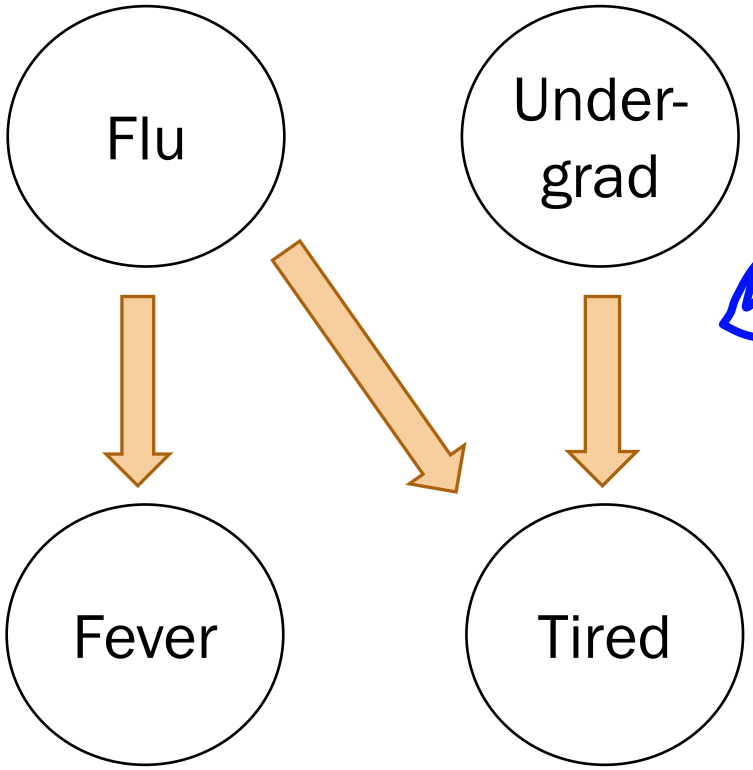
```python
        })

class DeclareExpTask(Decision):

    def renderCode(self):
        explicit = self.getChoice('explicitRv')
        if explicit:
            return self.expand('DeclareExplicitExpTa
        else:
            return self.expand('DeclareSubtleExpTask


TEMPLATES = {
    'standard': {
        'template':'what is the expected number of {
        'weight':5
    },
    'v2': {
        'template':'what is the expectation of {succ
        'weight':5
    },
    'v3': {
        'template':'what is the average number of {s
        'weight':2
    },
}

class DeclareSubtleExpTask(Decision):
    def registerChoices(self):
        self.addChoice('expStyle1', gu.makeChoicesFr

    def renderCode(self):
        tempVars = {
            'successes':self.getState('successesStr'
        }
        key = self.getChoice('expStyle1')
        template = TEMPLATES[key]['template']
```

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

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


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

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


----
You are running in an election. The number of votes for you can be represented by a r
andom variable X. X ~ Bin(n = 100, p = 1/20). What is the probability that X is equal
 to 6?

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


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

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

What haven’t we talked about?

Inference via math

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



1. Learn this from data

2. Learn this from data

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

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

See you at the midterm