

Bayesian Statistics Made Simple

Allen B. Downey
Olin College

Follow along at home

sites.google.com/site/simplebayes

The plan

From Bayes's Theorem to Bayesian inference.

A computational framework.

Work on example problems.

Goals

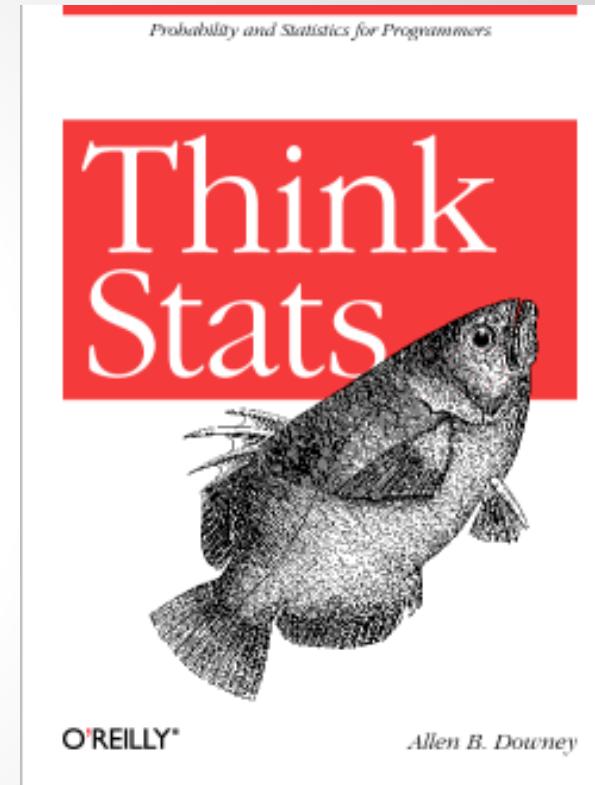
At 4:20, you should be ready to:

- Work on similar problems.
- Learn more on your own.

Think Stats

This tutorial is based on my book,
Think Stats: Probability and Statistics for Programmers.

Published by O'Reilly Media
and available under a
Creative Commons license from
thinkstats.com



Think Bayes

Also based on my new project,

Think Bayes: Bayesian Statistics Made Simple

Current draft available from

thinkbayes.com

Prerequisites

Conditional probability

$p(A)$: the probability that A occurs.

$p(A|B)$: the probability that A occurs, given that B has occurred.

Conjoint probability

$p(A \text{ and } B) = p(A) p(B|A)$

Bayes's Theorem

By definition of conjoint probability:

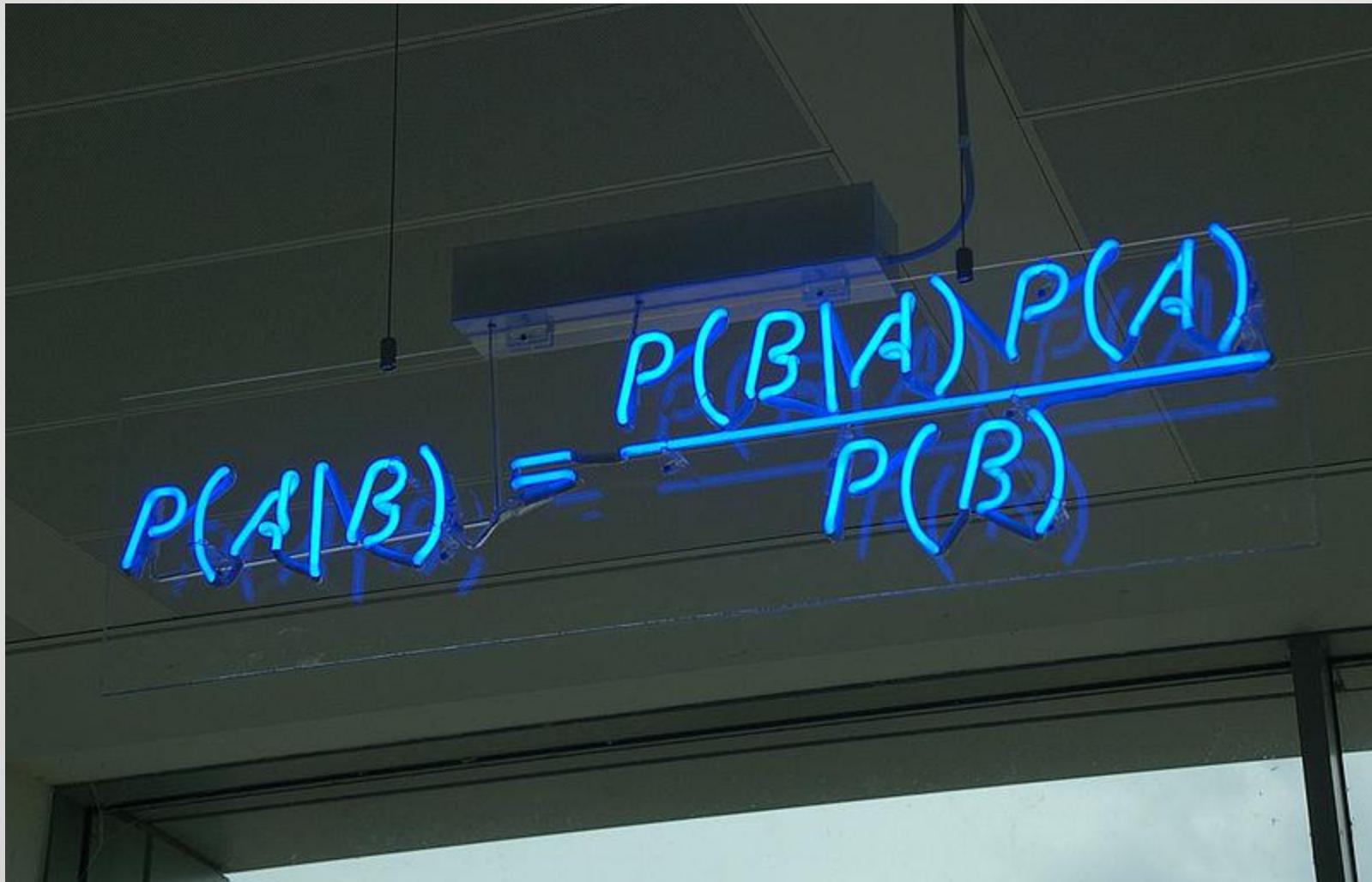
$$\begin{aligned} p(A \text{ and } B) &= p(A) p(B|A) = \\ p(B \text{ and } A) &= p(B) p(A|B) \end{aligned} \tag{1}$$

Equate the right hand sides

$$p(B) p(A|B) = p(A) p(B|A) \tag{2}$$

Divide by $p(B)$ and ...

Bayes' Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$


Bayes's Theorem

One way to think about BT:

Bayes's Theorem is an algorithm to get from $p(B|A)$ to $p(A|B)$.

Useful if $p(B|A)$, $p(A)$ and $p(B)$ are easier than $p(A|B)$.

OR ...

Diachronic interpretation

H: Hypothesis

D: Data

Given $p(H)$, the probability of the hypothesis before you saw the data.

Find $p(H|D)$, the probability of the hypothesis after you saw the data.

A cookie problem

from Wikipedia

Suppose there are two bowls of cookies.

Bowl #1 has 10 chocolate and 30 vanilla.

Bowl #2 has 20 of each.

Fred picks a bowl at random, and then picks a cookie at random. The cookie turns out to be vanilla.

What is the probability that Fred picked from Bowl #1?

Cookie problem

H: Hypothesis that cookie came from Bowl 1.

D: Cookie is vanilla.

Given $p(H)$, the probability of the hypothesis before you saw the data.

Find $p(H|D)$, the probability of the hypothesis after you saw the data.

Diachronic interpretation

$$p(H|D) = p(H) p(D|H) / p(D)$$

$p(H)$: prior

$p(D|H)$: conditional likelihood of the data

$p(D)$: total likelihood of the data

Diachronic interpretation

$$p(H|D) = p(H) p(D|H) / p(D)$$

$p(H)$: prior = 1/2

$p(D|H)$: conditional likelihood of the data = 3/4

$p(D)$: total likelihood of the data = 5/8

Diachronic interpretation

$$p(H|D) = (1/2)(3/4) / (5/8) = 3/5$$

$p(H)$: prior = 1/2

$p(D|H)$: conditional likelihood of the data = 3/4

$p(D)$: total likelihood of the data = 5/8

Computation

Pmf represents a Probability Mass Function

Maps from possible values to probabilities.

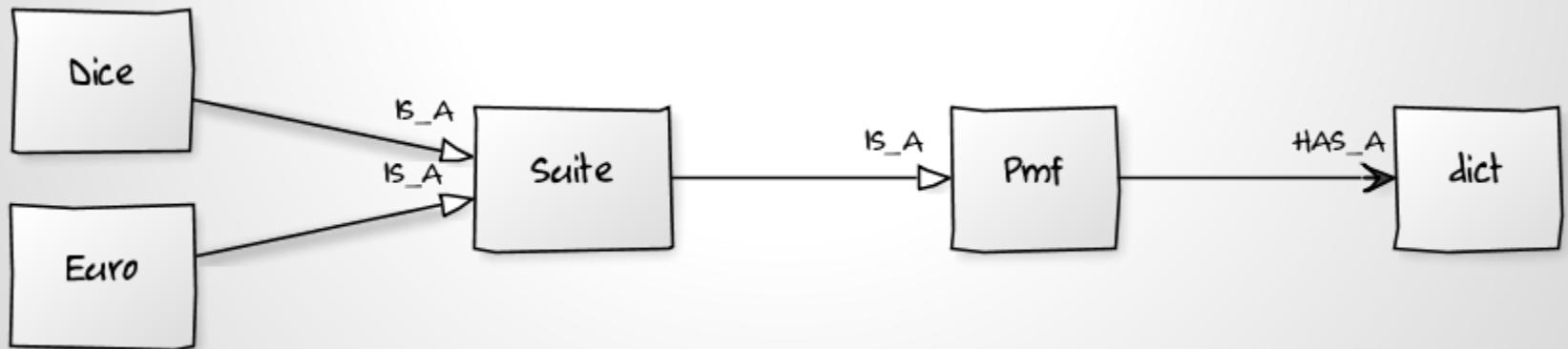


Diagram by yuml.me

Install test

How many of you got `install_test.py` running?

Don't try to fix it now!

Instead...

Partner up

- If you don't have a working environment, find a neighbor who does.
- Even if you do, try pair programming!
- Take a minute to introduce yourself.
- Questions? Ask your partner first (please).

Icebreaker

What was your first computer?

What was your first programming language?

What is the longest time you have spent finding a stupid bug?

Start your engines

1. You downloaded bayes_pycon13.zip, right?
2. cd into that directory (or set PYTHON_PATH)
3. Start the Python interpreter.

```
>>> from thinkbayes import Pmf
```

Pmf

```
from thinkbayes import Pmf

# make an empty Pmf
pmf = Pmf()

# outcomes of a six-sided die
for x in [1,2,3,4,5,6]:
    pmf.Set(x, 1)
```

Pmf

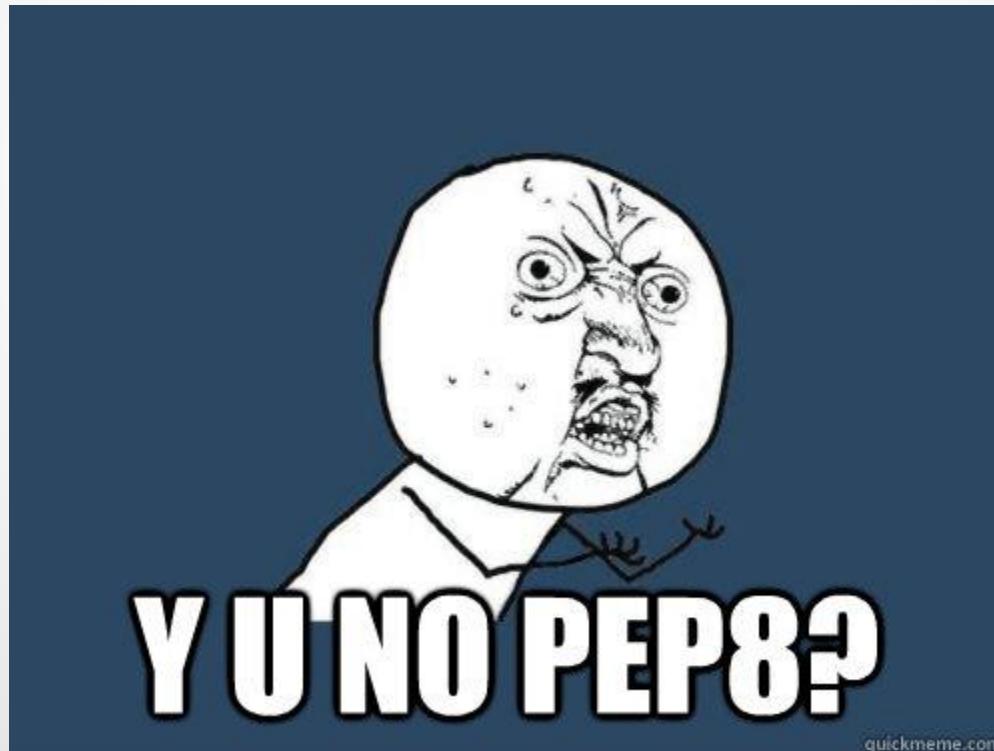
```
pmf.Print()
```

```
pmf.Normalize()
```

```
pmf.Print()
```

Style

I know what you're thinking.



The framework

- 1) Build a Pmf that maps from each hypothesis to a **prior probability**, $p(H)$.
- 2) Multiply each prior probability by the **likelihood** of the data, $p(D|H)$.
- 3) **Normalize**, which divides through by the total likelihood, $p(D)$.

Prior

```
pmf = Pmf()
```

```
pmf.Set('Bowl 1', 0.5)
```

```
pmf.Set('Bowl 2', 0.5)
```

Update

$p(\text{Vanilla} \mid \text{Bowl } 1) = 30/40$
 $p(\text{Vanilla} \mid \text{Bowl } 2) = 20/40$

`pmf.Mult('Bowl 1', 0.75)
pmf.Mult('Bowl 2', 0.5)`

Normalize

```
pmf.Normalize()
```

```
print pmf.Prob('Bowl 1')
```

Summary

Bayes's Theorem,
Cookie problem,
Pmf class.



The dice problem

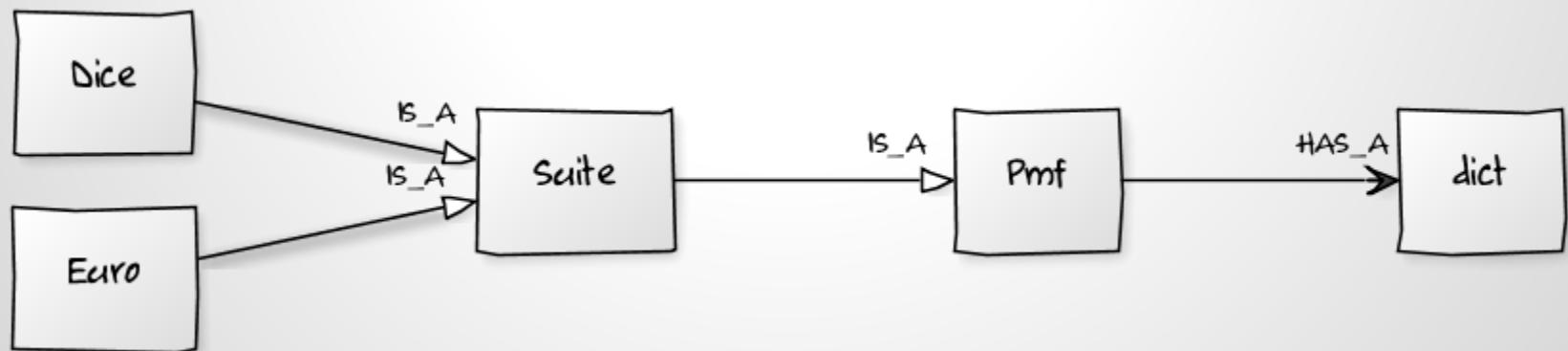
I have a box of dice that contains a 4-sided die, a 6-sided die, an 8-sided die, a 12-sided die and a 20-sided die.

Suppose I select a die from the box at random, roll it, and get a 6. What is the probability that I rolled each die?

Hypothesis suites

A **suite** is a mutually exclusive and collectively exhaustive set of hypotheses.

Represented by a Suite that maps
hypothesis → probability.



Suite

```
class Suite(Pmf):  
    """Represents a suite of hypotheses and  
    their probabilities."  
  
    def __init__(self, hypos):  
        """Initializes the distribution."  
        for hypo in hypos:  
            self.Set(hypo, 1)  
        self.Normalize()
```

Dice

```
from thinkbayes import Suite  
  
# start with equal priors  
suite = Suite([4, 6, 8, 12, 20])  
  
suite.Print()
```

Suite

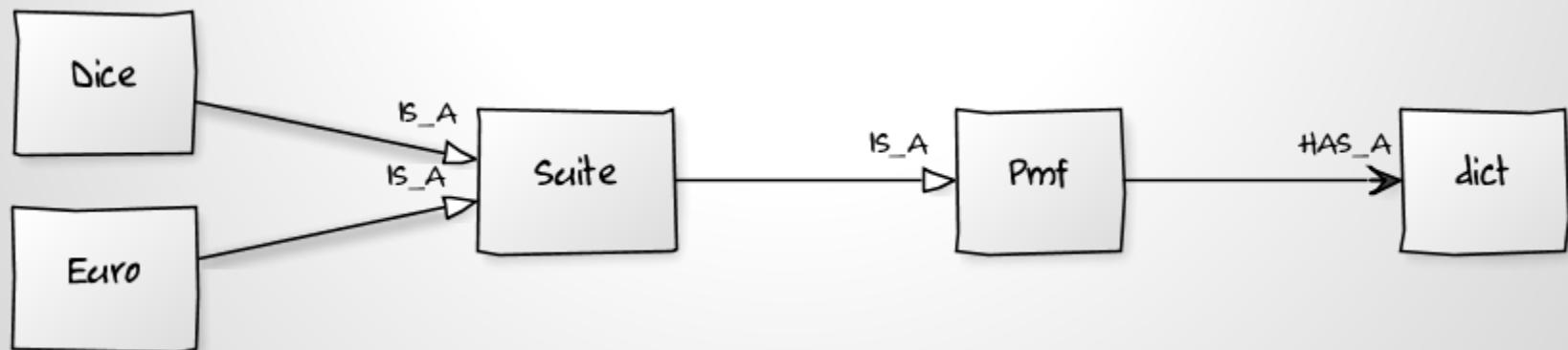
```
def Update(self, data):  
    "Updates the suite based on data."  
  
    for hypo in self.Values():  
        like = self.Likelihood(hypo, data)  
        self.Mult(hypo, like)  
  
    self.Normalize()
```

self.Likelihood?

Suite

Likelihood **is** an abstract method.

Child classes inherit Update,
provide Likelihood.



Likelihood

Outcome: 6

- What is the likelihood of this outcome on a six-sided die?
- On a ten-sided die?
- On a four-sided die?

What is the likelihood of getting n on an m -sided die?

Likelihood

```
# hypo is the number of sides on the die
# data is the outcome

class Dice(Suite):

    def Likelihood(self, hypo, data):
        # write this method!
```

Write your solution in `dice.py`

Likelihood

```
# hypo is the number of sides on the die
# data is the outcome

class Dice(Suite):

    def Likelihood(self, hypo, data):
        if hypo < data:
            return 0
        else:
            return 1.0/hypo
```

Dice

```
# start with equal priors
suite = Dice([4, 6, 8, 12, 20])

# update with the data
suite.Update(6)

suite.Print()
```

Dice

Posterior distribution:

4	0 . 0
6	0 . 39
8	0 . 30
12	0 . 19
20	0 . 12

More data? No problem...

Dice

```
for roll in [6, 4, 8, 7, 7, 2]:  
    suite.Update(roll)  
  
suite.Print()
```

Dice

Posterior distribution:

4 0.0

6 0.0

8 0.92

12 0.080

20 0.0038

Summary

Dice problem,
Likelihood function,
Suite class.



Recess!



Tanks

The German tank problem.

http://en.wikipedia.org/wiki/German_tank_problem



Trains

The trainspotting problem:

- You believe that a freight carrier operates between 100 and 1000 locomotives with consecutive serial numbers.
- You spot locomotive #321.
- How many trains does the carrier operate?

Modify `train.py` to compute your answer.

Trains

- If there are m trains, what is the chance of spotting train # n ?
- What does the posterior distribution look like?
- What if we spot more trains?
- Why did we do this example?

A Euro problem

"When spun on edge 250 times, a Belgian one-euro coin came up heads 140 times and tails 110. 'It looks very suspicious to me,' said Barry Blight, a statistics lecturer at the London School of Economics. 'If the coin were unbiased, the chance of getting a result as extreme as that would be less than 7%.' "

From "The Guardian" quoted by MacKay, *Information Theory, Inference, and Learning Algorithms*.

A Euro problem

MacKay asks, *"But do these data give evidence that the coin is biased rather than fair?"*

Assume that the coin has probability x of landing heads.

(Forget that x is a probability; just think of it as a physical characteristic.)

A Euro problem

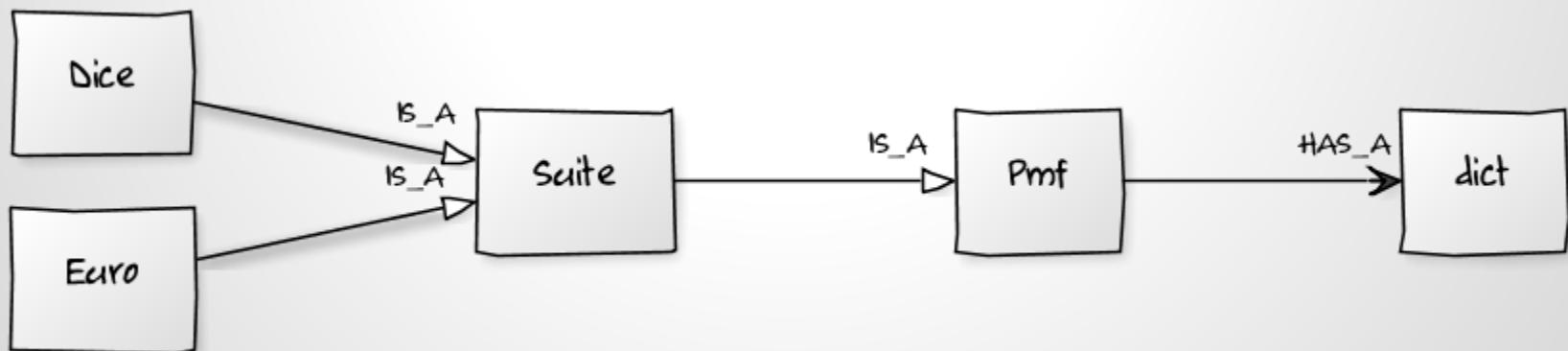
Estimation: Based on the data (140 heads, 110 tails), what is x ?

Hypothesis testing: What is the probability that the coin is fair?

Euro

We can use the Suite template again.

We just have to figure out the likelihood function.



Likelihood

```
# hypo is the prob of heads (1-100)
# data is a string, either 'H' or 'T'

class Euro(Suite):

    def Likelihood(self, hypo, data):
        # one more, please!
```

Modify euro.py to compute your answer.

Likelihood

```
# hypo is the prob of heads (1-100)
# data is a string, either 'H' or 'T'

class Euro(Suite):

    def Likelihood(self, hypo, data):
        x = hypo / 100.0
        if data == 'H':
            return x
        else:
            return 1-x
```

Prior

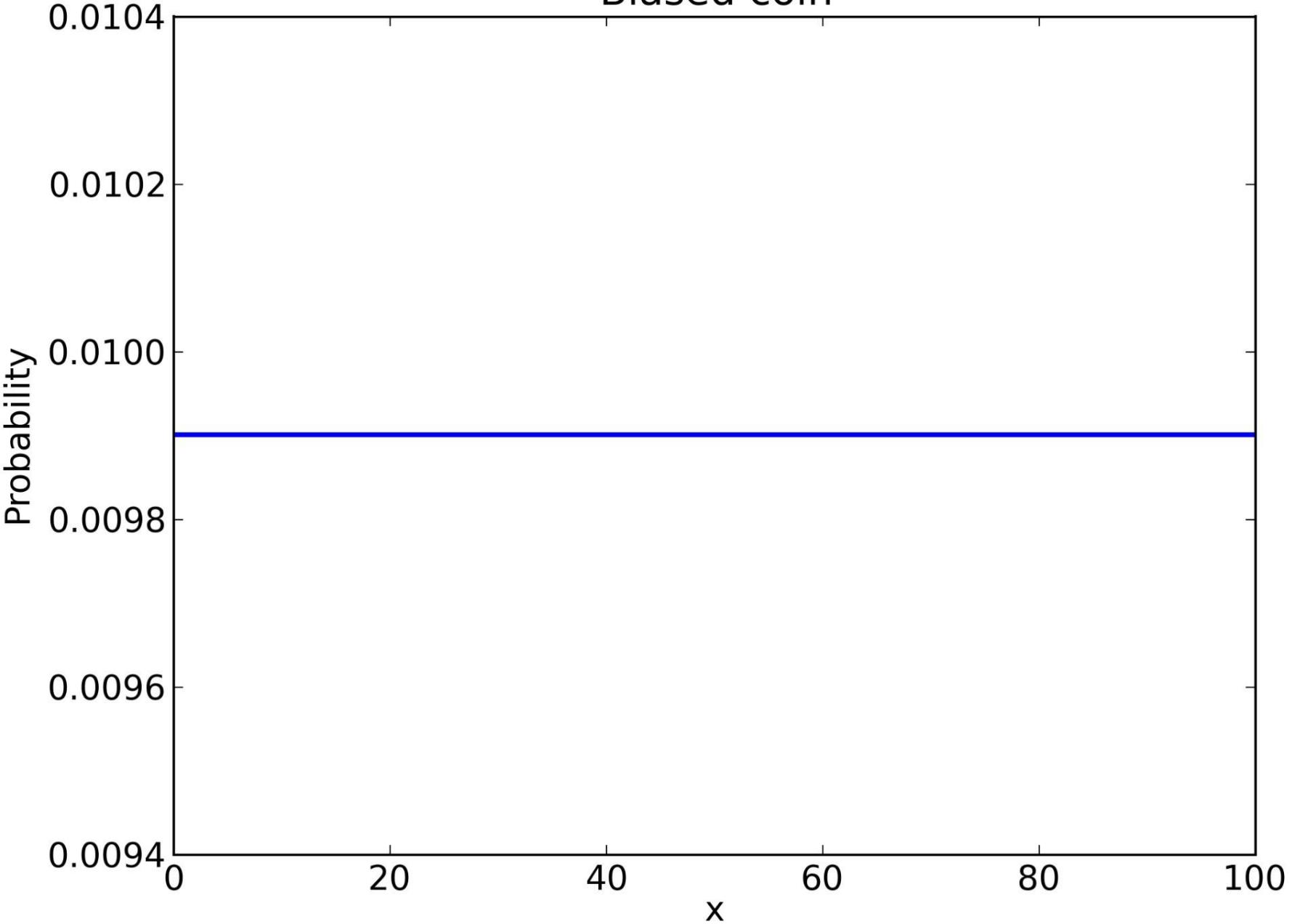
Start with something simple; we'll come back and review.

Uniform prior: any value of x between 0% and 100% is equally likely.

Prior

```
suite = Euro(range(0, 101))
```

Biased coin



Update

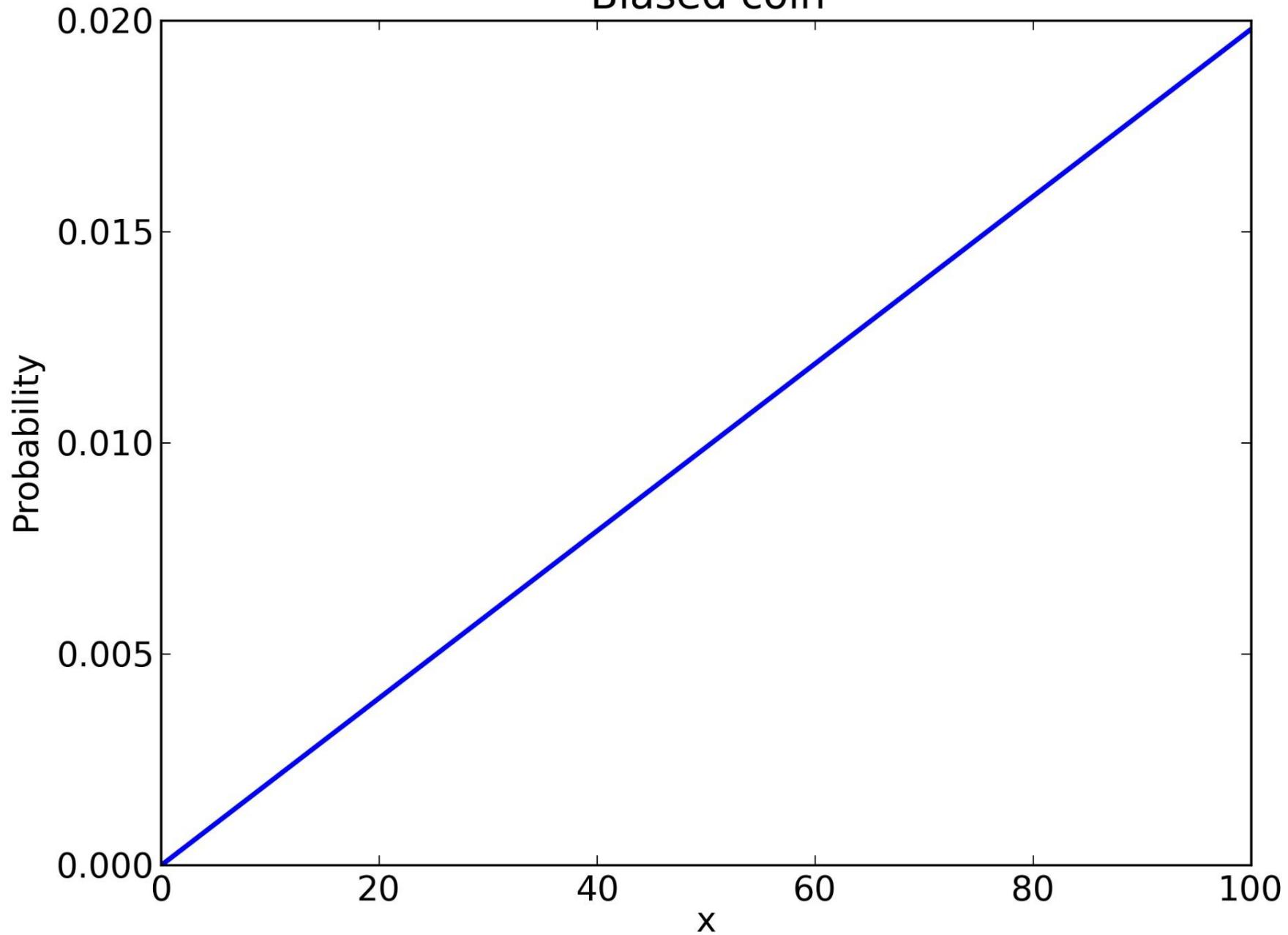
Suppose we spin the coin once and get heads.

```
suite.Update( 'H' )
```

What does the posterior distribution look like?

Hint: what is $p(x=0\% | D)$?

Biased coin



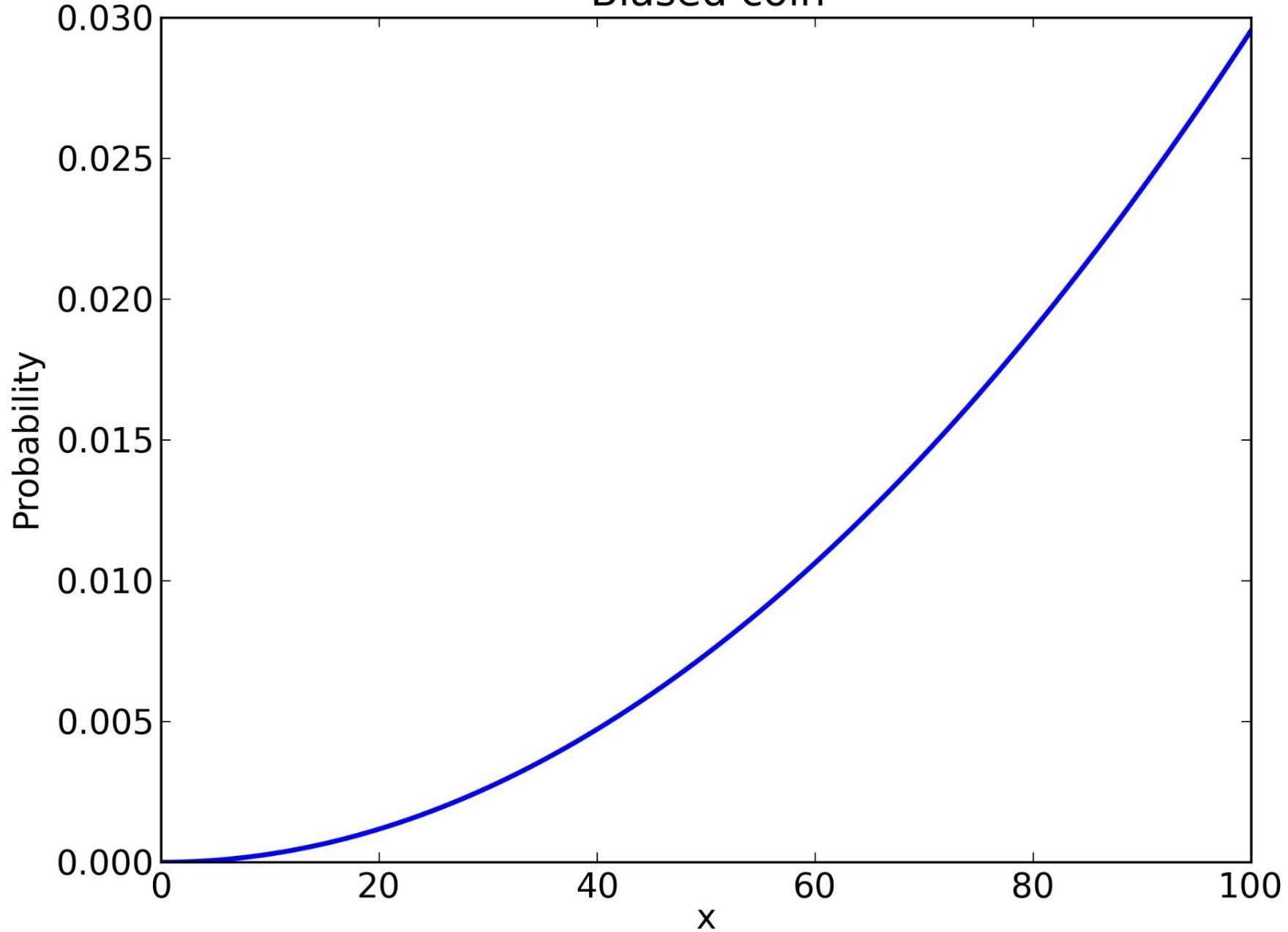
Update

Suppose we spin the coin again, and get heads again.

```
suite.Update('H')
```

What does the posterior distribution look like?

Biased coin



Update

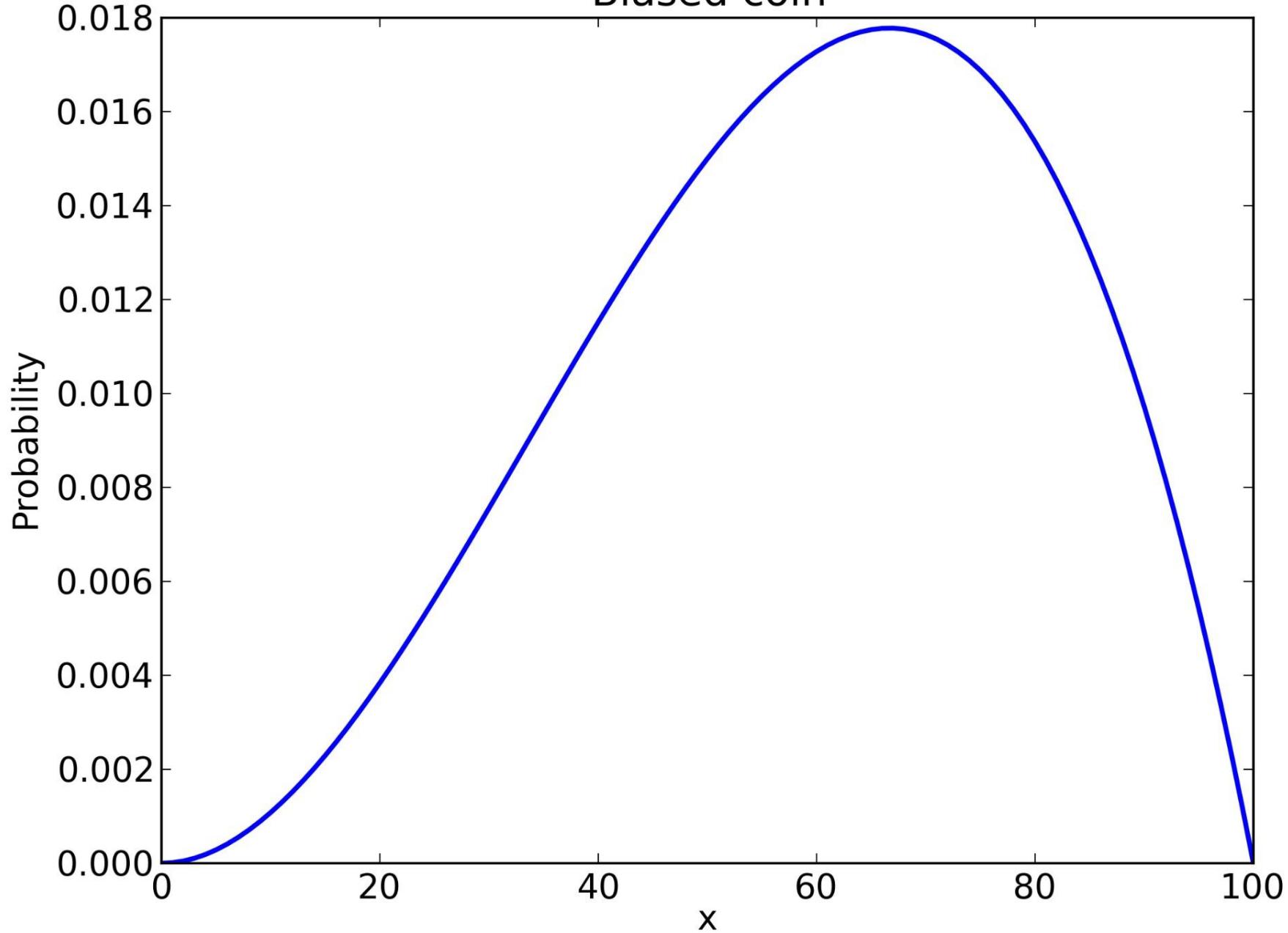
Suppose we spin the coin again, and get tails.

```
suite.Update('T')
```

What does the posterior distribution look like?

Hint: what's $p(x=100\% | D)$?

Biased coin

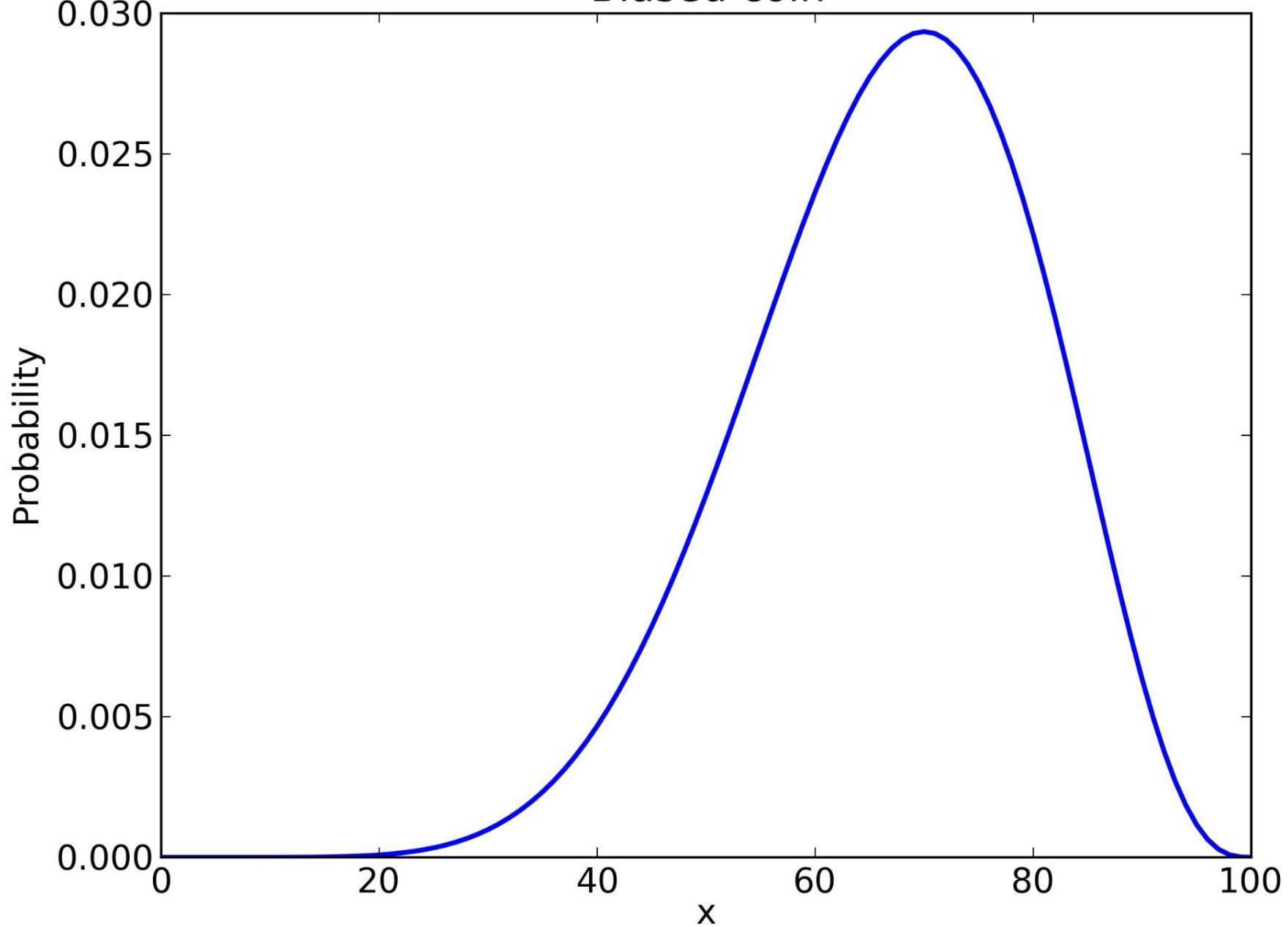


Update

After 10 spins, 7 heads and 3 tails:

```
for outcome in 'HHHHHHHHTT' :  
    suite.Update(outcome)
```

Biased coin

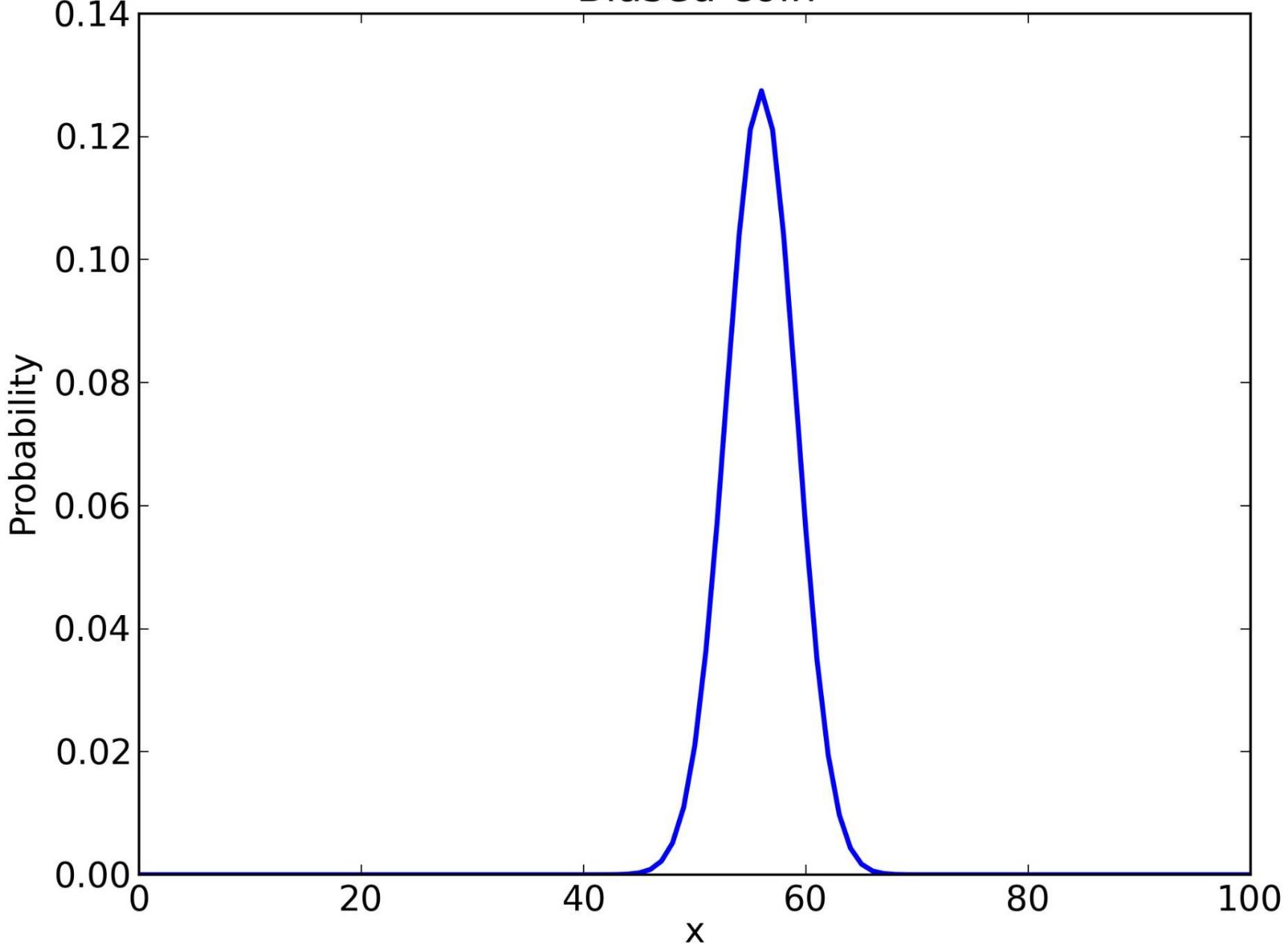


Update

And finally, after 140 heads and 110 tails:

```
evidence = 'H' * 140 + 'T' * 110
for outcome in evidence:
    suite.Update(outcome)
```

Biased coin



Posterior

- Now what?
- How do we summarize the information in the posterior PMF?

Posterior

Given the posterior distribution, what is the probability that x is 50%?

```
print pmf.Prob(50)
```

And the answer is... 0.021

Hmm. Maybe that's not the right question.

Posterior

How about the most likely value of x ?

```
def MLE(pmf):  
    prob, val = max( (prob, val)  
                     for val, prob in  
                     pmf.Items() )  
    return val
```

And the answer is 56%.

Posterior

Or the expected value?

```
def Mean(pmf):  
    total = 0  
    for val, prob in pmf.Items():  
        total += val * prob  
    return total
```

And the answer is 55.95%.

Posterior

The 5th percentile is 51.

```
def Percentile(pmf, p):  
    total = 0  
    for val, prob in sorted(pmf.Items()):  
        total += prob  
        if total >= p:  
            return val
```

Posterior

The 5th percentile is 51.

The 95th percentile is 61.

These values form a 90% credible interval.

So can we say: "There's a 90% chance that x is between 51 and 61?"

Frequentist response



Thank you smbc-comics.com

Bayesian response

Yes, x is a random variable,

Yes, $(51, 61)$ is a 90% credible interval,

Yes, x has a 90% chance of being in it.

However...

The prior is subjective

Remember the prior?

We chose it pretty arbitrarily, and reasonable people might disagree.

Is x as likely to be 1% as 50%?

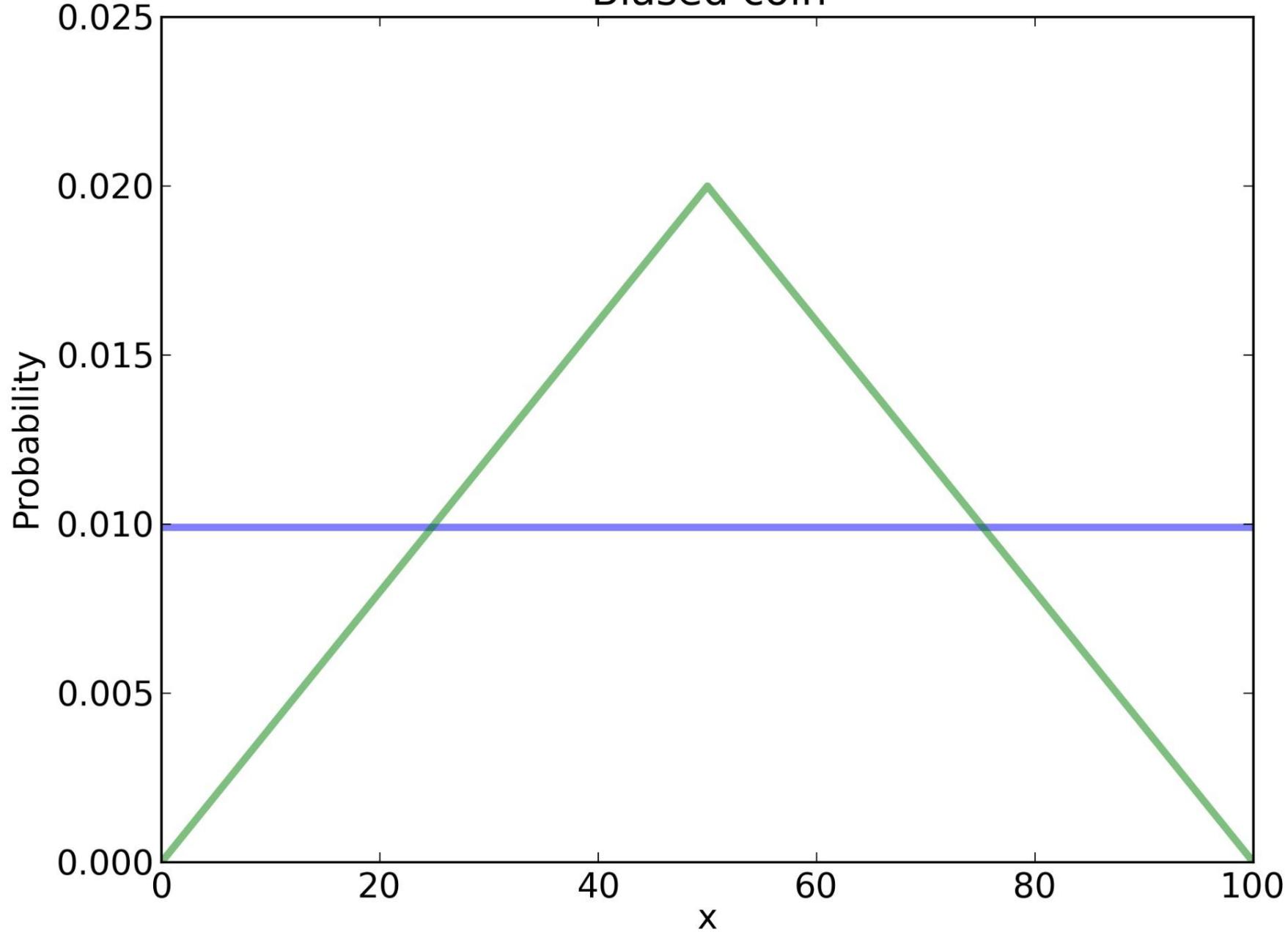
Given what we know about coins, I doubt it.

Prior

How should we capture background knowledge about coins?

Try a triangle prior.

Biased coin

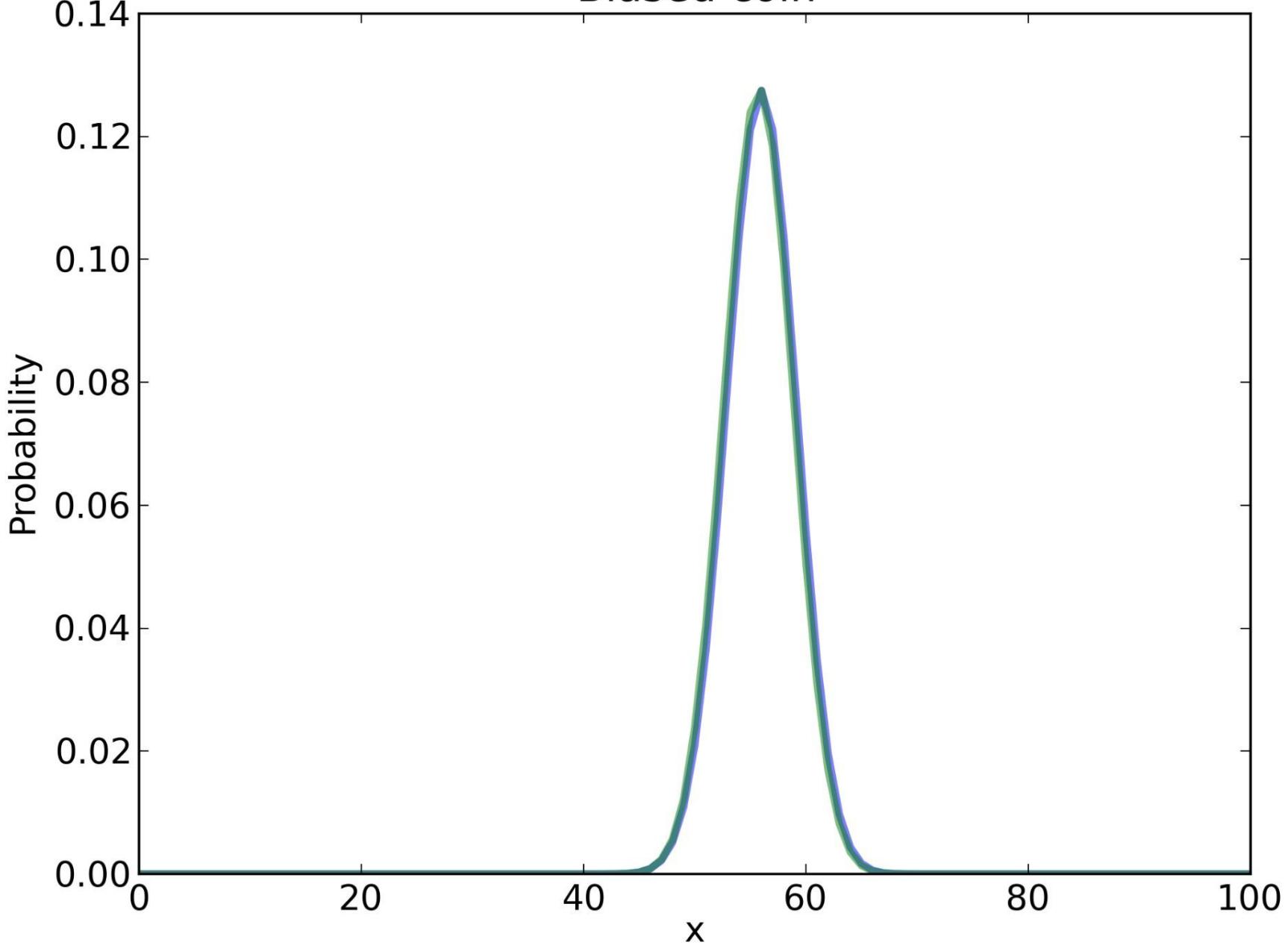


Posterior

What do you think the posterior distribution looks like?

I was going to put an image here, but then I Googled "posterior". Never mind.

Biased coin



Swamp the prior

With enough data,
reasonable people converge.

But if any $p(H_i) = 0$, no data
will change that.

Swamp the prior

Priors can be arbitrarily low,
but avoid 0.

See [wikipedia.org/wiki/
Cromwell's rule](https://en.wikipedia.org/wiki/Cromwell%27s_rule)

*"I beseech you, in the bowels
of Christ, think it possible that
you may be mistaken."*



Summary of estimation

1. Form a suite of hypotheses, H_i .
2. Choose prior distribution, $p(H_i)$.
3. Compute likelihoods, $p(D|H_i)$.
4. Turn off brain.
5. Compute posteriors, $p(H_i|D)$.

Recess!



http://images2.wikia.nocookie.net/_cb20120116013507/recess/images/4/4c/Recess_Pic_for_the_Internet.png

Hypothesis testing

Remember the original question:

"But do these data give evidence that the coin is biased rather than fair?"

What does it mean to say that data give evidence for (or against) a hypothesis?

Hypothesis testing

D is evidence in favor of H if

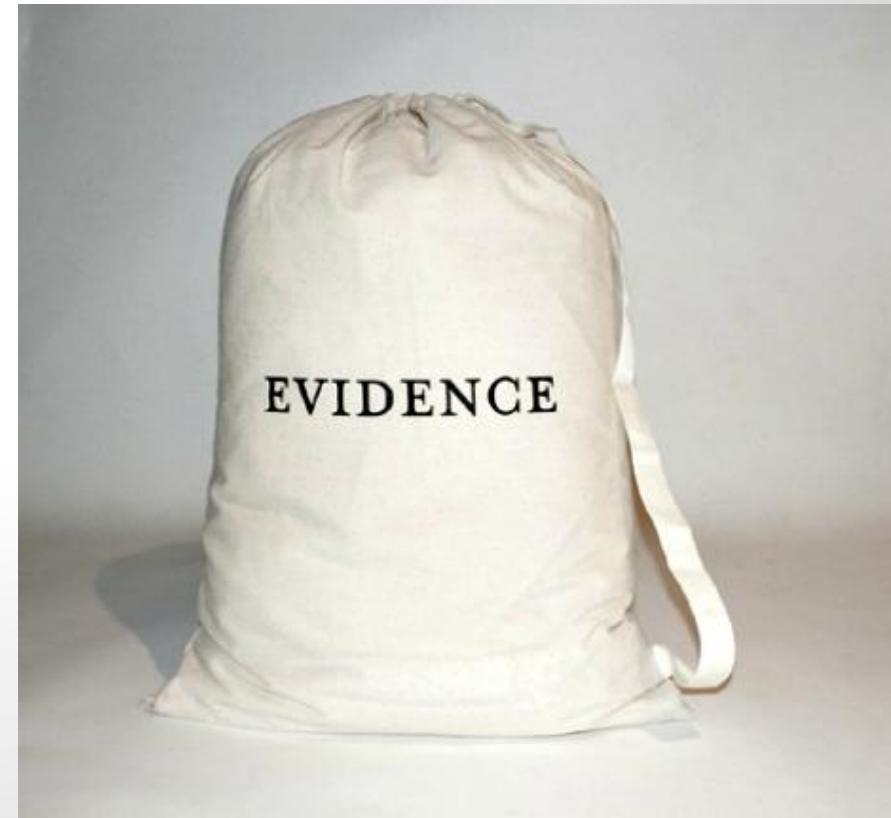
$$p(H|D) > p(H)$$

which is true if

$$p(D|H) > p(D|\sim H)$$

or equivalently if

$$p(D|H) / p(D|\sim H) > 1$$



Hypothesis testing

This term

$$p(D|H) / p(D|\sim H)$$

is called the likelihood ratio, or Bayes factor.

It measures the strength of the evidence.

Hypothesis testing

F: hypothesis that the coin is fair

B: hypothesis that the coin is biased

$p(D|F)$ is easy.

$p(D|B)$ is hard because B is underspecified.

Bogosity

Tempting: we got 140 heads out of 250 spins,
so B is the hypothesis that $x = 140/250$.

But,

1. Doesn't seem right to use the data twice.
2. By this process, almost any data would be evidence in favor of B.

We need some rules

1. You have to choose your hypothesis before you see the data.
2. You can choose a suite of hypotheses, but in that case we average over the suite.



Likelihood

```
def AverageLikelihood(suite, data):  
    total = 0  
  
    for hypo, prob in suite.Items():  
        like = suite.Likelihood(hypo, data)  
        total += prob * like  
  
    return total
```

Hypothesis testing

F: hypothesis that $x = 50\%$.

B: hypothesis that x is not 50% , but might be any other value with equal probability.

Prior

```
fair = Euro()  
fair.Set(50, 1)
```



Prior

```
bias = Euro()  
for x in range(0, 101):  
    if x != 50:  
        bias.Set(x, 1)  
bias.Normalize()
```



Bayes factor

```
data = 140, 110
```

```
like_fair = AverageLikelihood(fair, data)
like_bias = AverageLikelihood(bias, data)
```

```
ratio = like_bias / like_fair
```

Hypothesis testing

Read `euro2.py`.

Notice the new Likelihood function.

Run it and interpret the results.

Hypothesis testing

And the answer is:

$$p(D|B) = 2.6 \cdot 10^{-76}$$

$$p(D|F) = 5.5 \cdot 10^{-76}$$

Likelihood ratio is about 0.47.

So this dataset is evidence **against** B.

Fair comparison?

- Modify the code that builds bias; try out a different definition of B and run again.

Summary

Euro problem,
Bayesian estimation,
Bayesian hypothesis testing.



Recess!



Word problem for geeks

ALICE: What did you get on the math SAT?

BOB: 760

ALICE: Oh, well I got a 780. I guess that means I'm smarter than you.

NARRATOR: Really? What is the probability that Alice is smarter than Bob?

Assume, define, quantify

Assume: each person has some probability, x , of answering a random SAT question correctly.

Define: "Alice is smarter than Bob" means $x_a > x_b$.

How can we quantify $\text{Prob} \{ x_a > x_b \}$?

Be Bayesian

Treat x as a random quantity.

Start with a prior distribution.

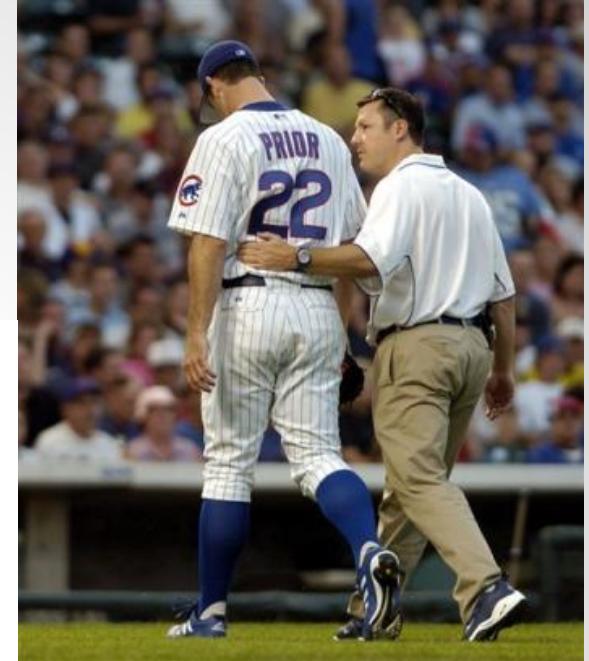
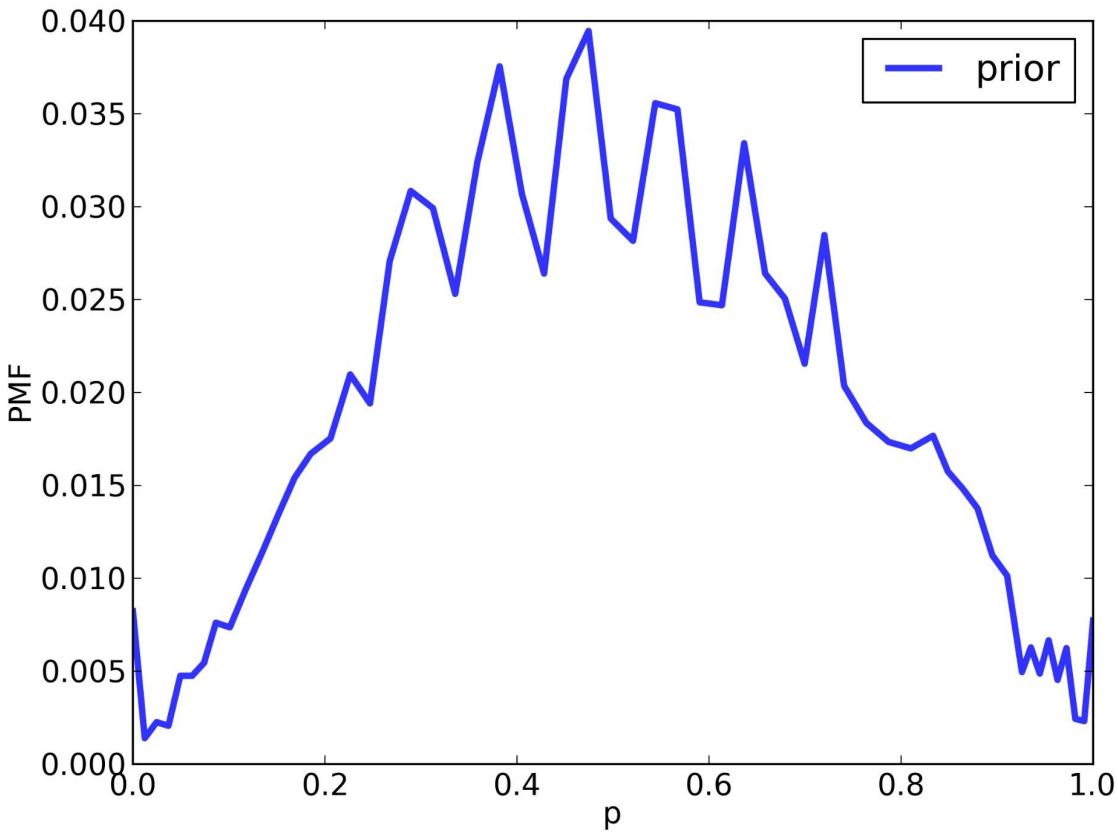
Update it.

Compare posterior distributions.



Prior?

Distribution of raw scores.

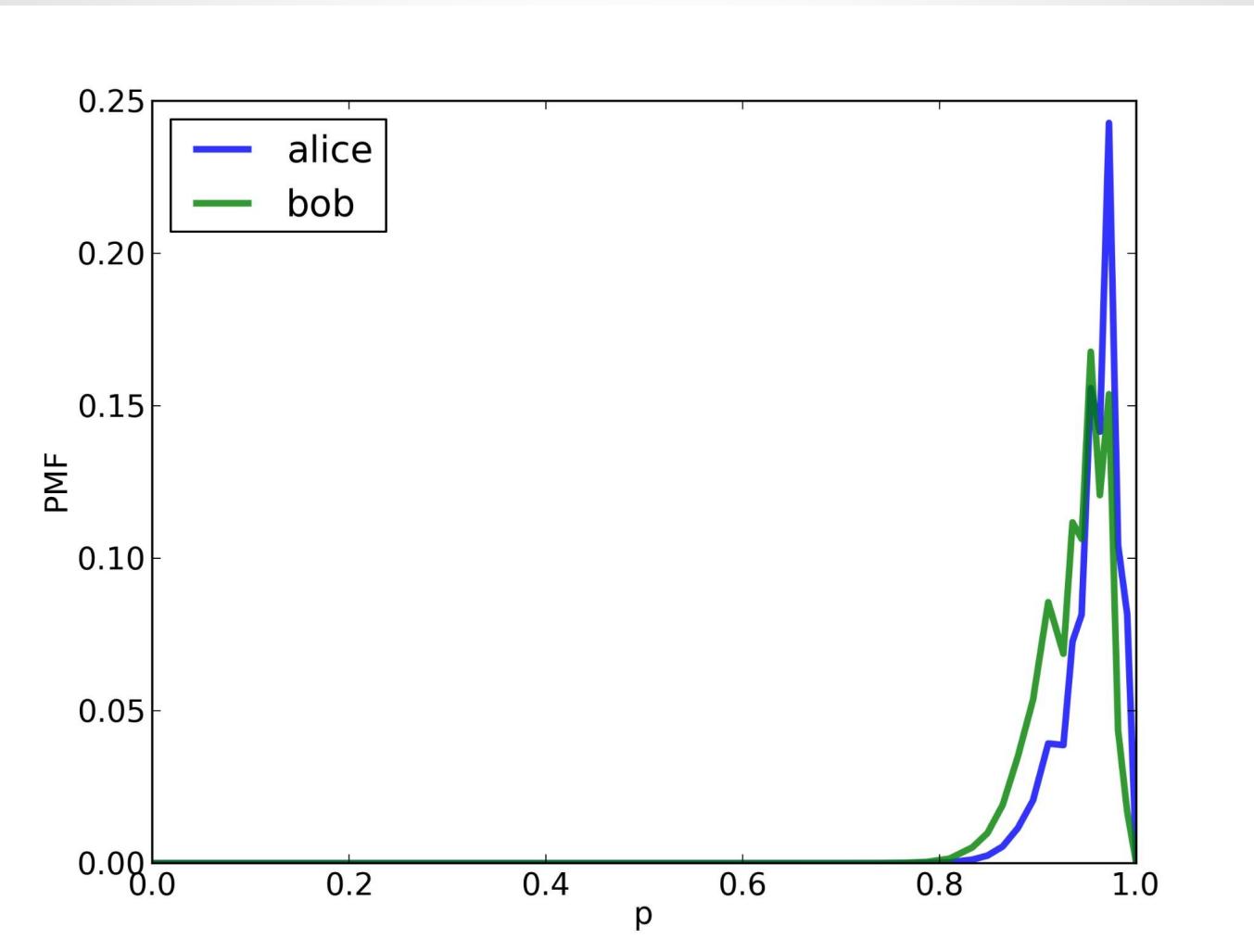


Likelihood

```
def Likelihood(data, hypo):
    x = hypo
    score = data
    raw = self.exam.Reverse(score)

    yes, no = raw, self.exam.max_score - raw
    like = x**yes * (1-x)**no
    return like
```

Posterior



ProbBigger

```
def ProbBigger(pmf1, pmf2):
    """Returns the prob that a value from pmf1
    is greater than a value from pmf2."""
```

ProbBigger

```
def ProbBigger(pmf1, pmf2):
    """Returns the prob that a value from pmf1
    is greater than a value from pmf2."""
```

Iterate through all pairs of values.

Check whether the value from pmf1 is greater.

Add up total probability of successful pairs.

ProbBigger

```
def ProbBigger(pmf1, pmf2):  
  
    for x1, p1 in pmf1.Items():  
        for x2, p2 in pmf2.Items():  
            # FILL THIS IN!
```

ProbBigger

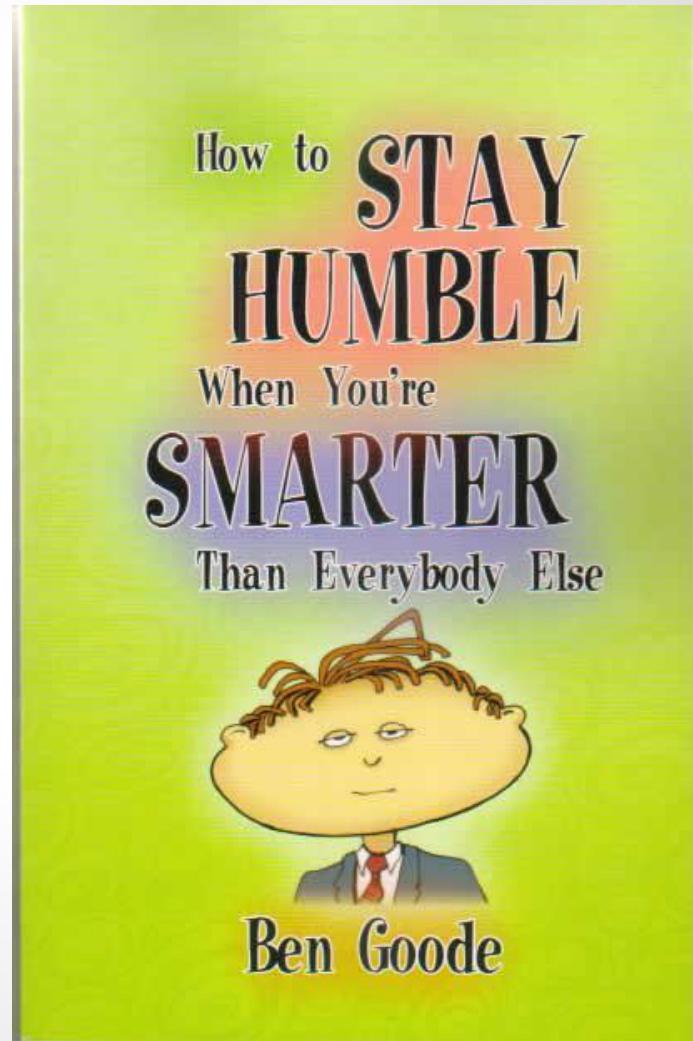
```
def ProbBigger(pmf1, pmf2):  
  
    total = 0.0  
    for x1, p1 in pmf1.Items():  
        for x2, p2 in pmf2.Items():  
            if x1 > x2:  
                total += p1 * p2  
  
    return total
```

And the answer is...

Alice: 780

Bob: 760

Probability that Alice is
"smarter": **61%**



Two points

Posterior distribution is often the input to the next step in an analysis.

Real world problems start (and end!) with modeling.

Modeling

- This result is based on the simplification that all SAT questions are equally difficult.
- An alternative (in the book) is based on item response theory.

Modeling

- For most real world problems, there are several reasonable models.
- The best choice depends on your goals.
- Modeling errors probably dominate.

Modeling

Therefore:

- Don't mistake the map for the territory.
- Don't sweat approximations smaller than modeling errors.
- Iterate.

Recess!



http://images2.wikia.nocookie.net/_cb20120116013507/recess/images/4/4c/Recess_Pic_for_the_Internet.png



Unseen species problem



Unseen species problem

Belly Buttons 2.0

The belly button has captured the human imagination for centuries – it's made us giggle, it's made us blush, it's made us turn our heads in horror (just do a Google images search for belly buttons). It's the portal through which we were connected to our mothers; and it's the physical reminder of our evolutionary past.

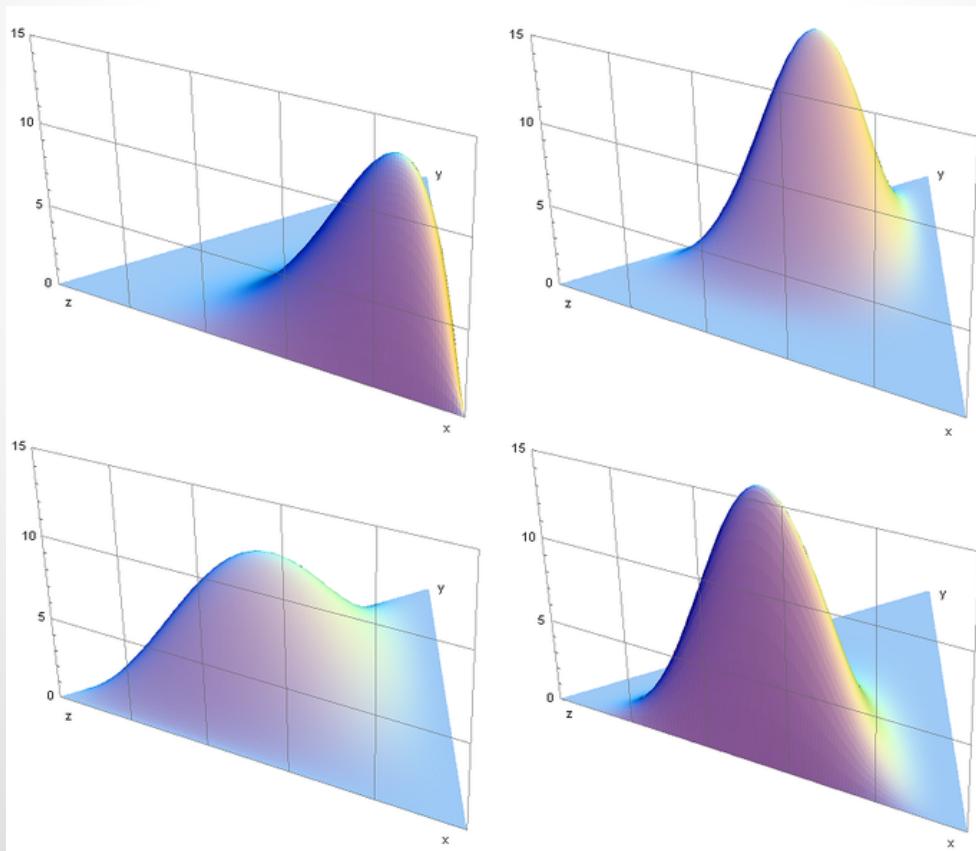
With the Belly Button Biodiversity project, we transformed the meaning of navel-gazing. We recruited over 500 of you to twirl a Q-tip in your belly button for the sake of scientific discovery. We inspired countless more to pause, ponder, and dare we say *appreciate* the microbes that call your body home.

We're moving into a new phase with the project – let's call it **Belly Button 2.0**.



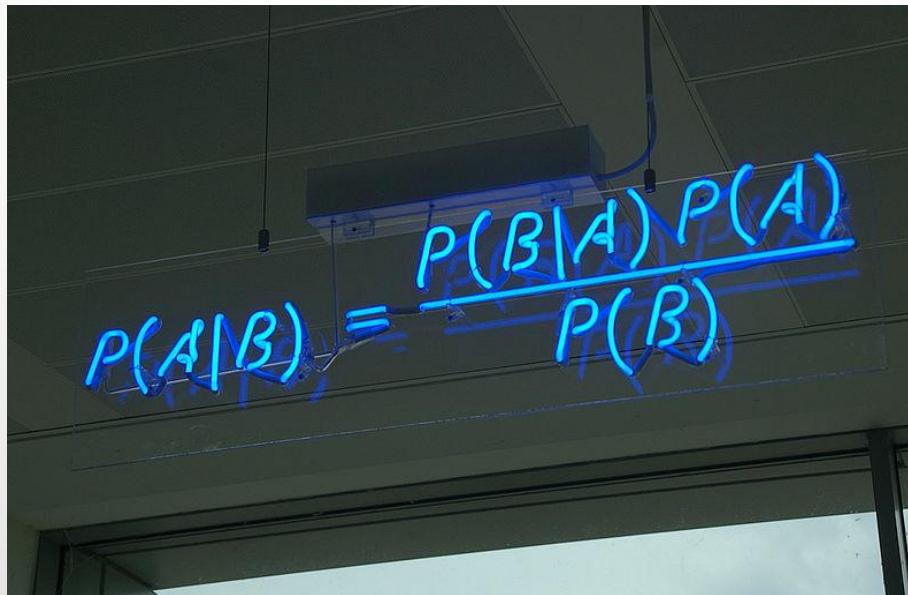
Dirichlet distribution

- If we knew the number of species, we could estimate the prevalence of each.



Hierarchical Bayes

- For a hypothetical value of n , we can compute the likelihood of the data.
- Using Bayes's Theorem, we can get the posterior distribution of n .

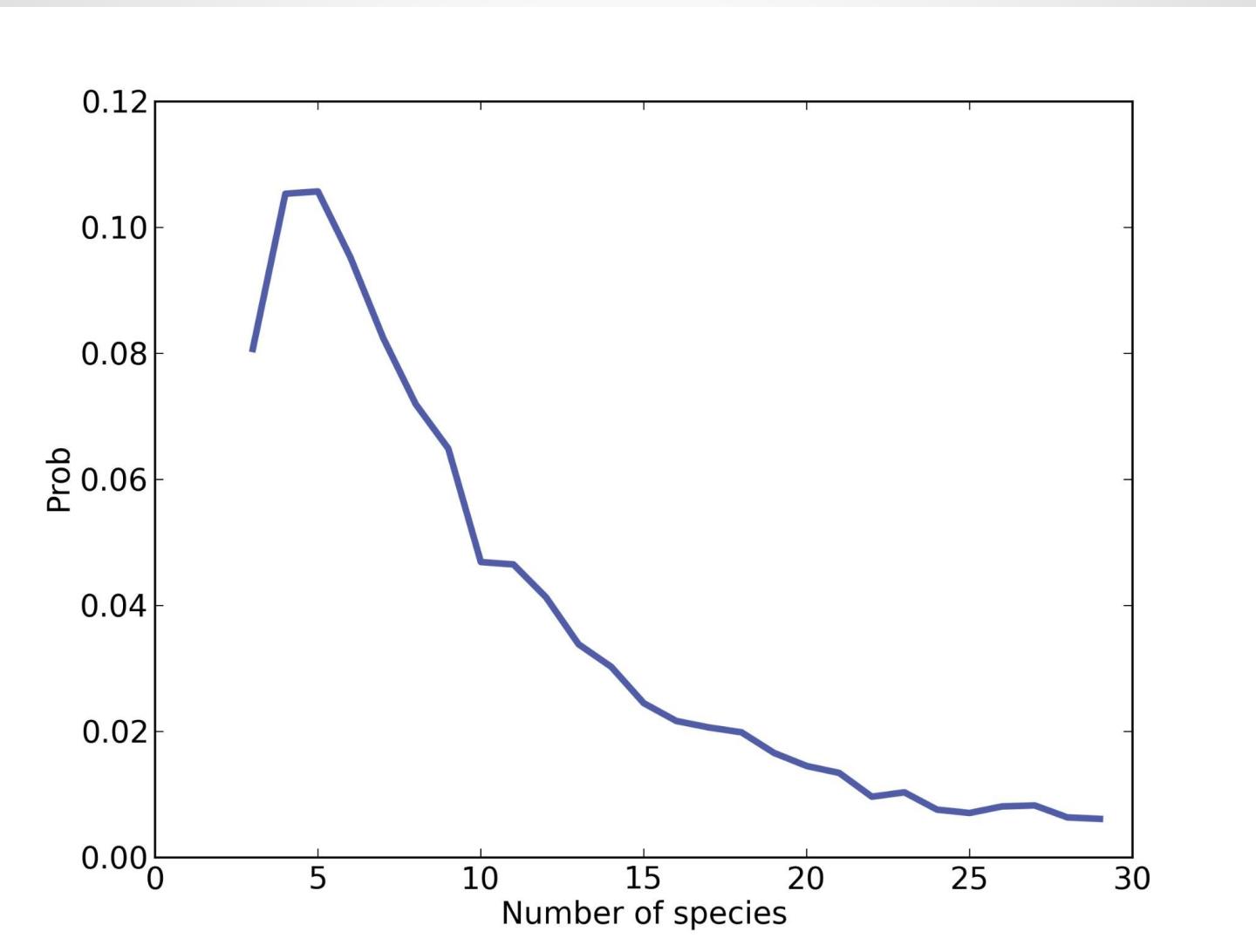

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Lions and tigers and bears

- Suppose we have seen 3 lions and 2 tigers and 1 bear.
- There must be at least 3 species.
- And up to 30?



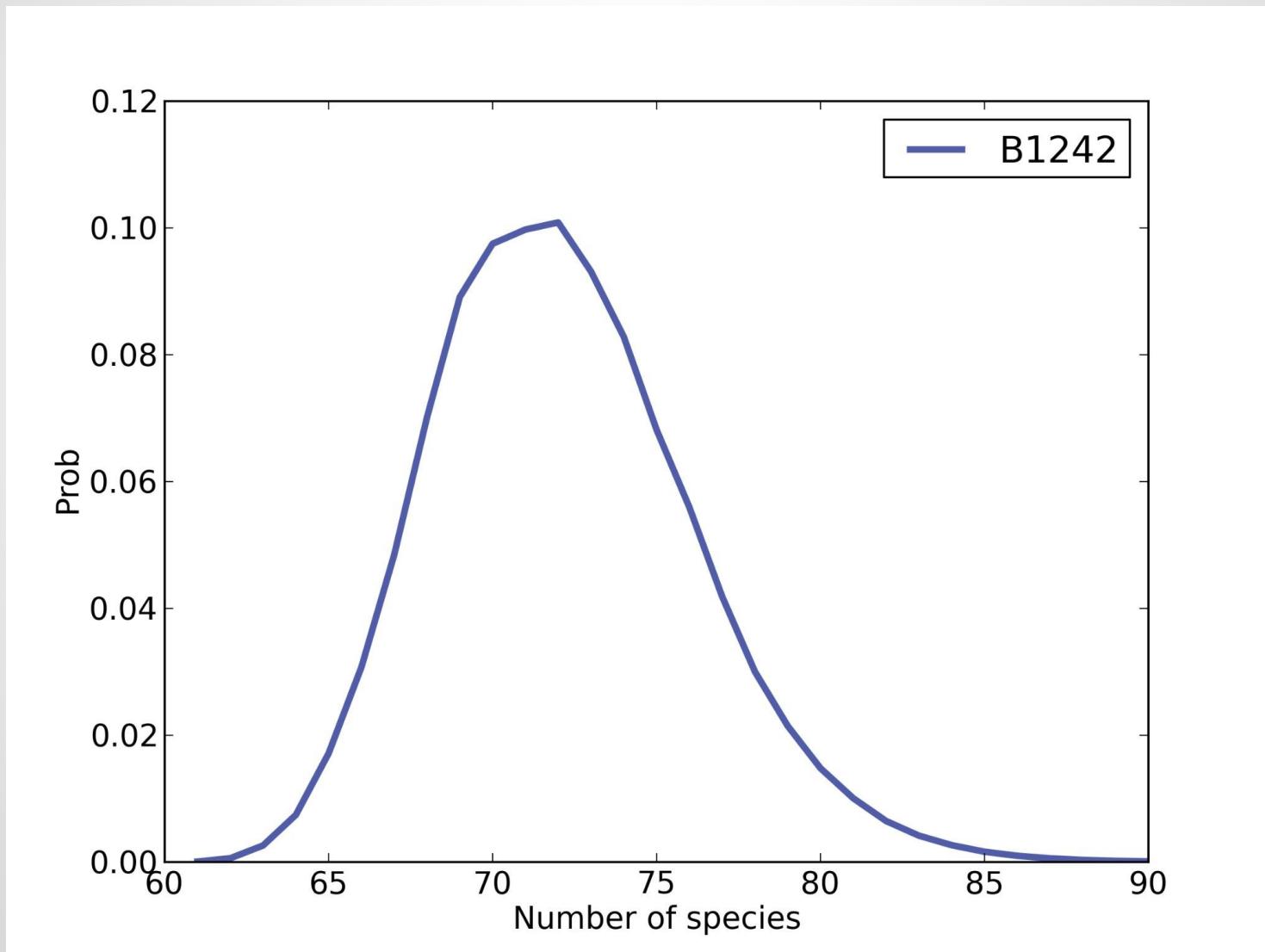
The posterior



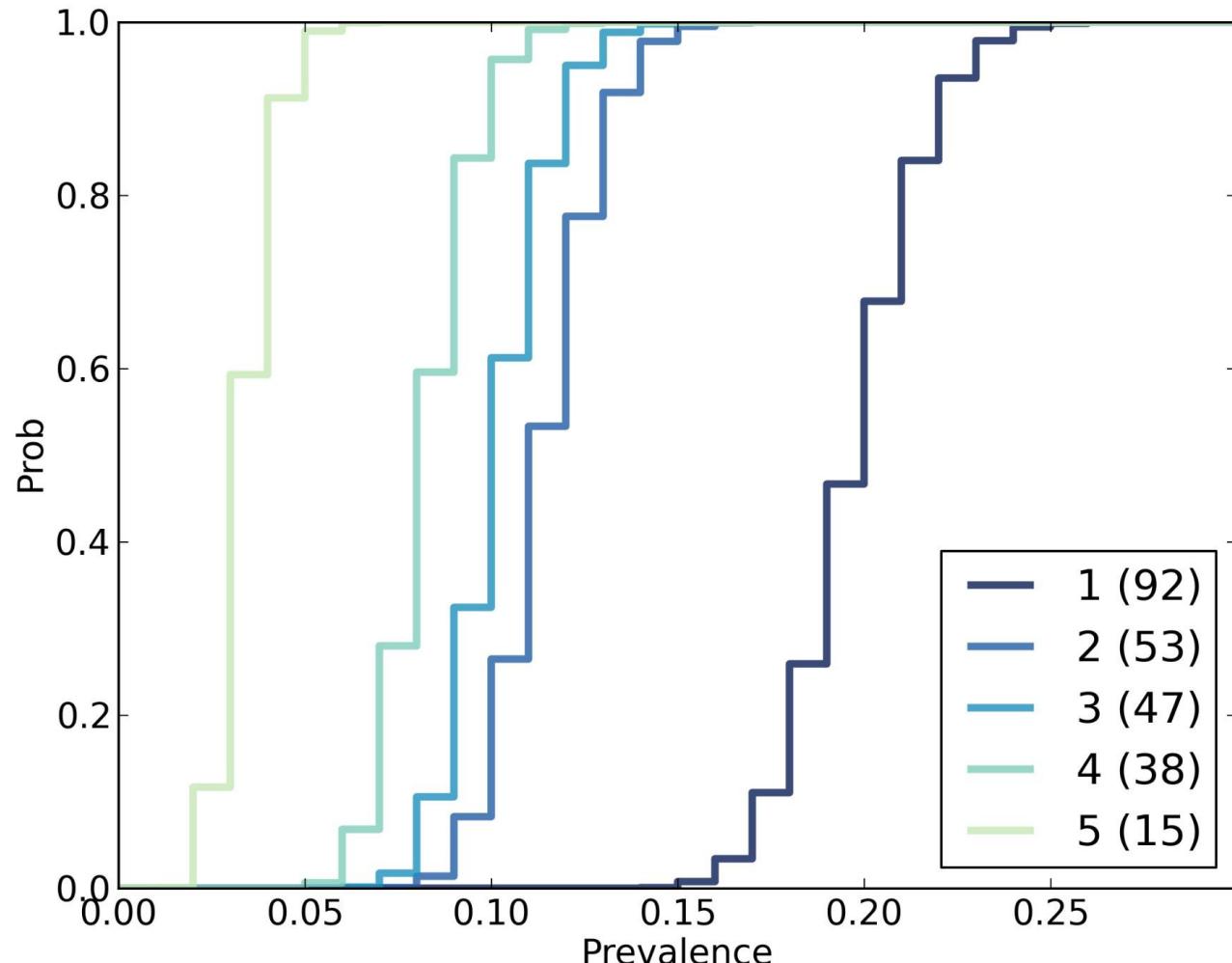
B1242

92 *Corynebacterium*
53 *Anaerococcus*
47 *Finegoldia*
38 *Staphylococcus*
15 *Peptoniphilus*
14 *Anaerococcus*
12 unidentified
10 Clostridiales
8 Lactobacillales
7 *Corynebacterium*
and 1 *Partridgeinnapeartrium*.

Number of species



Estimated prevalences

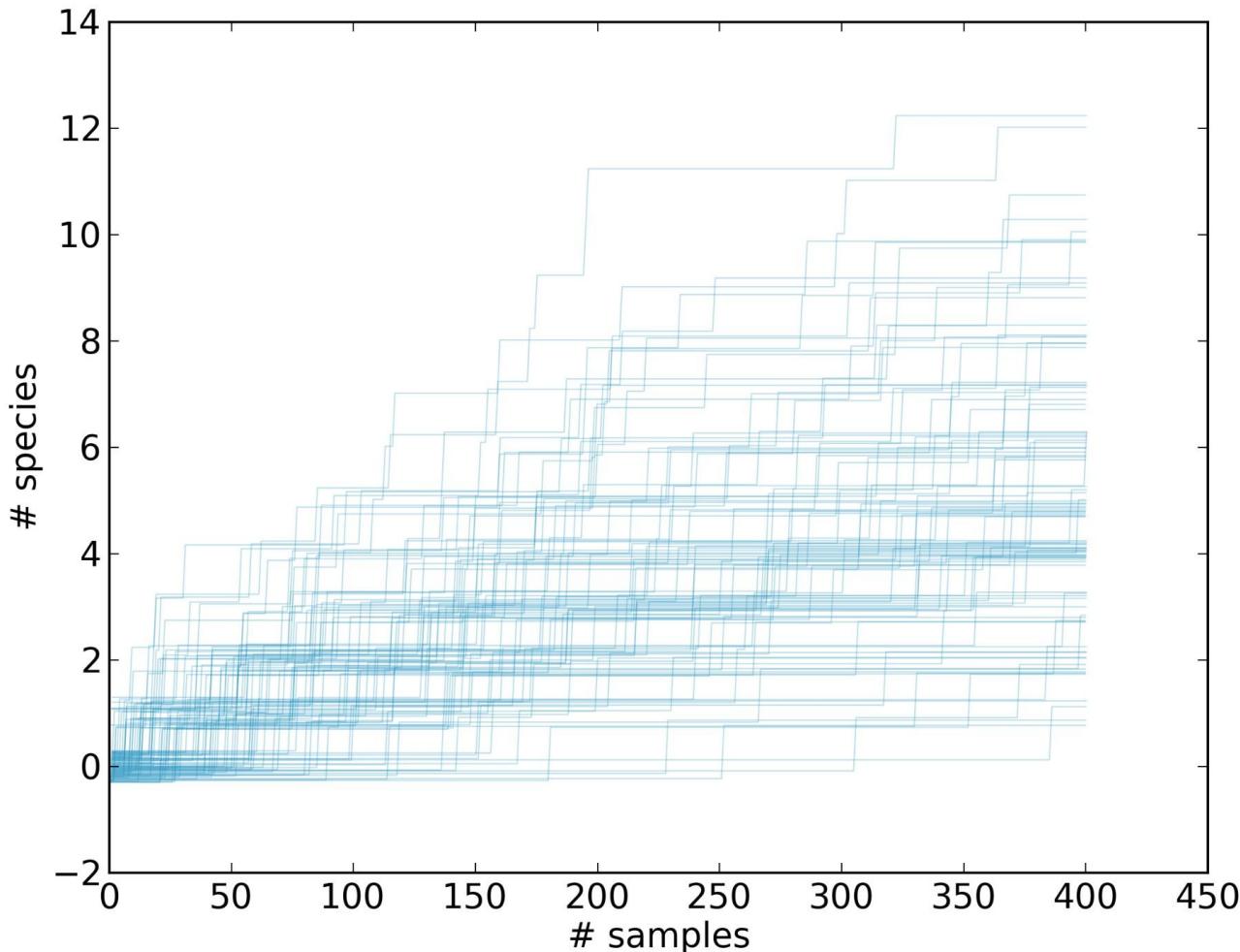


What is this good for?

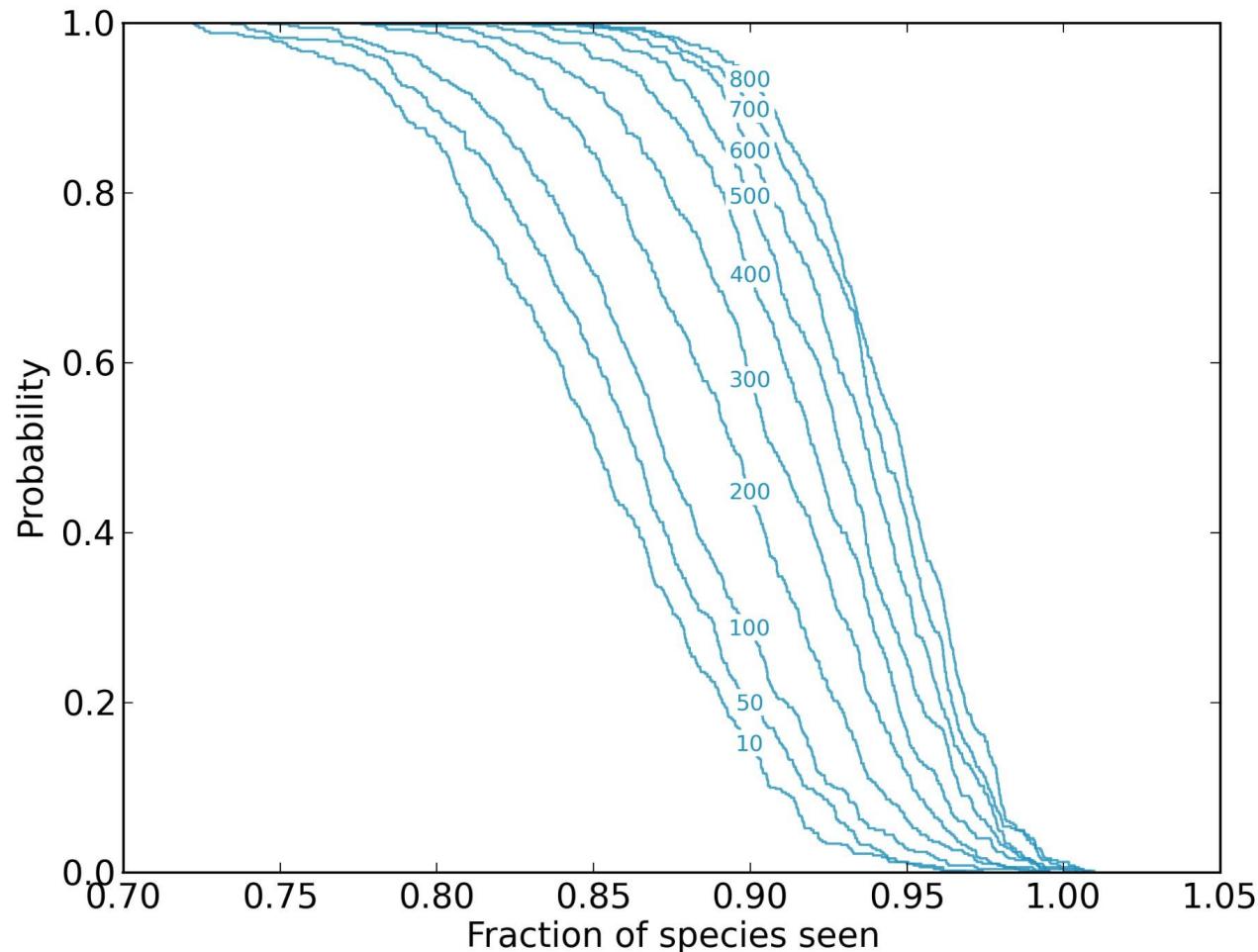
These distributions know *everything*.

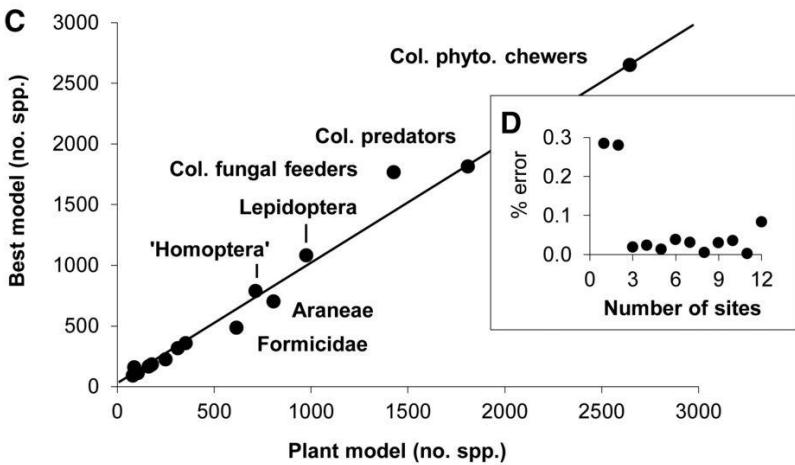
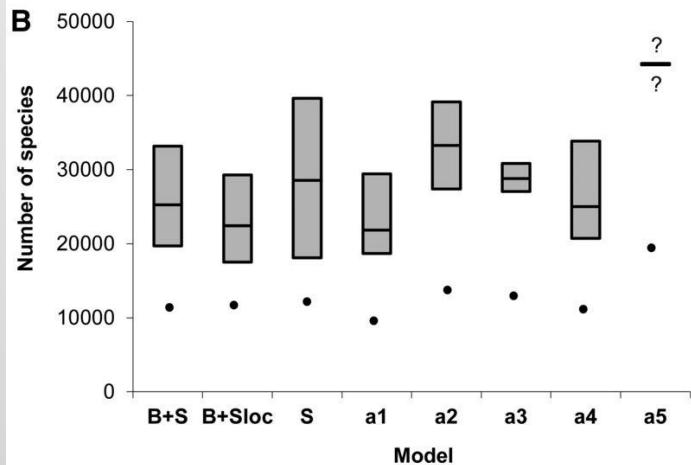
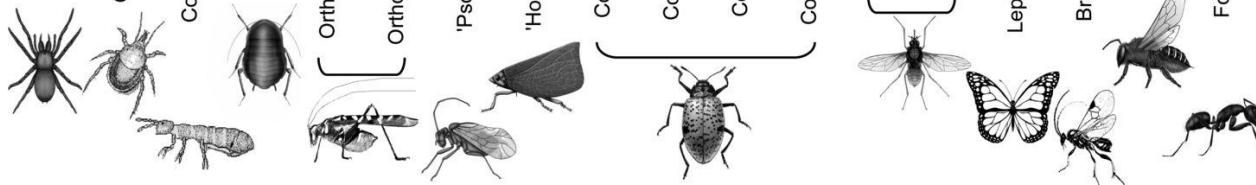
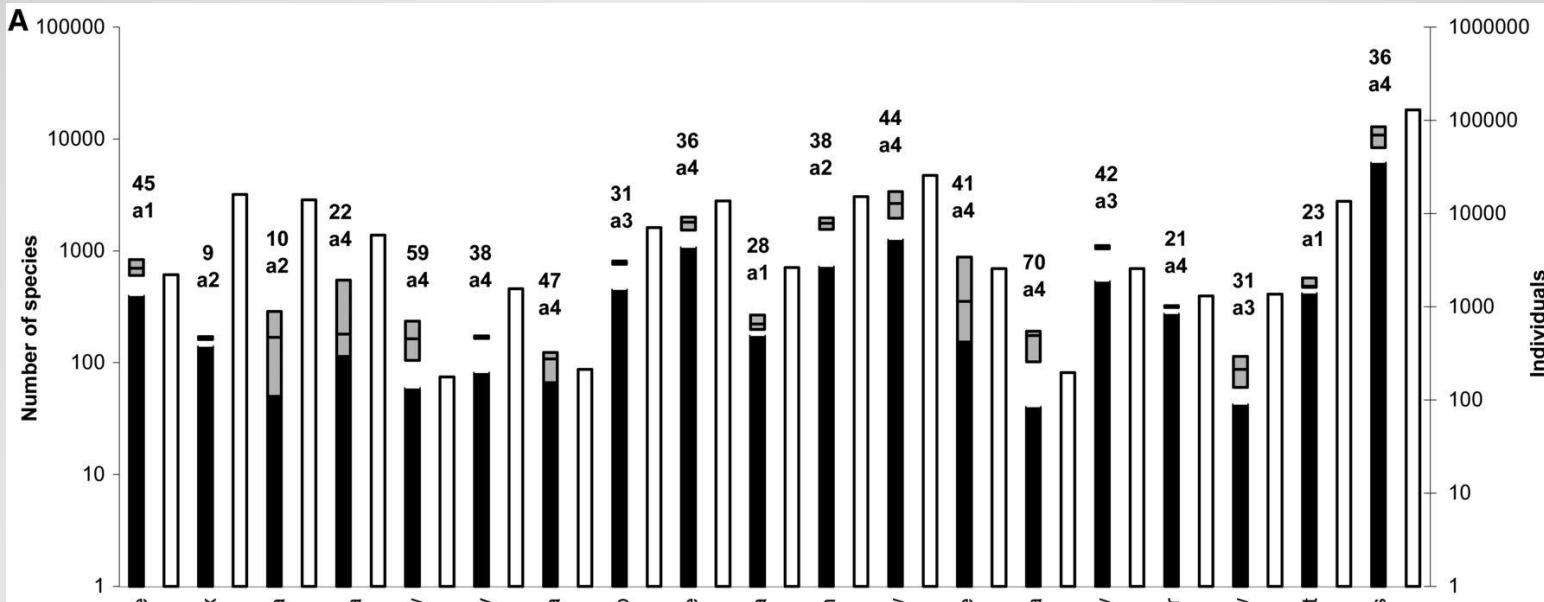
- If I take more samples, how many more species will I see?
- How many samples do I need to reach a given coverage?

More samples, more species



More species, more samples





Shakespeare

436

BRADLEY EFRON AND RONALD THISTED

of approximately 35000 more word types in addition to the 31534 already observed. Our bound involves the theory of empirical Bayes estimation (Robbins, 1956; Good, 1953). It also involves linear programming in both a computational and a theoretical sense. This approach is similar to that taken by Harris (1959). More details are given in an unpublished report of the same title, available from the authors on request.

Table 1. *Shakespeare's word type frequencies*

x	1	2	3	4	5	6	7	8	9	10	Row total
0+	14376	4343	2292	1463	1043	837	638	519	430	364	26305
10+	305	259	242	223	187	181	179	130	127	128	1961
20+	104	105	99	112	93	74	83	76	72	63	881
30+	73	47	56	59	53	45	34	49	45	52	513
40+	49	41	30	35	37	21	41	30	28	19	331
50+	25	19	28	27	31	19	19	22	23	14	227
60+	30	19	21	18	15	10	15	14	11	16	169
70+	13	12	10	16	18	11	8	15	12	7	122
80+	13	12	11	8	10	11	7	12	9	8	101
90+	4	7	6	7	10	10	15	7	7	5	78

Entry x is n_x , the number of word types used exactly x times. There are 846 word types which appear more than 100 times, for a total of 31534 word types.

Almost done!

https://www.surveymonkey.com/s/pycon2013_tutorials

And then some reading suggestions.

More reading

Think Bayes: Bayesian Statistics Made Simple

<http://thinkbayes.com>

It's a draft!

Corrections and suggestions welcome.

Case studies

Think Bayes has 12 chapters.

- Euro problem
- SAT problem
- Unseen species problem
- Hockey problem
- Paintball problem
- Variability hypothesis
- Kidney problem

Case studies

My students are working on case studies:

- Fire (Bayesian regression)
- Collecting whale snot
- Poker
- Bridge
- Rating the raters
- Predicting train arrival times

Case studies

Looking for more case studies.

- Motivated by a real problem.
- Uses the framework.
- Extends the framework?
- 5-10 page report.
- Best reports included in the published version (pending contract).

Need help?

I am always looking for interesting projects.

- Summer 2013.
- Sabbatical 2014-15.



Remember

- The Bayesian approach is a divide and conquer strategy.
- You write Likelihood().
- Bayes does the rest.

More reading

MacKay, *Information Theory, Inference, and Learning Algorithms*

Sivia, *Data Analysis: A Bayesian Tutorial*

Gelman et al, *Bayesian Data Analysis*

Where does this fit?

Usual approach:

- Analytic distributions.
- Math.
- Multidimensional integrals.
- Numerical methods (MCMC).

Where does this fit?

Problem:

- Hard to get started.
- Hard to develop solutions incrementally.
- Hard to develop understanding.

My theory

- Start with non-analytic distributions.
- Use background information to choose meaningful priors.
- Start with brute-force solutions.
- If the results are good enough and fast enough, **stop**.
- Otherwise, optimize (where analysis is one kind of optimization).
- Use your reference implementation for regression testing.