If we want $\mathbb{E} h(X)$ but the maths is too complicated, we can approximate

$$\mathbb{E} h(x) \approx n^{-1} \sum_{i=1}^{n} h(x_i)$$

where $x_1, \ldots, x_n$ are sampled from $X$.

This approximation also tells us how to estimate probabilities, since

$$\mathbb{P}(X \in A) = \mathbb{E} 1_{X \in A}$$

For computational Bayes, we need something a bit fancier: **weighted samples**
probability of heads, unknown

\[ \Theta \sim U[0,1] \]

number of heads from 4 coin tosses

\[ X \sim \text{Bin}(n, \Theta) \]

0. First write out our probability model for the data \( \Pr_X(x|\Theta = \theta) \)

1. Write out \( \Pr_\Theta(\theta) \)

2. Use the formula

\[ \Pr_\Theta(\theta|X = x) = \kappa \Pr_\Theta(\theta) \Pr_X(x|\Theta = \theta) \]

then find \( \kappa \) to make this integrate to 1

... but these are usually intractable

This lets us calculate probabilities:

\[ \mathbb{P}(\Theta \in \text{range}|X = x) = \int_{\Theta \in \text{range}} \Pr_\Theta(\theta|X = x) \, d\theta \]
One way to do **COMPUTATIONAL BAYES**

1. Generate a sample \((\theta_1, ..., \theta_n)\) from \(\Theta\)
2. Compute weights
   \[ w_i = \Pr_X(x|\Theta = \theta_i), \]
   then rescale weights to sum to one

\[
\Pr(\Theta \in \text{range}|X = x) \approx \sum_{i=1}^{n} w_i \delta_{\theta_i \in \text{range}}
\]

It’s more elegant to use the generalized version

\[
\mathbb{E}[h(\Theta)|X = x] \approx \sum_i w_i h(\theta_i)
\]

**ALGEBRAIC BAYES**

0. First write out our probability model for the data \(\Pr_X(x|\Theta = \theta)\)
1. Write out \(\Pr(\theta)\)
2. Use the formula
   \[
   \Pr(\theta|X = x) = \kappa \Pr(\theta) \Pr_X(x|\Theta = \theta)
   \]
   then find \(\kappa\) to make this integrate to 1

This lets us calculate probabilities:

\[
\Pr(\Theta \in \text{range}|X = x) \approx \int_{\Theta \in \text{range}} \Pr(\Theta|X = x) d\Theta
\]

§6.2

... but these are usually intractable
One way to do

**COMPUTATIONAL BAYES**

0. First write out our probability model for the data $\Pr_X(x|\Theta = \theta)$

1. Generate a sample $(\theta_1, ..., \theta_n)$ from $\Theta$

2. Compute weights $w_i = \Pr_X(x|\Theta = \theta_i)$, then rescale weights to sum to one

Reason about $(\Theta|X = x)$ indirectly, using

$$\mathbb{E}[h(\Theta)|X = x] \approx \sum_i w_i h(\theta_i)$$
Example

I got $x = 1$ head out of $n = 4$ coin tosses. I propose the probability model $X \sim \text{Bin}(n, \Theta)$. I don’t know $\Theta$, so I’ll treat it as a random variable, $\Theta \sim U[0,1]$.

Plot the distribution of $(\Theta|X = x)$.

Likelihood of the data:

\[
X \sim \text{Bin}(n, \Theta) \quad \quad \Pr_X(x|\Theta = \Theta) = \binom{n}{x} \Theta^x (1-\Theta)^{n-x}
\]

For $n=4, x=1$:

\[
= 4 \Theta (1-\Theta)^3
\]

Generate a sample $(\theta_1, \ldots, \theta_n)$ from $\Theta$:

\[
\Theta_{\text{samp}} = \text{np.random.uniform}(0,1, \text{size}=1000)
\]

Compute weights $w_i = \Pr_X(x|\Theta = \theta_i)$, then rescale weights to sum to one:

\[
w = 4 * \Theta_{\text{samp}}^1 * (1-\Theta_{\text{samp}})^3 \\
w = w / \text{np.sum}(w)
\]

Reason about $(\Theta|X = x)$ indirectly, using

\[
\mathbb{E}[h(\Theta)|X = x] \approx \Sigma_i w_i h(\theta_i)
\]
Example

I got \( x = 1 \) head out of \( n = 4 \) coin tosses. I propose the probability model \( X \sim \text{Bin}(n, \Theta) \). I don’t know \( \Theta \), so I’ll treat it as a random variable, \( \Theta \sim \mathcal{U}[0,1] \).

Plot the distribution of \( (\Theta \mid X = x) \).

Reason about \( (\Theta \mid X = x) \) indirectly, using

\[
\mathbb{E}[h(\Theta) \mid X = x] \approx \sum_i w_i h(\theta_i)
\]

\[
P(\Theta \in \text{bin} \mid \text{data}) = \mathbb{E}(1_{\Theta \in \text{bin}} \mid \text{data})
\]

\[
= \mathbb{E}(h(\Theta) \mid \text{data}) \quad \text{where } h(\Theta) = 1_{\Theta \in \text{bin}}
\]

\[
\approx \sum_i w_i h(\theta_i) \quad \text{where } \theta_i \text{ sampled from } \Theta \sim \mathcal{U}[0,1]
\]

\[
= \sum_i w_i 1_{\theta_i \in \text{bin}}
\]

\[
= \sum_{i : \theta_i \in \text{bin}} w_i
\]

For each \( \theta \)-bin, let’s show a bar of height

\[
P(\theta \in \text{bin} \mid X = x)
\]

plt.hist(\(\theta\)samp, weights\(w\))

For each bin, sum up the weights of the \( \Theta \)-samples that are in that bin.
For samples of a continuous random variable, I prefer to plot *density histograms*, where the bar heights are rescaled so that the total area is 1.

This makes them directly comparable to a pdf.

```python
plt.hist(θsamp, weights=w, density=True)
```

pdf of Beta(2,4) (which is the posterior distribution we derived mathematically)
Exercise 6.2.1
Consider the probability model

```python
def rxy():
    x = np.random.uniform(-1, 1)
    y = np.random.normal(loc=x**2, scale=0.1)
    return (x,y)
```

Suppose we have observed $Y = 0.2$ and we want to know the likely range of $X$. Plot a histogram of $(X|Y = 0.2)$.

Likelihood of the data:

$$P_Y(0.2 | x = x) = \text{scipy.stats.norm.pdf}(0.2, \text{loc}=x \times x^2, \text{scale}=0.1)$$

Generate a sample $(\theta_1, \ldots, \theta_n)$ from $\Theta$:

$$P_Y(0.2 | x = x_i)$$

Compute weights $w_i = P_X(x|\Theta = \theta_i)$, then rescale weights to sum to one:

```python
xsamp = np.random.uniform(-1, 1, size=10000)

# weight[i] = Pr_Y(0.2 | x=xsamp[i])
w = scipy.stats.norm.pdf(.2, loc=xsamp**2, scale=.1)
w = w / np.sum(w)
plt.hist(xsamp, weights=w, density=True, bins=np.linspace(-1,1,100))
plt.show()
```
Exercise 8.3.2 (Multiple unknowns)

We have a dataset \( x_1, \ldots, x_n \). We propose to model it as independent samples from \( U[A, A + B] \), where \( A \) and \( B \) are unknown parameters.

Using \( A \sim \text{Exp}(0.5) \) and \( B \sim \text{Exp}(1.0) \) as prior distributions for the unknown parameters, find the distribution of \( (B | \text{data}) \).

**Likelihood of the data:**

\[
Pr(x_1, \ldots, x_n | A=a, B=b) = \prod_{i=1}^{n} Pr(x_i | A=a, B=b) \quad \text{since our model says they're independent}
\]

\[
= \prod_{i=1}^{n} \frac{1}{b} \mathbf{1}_{a \leq x_i \leq a+b} \quad \text{the pdf of } U[a, a+b]
\]

\[
= \frac{1}{b^n} \mathbf{1}_{a \leq \min x \leq a+b} \mathbf{1}_{\max x \leq a+b}
\]

\[
(A, B) \quad \{(a_i, b_i), \ldots, (a_n, b_n)\}
\]

**Generate a sample \( (\theta_1, \ldots, \theta_n) \) from \( \Theta \):**

\[
Pr(\text{data} | (A, B)) = (a_i, b_i)
\]

**Compute weights** \( w_i = Pr(x_i | \Theta = \theta_i) \), then rescale weights to sum to one:

\[
x = [2, 3, 2.1, 2.4, 3.14, 1.8]
\]

```python
# Assume that A and B are independent. To generate samples of (A, B) ...
asamp = np.random.exponential(scale=1/0.5, size=1000000)
bsamp = np.random.exponential(scale=1/1.0, size=1000000)
#absamp = zip(asamp, bsamp)

w = 1/bsamp**(len(x)) * np.where((asamp <= min(x)) & (max(x) <= asamp+bsamp), 1, 0)
w = w / np.sum(w)

plt.hist(bsamp, weights=w, density=True, bins=np.linspace(0, 100))
plt.show()
```
Exercise 8.3.2 (Multiple unknowns)

We have a dataset $[x_1, \ldots, x_n]$. We propose to model it as independent samples from $U[1, A + B]$, where $A$ and $B$ are unknown parameters.

Using $A \sim \text{Exp}(0.5)$ and $B \sim \text{Exp}(1.0)$ as prior distributions for the unknown parameters, find the distribution of $(B | \text{data})$.

**TIP.** First find the joint posterior distribution for all the unknown parameters. Then, pick out just the one you’re interested in.

We call this *marginalization*.

**TIP.** If $n$ is large, you can run into underflow problems if you compute $\Pr(x_1, \ldots, x_n | \text{params})$ directly.

Be clever about rescaling the weights, using the log-sum-exp trick (exercise 8.3.4).
Why does computational Bayes work?

\( \Theta \rightarrow X \)

- non-uniform distribution
- \( \sim N(\Theta^2, 0.1^2) \)

Joint pdf

\[
Pr_{\Theta, X}(\theta, x) = Pr_{\Theta}(\theta) \cdot Pr_X(x|\Theta = \theta)
\]

Pr\(_\Theta(\theta|X = x) \propto Pr_{\Theta, X}(\theta, x) \propto Pr_{\Theta}(\theta) \cdot Pr_X(x|\Theta = \theta)

Pr\(_\Theta(\theta|X = 0.2)

samples from \( \Theta \)

num. samples near \( \theta \) \( \propto Pr_{\Theta}(\theta) \)

weighted samples

weight \( w_i = Pr_X(x|\Theta = \theta_i) \)

density hist. of \( (\Theta|X = 0.2) \)

sum up the weights in each bin

bin height at \( \theta \) \( \propto \) num. samples \( \times \) weights

\( \propto Pr_{\Theta}(\theta) \times Pr_X(x|\Theta = \theta) \)
Bayes’s rule for random variables

\[ \Pr_X(x | Y = y) = \Pr_X(x) \frac{\Pr_Y(y | X = x)}{\Pr_Y(y)} \]

\[ \mathbb{P}(X \in A | Y = y) \approx \sum_{i=1}^{n} w_i 1_{x_i \in A} \]

Bayesianism
When there’s an unknown parameter, you should express your uncertainty about it by treating it as a random variable.
Isn’t it crazy to take the unknown parameter to be a random variable? Would a physicist be prepared to say “Let the speed of light be a random variable?” No!

THOUGHT EXPERIMENT. If I draw a card, and ask you “What’s the probably of Hearts”, you’ll likely answer $\frac{1}{4}$. You’ll give this answer even if I can see the card. In other words, you’re treating it as random even though the value is known. You’re using randomness to express your uncertainty.

When we create a probability model, we’re not claiming that its randomness is a true reflection of the actual physical world. (The actual physical world does have randomness, via Schroedinger’s equation, but no sane data modeller would ever use that as their randomness.) When we model a coin as $\text{Bin}(1, \theta)$ that’s not meant to express the underlying physical reality. It’s just a mental construct – it’s all in our heads.