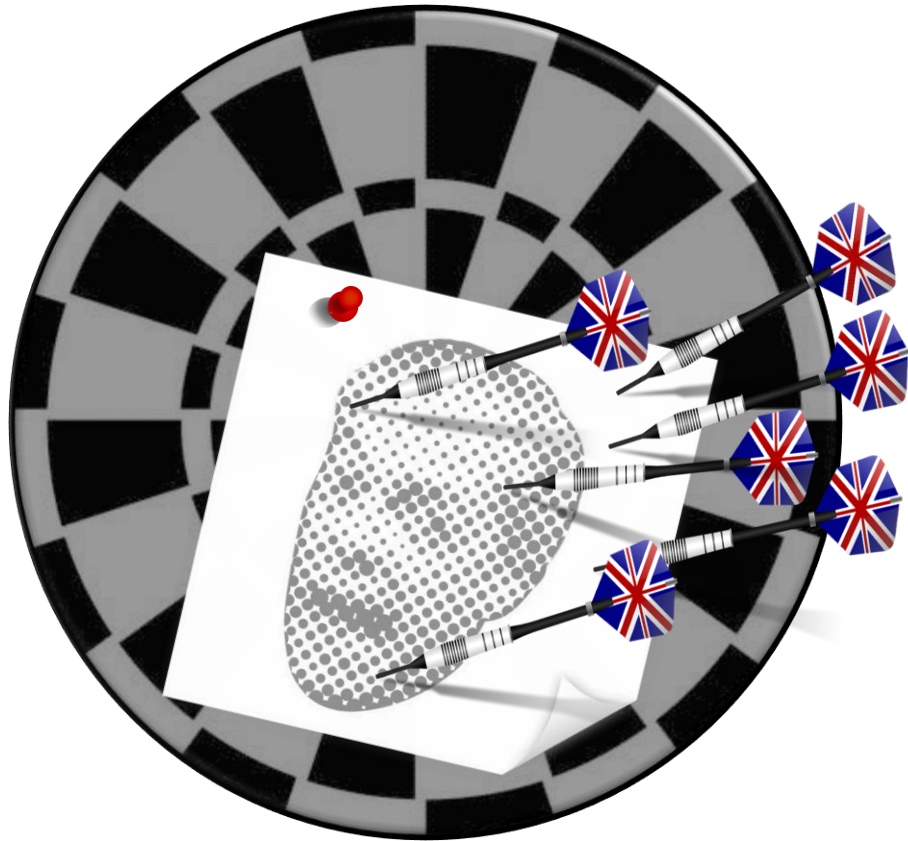


§6. Computational methods

What's the chance that a randomly thrown dart will hit the mystery object A ?



Let X be the location of a randomly thrown dart, and let $x_1, \dots, x_n$ be some throws.

The probability of hitting A is

$$\mathbb{P}(X \in A) \approx \frac{1}{n} \sum_{i=1}^n 1_{x_i \in A} \quad 1_{x \in A} = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{else} \end{cases}$$

```
1 # Let  $X \sim N(\mu = 1, \sigma = 3)$ . What is  $\mathbb{P}(X > 5)$ ?  
2 x = np.random.normal(loc=1, scale=3, size=10000)  
3 i = (x > 5)  
4 np.mean(i)
```

x is a vector of reals, length 10000
 $x > 5$ is a vector of booleans, len. 10000
typecast bool $\rightarrow$ number; true $\rightarrow$ 1, false $\rightarrow$ 0
and compute the average: fraction of samples that are > 5 .

Expectation

For a real-valued random variable X

$$\mathbb{E}X = \begin{cases} \sum_x x \Pr_X(x), & \text{if } X \text{ is discrete} \\ \int_x x \Pr_X(x) dx, & \text{if } X \text{ is continuous} \end{cases}$$

Law of the Unconscious Statistician

For a random variable X and a real-valued function h

$$\mathbb{E}h(X) = \begin{cases} \sum_x h(x) \Pr_X(x), & \text{if } X \text{ is discrete} \\ \int_x h(x) \Pr_X(x) dx, & \text{if } X \text{ is continuous} \end{cases}$$

If we want to know the average properties of a rich random variable (e.g. the average length of a random tweet), we have to use real-valued property readout functions $h(X)$ so that we can take averages.

Monte Carlo integration

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

where $x_1, \dots, x_n$ is a sample drawn from X

Let $h(x) = 1_{x \in A}$

By Monte Carlo

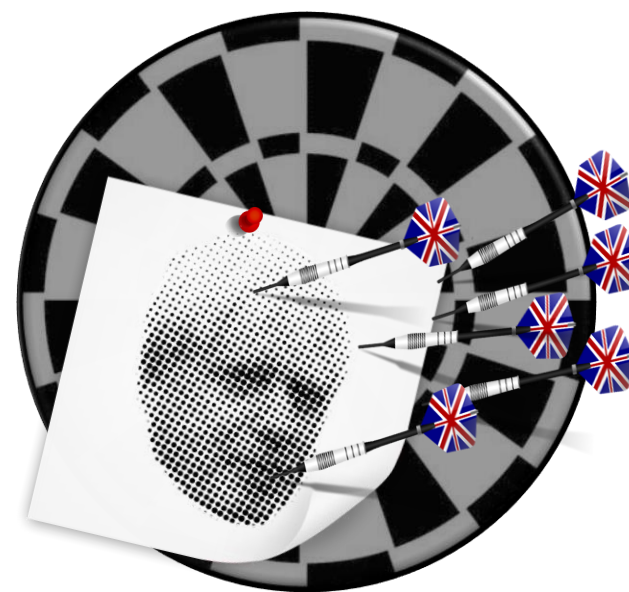
$$\mathbb{E} h(X) \approx \frac{1}{n} \sum_{i=1}^n h(x_i)$$

$$\frac{1}{n} \sum_i 1_{x_i \in A}$$

Let X be the location of a randomly thrown dart, and let $x_1, \dots, x_n$ be some throws. *sampled from X .*

The probability of hitting A is

$$\mathbb{P}(X \in A) \approx \frac{1}{n} \sum_{i=1}^n 1_{x_i \in A}$$



Let $Y = h(X)$
By defn of $\mathbb{E}Y$,

$$\mathbb{E}Y = 0 \times \mathbb{P}(Y=0) + 1 \times \mathbb{P}(Y=1)$$

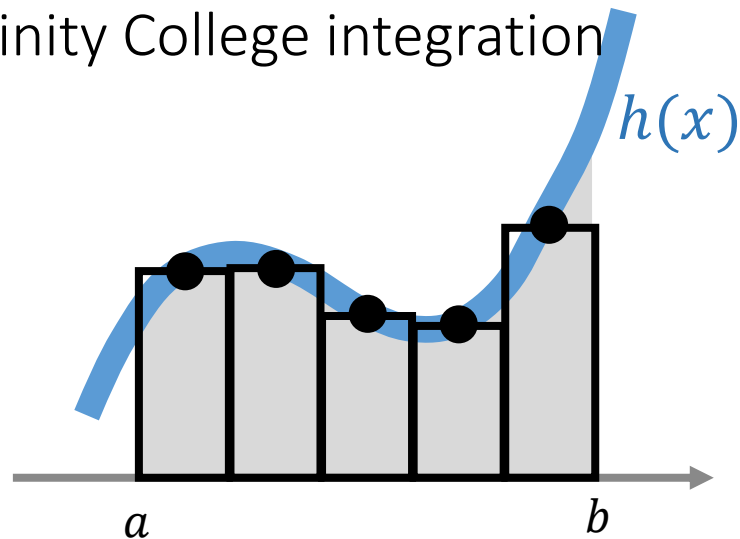
$$= \mathbb{P}(Y=1)$$

$$= \mathbb{P}(h(X)=1)$$

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

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

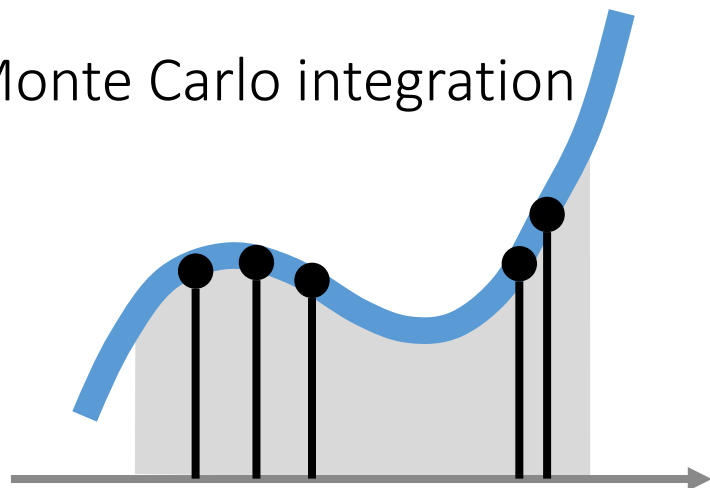
Trinity College integration



$$\int_{x=a}^b h(x) dx \approx \sum_{i=1}^n h(x_i) \frac{b-a}{n}$$

where x_i is the midpoint of interval i

Monte Carlo integration



Let's instead approximate this integral using Monte Carlo. Let $X \sim U[a, b]$.

By Monte Carlo,

$$\underbrace{\mathbb{E}h(X)}_{\downarrow} \approx \frac{1}{n} \sum_{i=1}^n h(x_i) \quad \text{where } x_1, \dots, x_n \text{ sampled from } X$$

$$\int_{x=a}^b h(x) \Pr_X(x) dx = \int_{x=a}^b h(x) \frac{1}{b-a} dx$$

Thus,

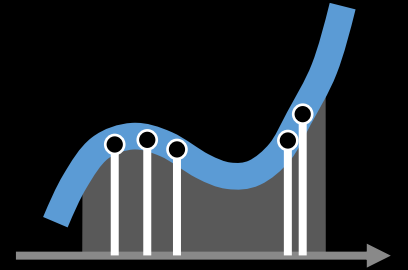
$$\int_{x=a}^b h(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n h(x_i)$$

COMPUTATIONAL METHODS

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

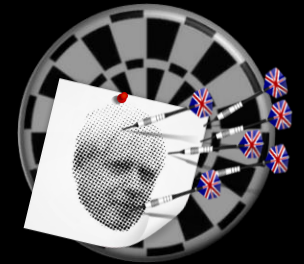
$$\mathbb{E}h(X) \approx n^{-1} \sum_{i=1}^n h(x_i)$$

where $x_1, \dots, x_n$ are sampled from X

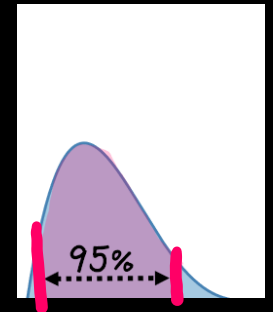


- ❖ Estimating probabilities is just a special case, since

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



- ❖ In Bayesian analysis, our aim is to estimate probabilities, e.g. confidence intervals. How can we do this computationally?



Recall the Bayesian update procedure:

prior belief + data → posterior belief
 $\Pr_{\Theta}(\theta)$ x $\Pr_{\Theta}(\theta|X = x) \propto \Pr_{\Theta}(\theta) \Pr_X(x|\Theta = \theta)$

Θ unknown parameter
 $\downarrow$
 x readout $X=x$

ALGEBRAIC BAYES

1. Write out $\Pr_{\Theta}(\theta)$ *Step 0. Write out the data likelihood, $\Pr_X(x|\Theta=\theta)$*

2. Use the formula
 $\Pr_{\Theta}(\theta|X = x) = \kappa \Pr_{\Theta}(\theta) \Pr_X(x|\Theta = \theta)$
 then find κ to make this integrate to 1

... but these are usually intractable

This lets us calculate confidence intervals:

$$\mathbb{P}(\Theta \in \text{range} | X = x) = \int_{\theta \in \text{range}} \Pr_{\Theta}(\theta | X = x) d\theta$$

One way to do

COMPUTATIONAL BAYES

1. Generate a sample $(\theta_1, \dots, \theta_n)$ from Θ
2. Compute a weight for each sampled θ_i
 $w_i = \Pr_X(x|\Theta = \theta_i)$,
 then rescale weights to sum to one

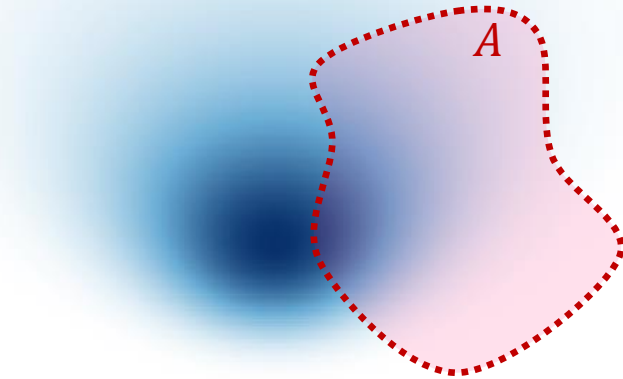
$$\mathbb{P}(\Theta \in \text{range} | X = x) \approx \sum_{i=1}^n w_i 1_{\theta_i \in \text{range}}$$

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

A heatmap, showing the posterior density $\Pr_{\Theta|\text{data}}(\theta)$ for a two-dimensional parameter $\Theta \in \mathbb{R}^2$



posterior
probability

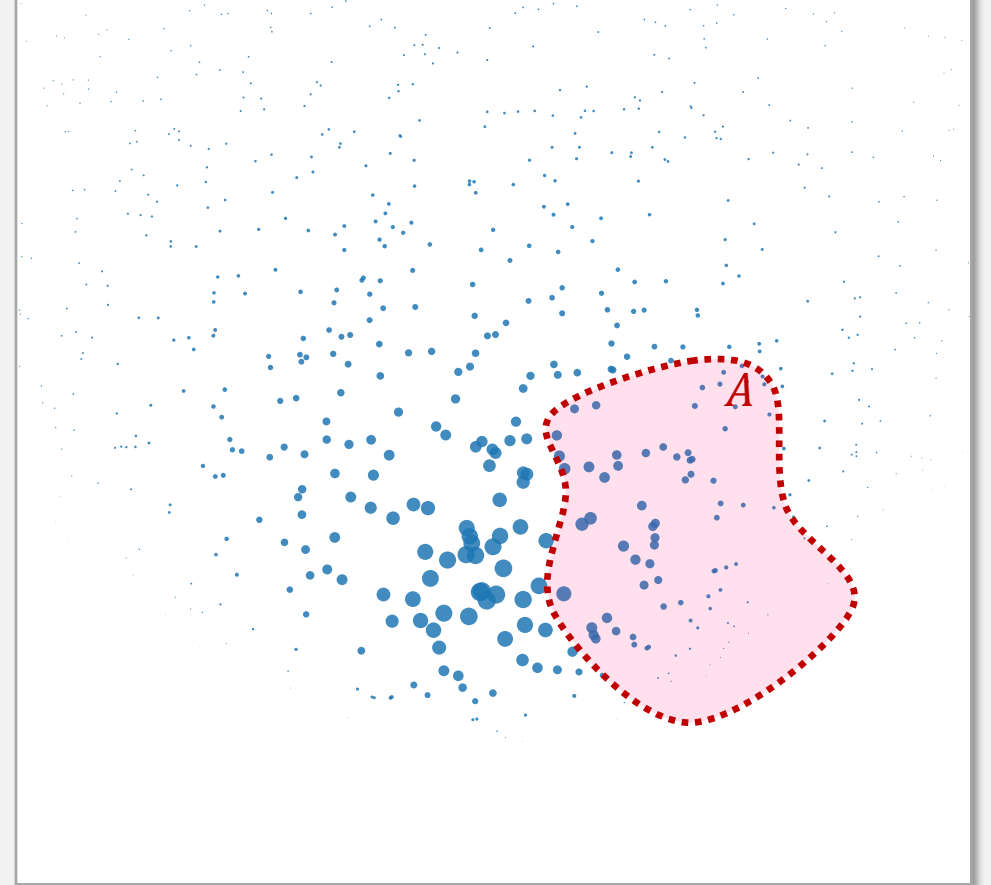
$$\mathbb{P}(\Theta \in A|\text{data}) = \int_{\theta \in A} \Pr_{\Theta|\text{data}}(\theta) d\theta$$

posterior
expectation

$$\mathbb{E}(h(\Theta)|\text{data}) = \int_{\theta} h(\theta) \Pr_{\Theta|\text{data}}(\theta) d\theta$$

COMPUTATIONAL BAYES

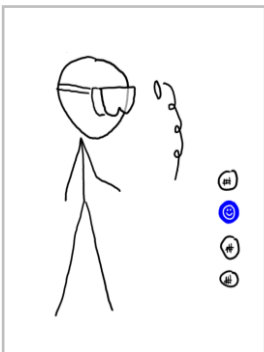
A weighted sample, $w_i \propto \Pr(\text{data}|\Theta = \theta_i)$



$$\mathbb{P}(\Theta \in A|\text{data}) \approx \sum_i w_i 1_{\theta_i \in A}$$

$$\mathbb{E}(h(\Theta)|\text{data}) \approx \sum_i w_i h(\theta_i)$$

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 θ , so I'll treat it as a random variable, $\theta \sim U[0,1]$.

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

$$\theta \sim U[0,1]$$



$$X \sim \text{Bin}(n, \theta)$$

observe
 $X = x$

Likelihood of the data:

$$\Pr_X(x | \theta = \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x} = 4 \theta (1-\theta)^3$$

1. Generate a sample $(\theta_1, \dots, \theta_n)$ from θ :

$$\theta_{\text{samp}} = \text{np.random.uniform}(0,1, \text{size} = 100000)$$

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

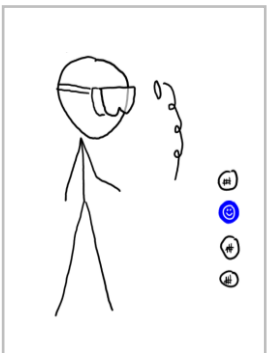
$$w = 4 * (\theta_{\text{samp}}) * ((1 - \theta_{\text{samp}}) ** 3)$$

$$w = w / \text{np.sum}(w)$$

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

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

Example

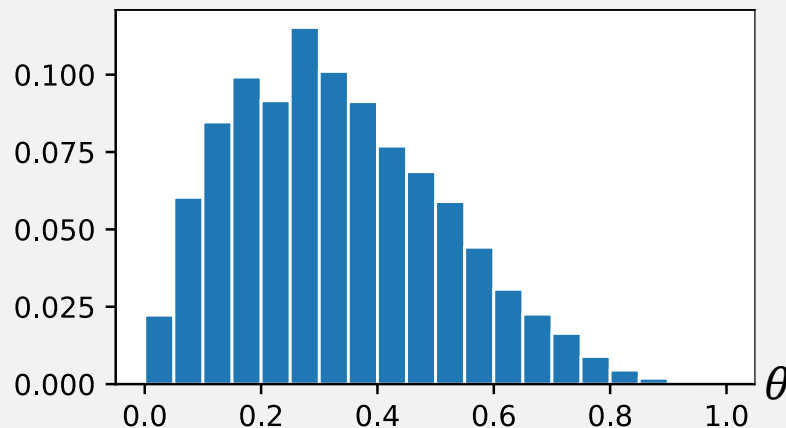


I got $x = 1$ head out of $n = 4$ coin tosses. I propose the probability model $X \sim \text{Bin}(n, \theta)$. I don't know θ , so I'll treat it as a random variable, $\theta \sim U[0,1]$.

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

Reason about $(\theta | X = x)$ indirectly, using
 $\mathbb{E}[h(\theta) | X = x] \approx \sum_i w_i h(\theta_i)$

Let's plot a histogram:
 we'll split the range of θ into bins,
 and for each θ -bin, we want a bar
 of height $\mathbb{P}(\theta \in \text{bin} | X = x)$



$$\mathbb{P}(\theta \in \text{bin} | \text{data})$$

$$= \mathbb{E}(1_{\theta \in \text{bin}} | \text{data}) = \mathbb{E}[h(\theta) | \text{data}] \quad \text{where } h(\theta) = 1_{\theta \in \text{bin}}$$

using the indicator/probability trick

$$\approx \sum_i w_i h(\theta_i)$$

by computational Bayes approximation

$$= \sum_i w_i 1_{\theta_i \in \text{bin}} = \sum_{i: \theta_i \in \text{bin}} w_i$$

unwrapping the indicator function

In code, to plot all bins: `plt.hist(theta_samp, weights=w)`

For continuous random variables, I prefer to scale the bar heights so that the total *area* is 1. This is known as a *density histogram*.

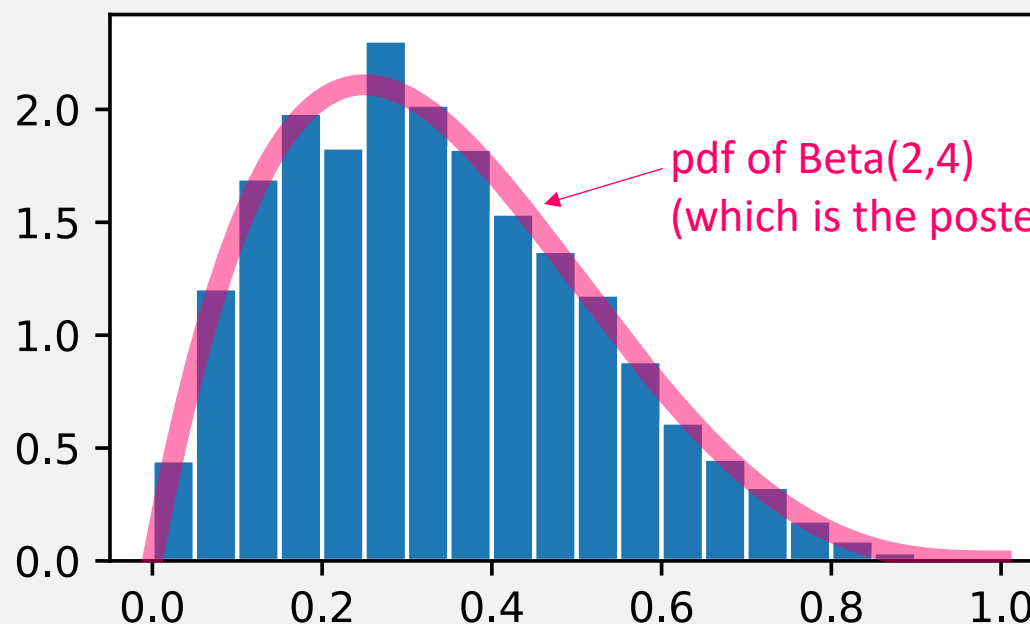
let the bar for a given bin have

$$\text{bar height} = \frac{\mathbb{P}(\Theta \in \text{bin}|\text{data})}{\text{bin width}}$$

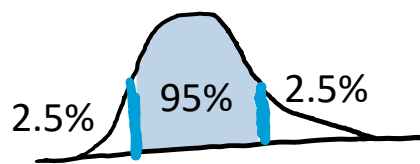
$\Rightarrow$ bar area = $\mathbb{P}(\Theta \in \text{bin}|\text{data})$

It means the plot is directly comparable to a pdf.

```
plt.hist( $\theta$ samp, weights=w) density=True)
```



How do we compute confidence intervals from a weighted sample?



For a real-valued parameter Θ ,
we'd like to report a *95% confidence interval* $[l_o, h_i]$ such that

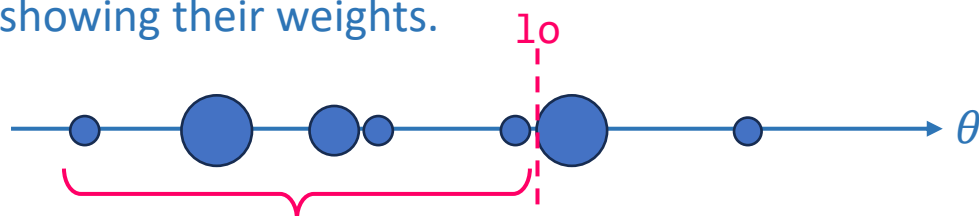
$$\mathbb{P}(\Theta < l_o \mid \text{data}) = 2.5\%$$

$$\mathbb{P}(\Theta > h_i \mid \text{data}) = 2.5\%$$

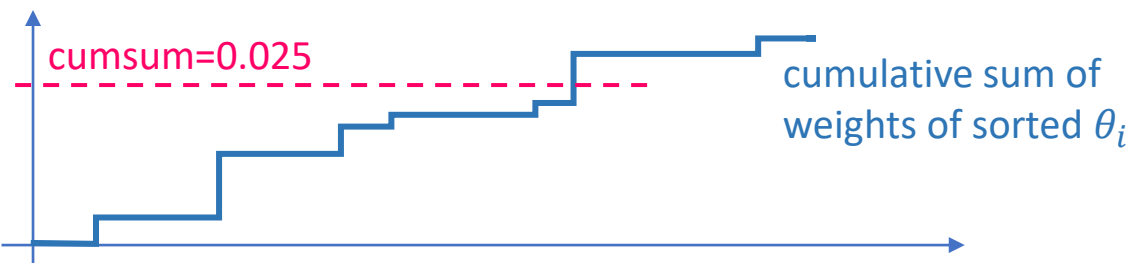
We can find l_o via a computational Bayes estimate (and h_i similarly):

$$\mathbb{P}(\Theta < l_o \mid \text{data}) \approx \sum_i w_i 1_{\theta_i < l_o}$$

Consider a dot-plot of the θ_i ,
showing their weights.



We want to choose l_o so
that the sum of weights
for these θ_i is 0.025



```

1   $\theta_{\text{samp}}, w = \dots$ 
2   $i = \text{np.argsort}(\theta_{\text{samp}})$ 
3   $\theta_{\text{samp}}, w = \theta_{\text{samp}}[i], w[i]$ 
4   $F = \text{np.cumsum}(w)$ 
5   $l_o = \theta_{\text{samp}}[F < 0.025][-1]$ 

```

Exercise 8.3.2 (Multiple unknowns)

We have a dataset $[x_1, \dots, x_n]$. 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 $(B|\text{data})$.

A, B unknown params
↓
 $x_1, \dots, x_n$ observe data:
 $x_1, \dots, x_n = x_1, \dots, x_n.$

$(a_1, b_1), \dots, (a_n, b_n)$

A, B

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

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

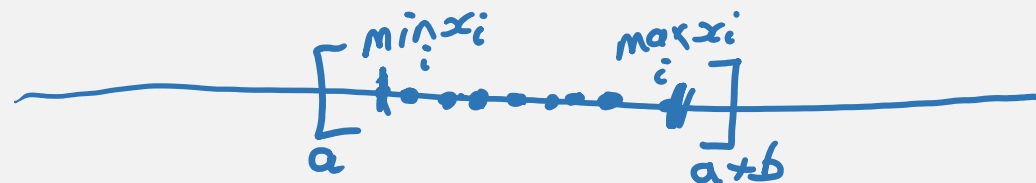
2. Compute weights $w_i = \Pr_X(x|\Theta = \theta_i)$, then rescale weights to sum to one

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

$$\Pr(x_1, \dots, x_n | A=a, B=b) = \prod_{i=1}^n \Pr(x_i | a, b)$$

$$= \prod_{i=1}^n \left\{ \frac{1}{b} \mathbb{1}_{a \leq x_i \leq a+b} \right\}$$

$$= \frac{1}{b^n} \mathbb{1}_{a \leq \min_i x_i} \mathbb{1}_{\max_i x_i \leq a+b}$$



```
# The dataset  
x = [2, 3, 2.1, 2.4, 3.14, 1.8]
```

```
# Step 1
```

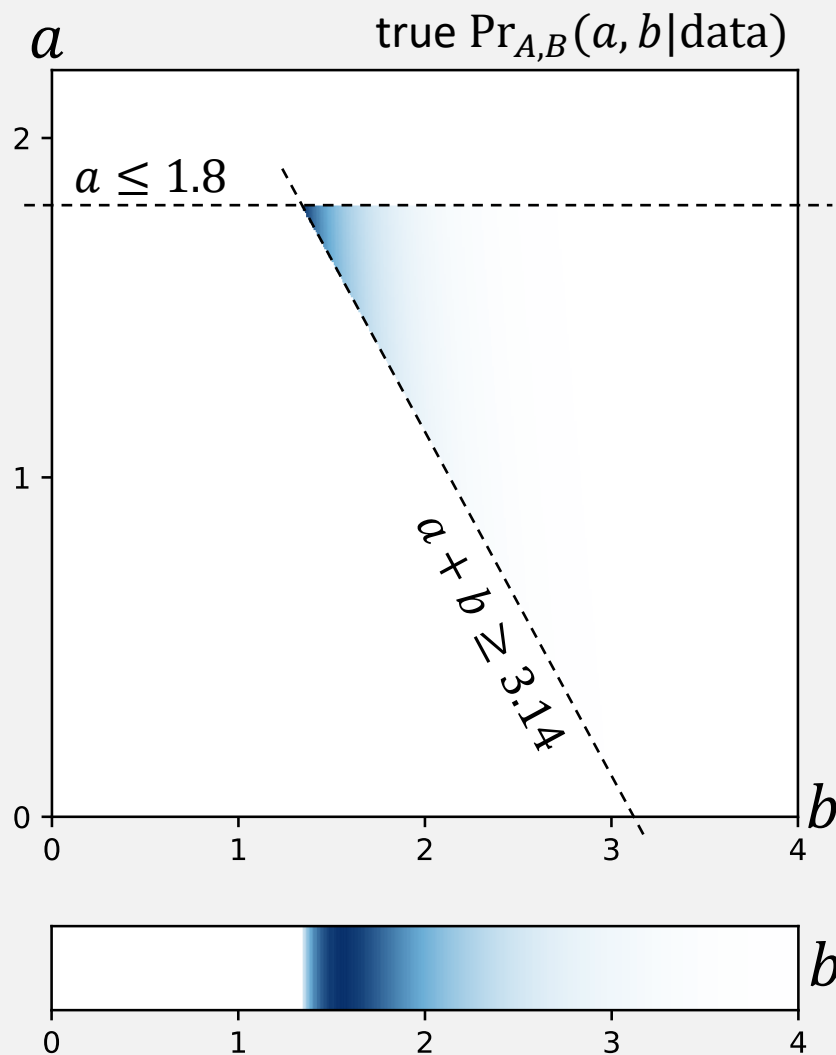
```
asamp = np.random.exponential(scale=1/0.5, size=100000)  
bsamp = np.random.exponential(scale=1/1.0, size=100000)  
#absamp = zip(asamp, bsamp)
```

```
# Step 2
```

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

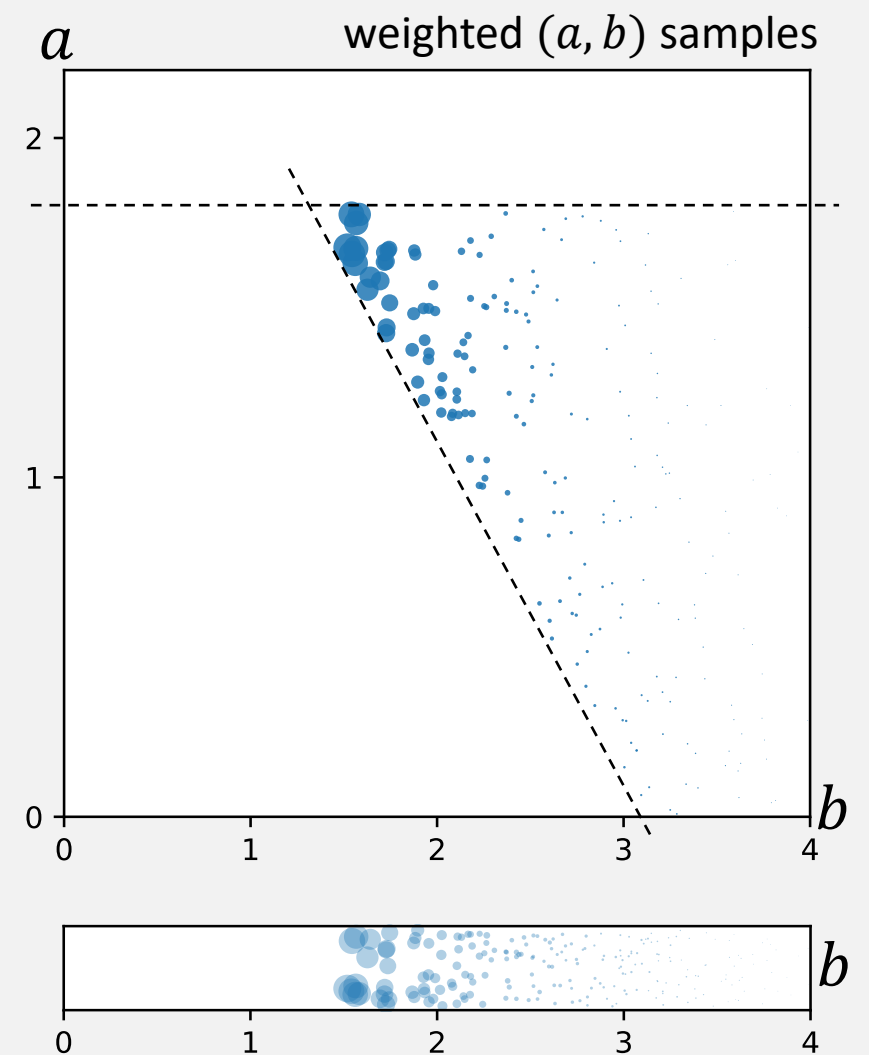
```
# Plot the posterior distribution of B
```

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



The marginal density for B is

$$\Pr_B(b) = \int_a \Pr_{A,B}(a, b) da$$



To get the marginal distribution of B :
get samples from the joint distribution of
 (A, B) then simply ignore A