

AM207 Lecture 2

<https://am207.info/>

AM207 Class Infrastructure

- Website am207.info
- Join [Piazza](#)
- Join [Slack](#)
- We may add Twitter if we're feeling adventurous so stay posted

AM207 Slack

- Please use for asking questions during lecture and lab (if you're not present to raise your hand and ask)
- The channel for the current lecture is #lecture
- The channel for the current lab is #lab
- We'll rename after class/lab to #lectureN and #labM
- Don't abuse (we'll announce any other future appropriate uses on Piazza)

Advice from your TFs

- **Collaboration** -- if you collaborate for assignments (HW and Paper/Tutorial) for which we allow students to work together PLEASE PLEASE SUBMIT ONE ASSIGNMENT.
- **Contacting Teaching Staff*** -- We pride ourselves on being available. Please come to OH (the class will be a lot easier if you do so).
- You can also email us at am207.info. Right now we have aliases for grading (grading@) and info (info@) .

Random Variables

Definition. A random variable is a mapping

$$X : \Omega \rightarrow \mathbb{R}$$

that assigns a real number $X(\omega)$ to each outcome ω .

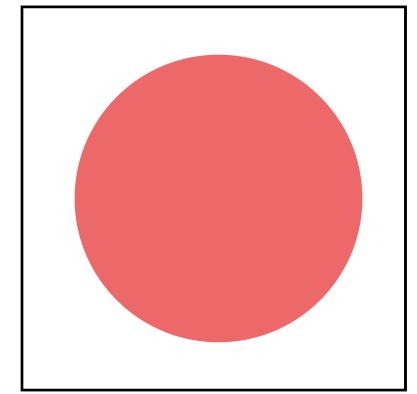
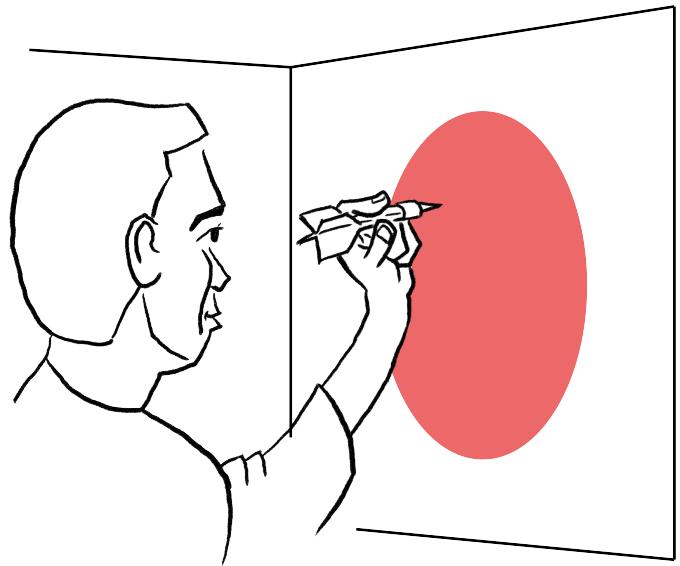
- Ω is the sample space. Points
- ω in Ω are called sample outcomes, realizations, or elements.
- Subsets of Ω are called Events.

Fundamental rules of probability:

1. $p(X) \geq 0$; probability must be non-negative
2. $0 \leq p(X) \leq 1$
3. $p(X) + p(X^-) = 1$ either happen or not happen.
4. $p(X + Y) = p(X) + p(Y) - p(X, Y)$

- Say $\omega = HHTTTTHHT$ then $X(\omega) = 3$ if defined as number of heads in the sequence ω .
- We will assign a real number $P(A)$ to every event A , called the probability of A .
- We also call P a probability distribution or a probability measure.

Probability as frequency



$$P(A) = \frac{\text{Red Circle}}{\text{White Square}}$$

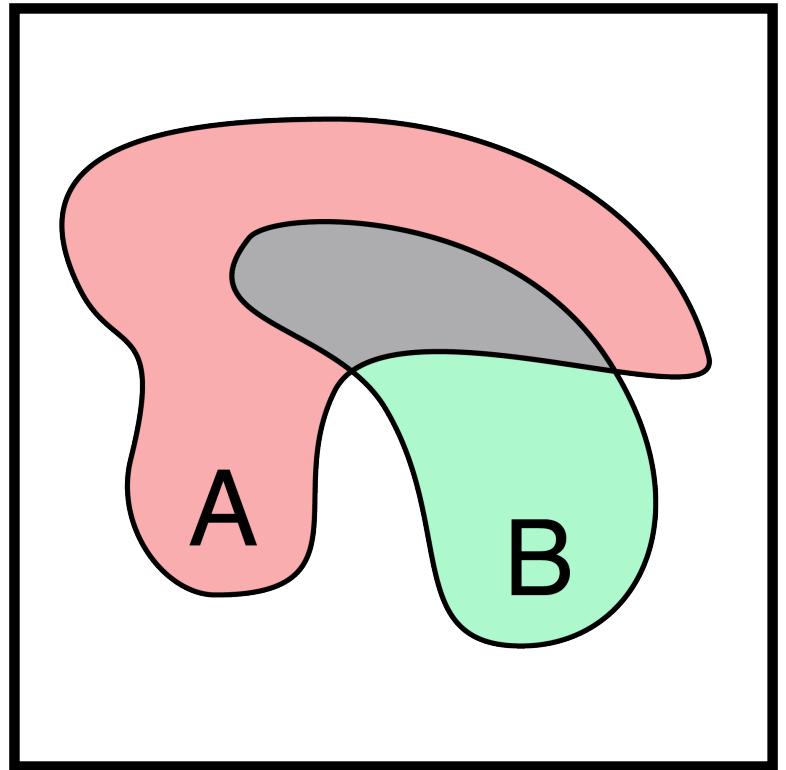
A Murder Mystery

(from the book: Model Based Machine Learning)

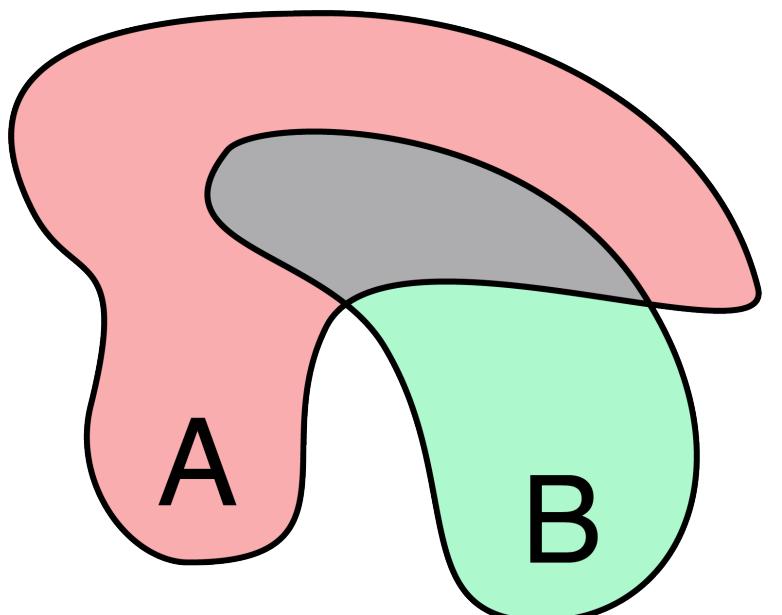
- Mr Black is dead
- We represent the murderer with a random variable murderer whose value we dont know. This variable equals either Auburn or Grey.
- $p(\text{murderer} = \text{Auburn}) = 0.7$
- The "prior" distribution for murder is the Bernoulli:
 $\text{murderer} \sim \text{Bernoulli}(0.7)$

Evidence and conditional probability

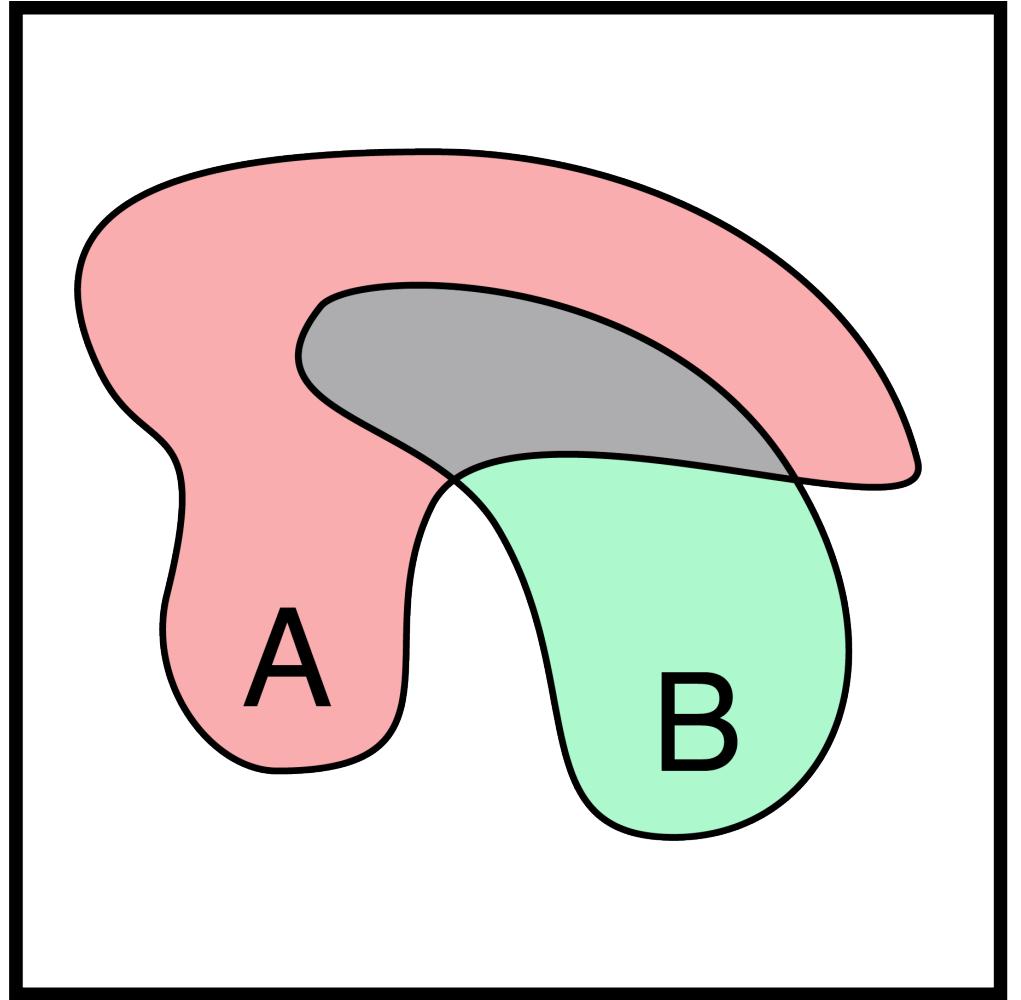
- an ornate ceremonial dagger and an old army revolver are found.
We thus introduce a new random variable **weapon**, in addition to the existing random variable **murderer**.
- $p(\text{weapon} = \text{revolver} \mid \text{murderer} = \text{grey}) = 0.9$,
 $p(\text{weapon} = \text{revolver} \mid \text{murderer} = \text{auburn}) = 0.2$



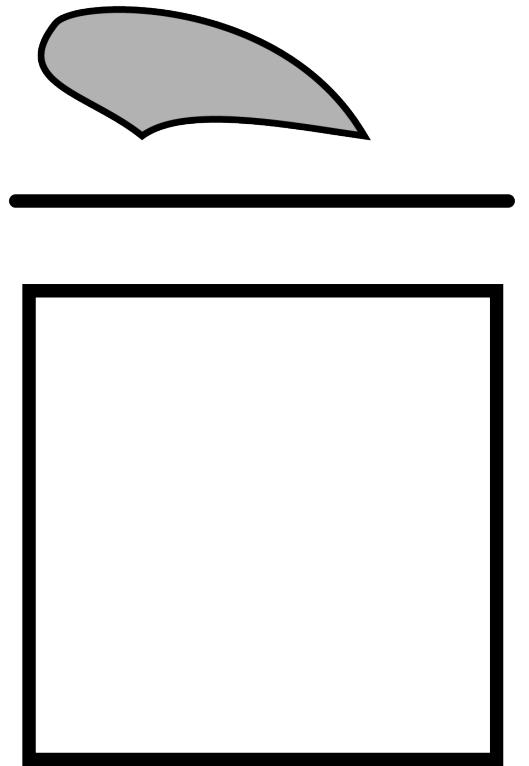
$$P(A|B) = \frac{\text{Shaded Area}}{\text{Total Area of B}}$$



$$P(B|A) = \frac{\text{Shaded Area}}{\text{Total Area of A}}$$



$$P(A, B) =$$



The joint Probability distribution

$$P(\text{weapon, murderer}) = P(\text{murderer}) \times P(\text{weapon}|\text{murderer})$$

Diagram illustrating the joint probability distribution:

- Joint Probability Matrix ($P(\text{weapon, murderer})$):

	0.14
0.27	0.56
0.03	
- Marginal Probability Matrix ($P(\text{murderer})$):

	0.30
0.30	0.70
0.10	
- Conditional Probability Matrix ($P(\text{weapon}|\text{murderer})$):

	0.20
0.80	

A probabilistic model is:

- A set of random variables,
- A joint probability distribution over these variables (i.e. a distribution that assigns a probability to every configuration of these variables such that the probabilities add up to 1 over all possible configurations).

Now we condition on some random variables and learn the values of others.

(paraphrased from Model Based Machine Learning)

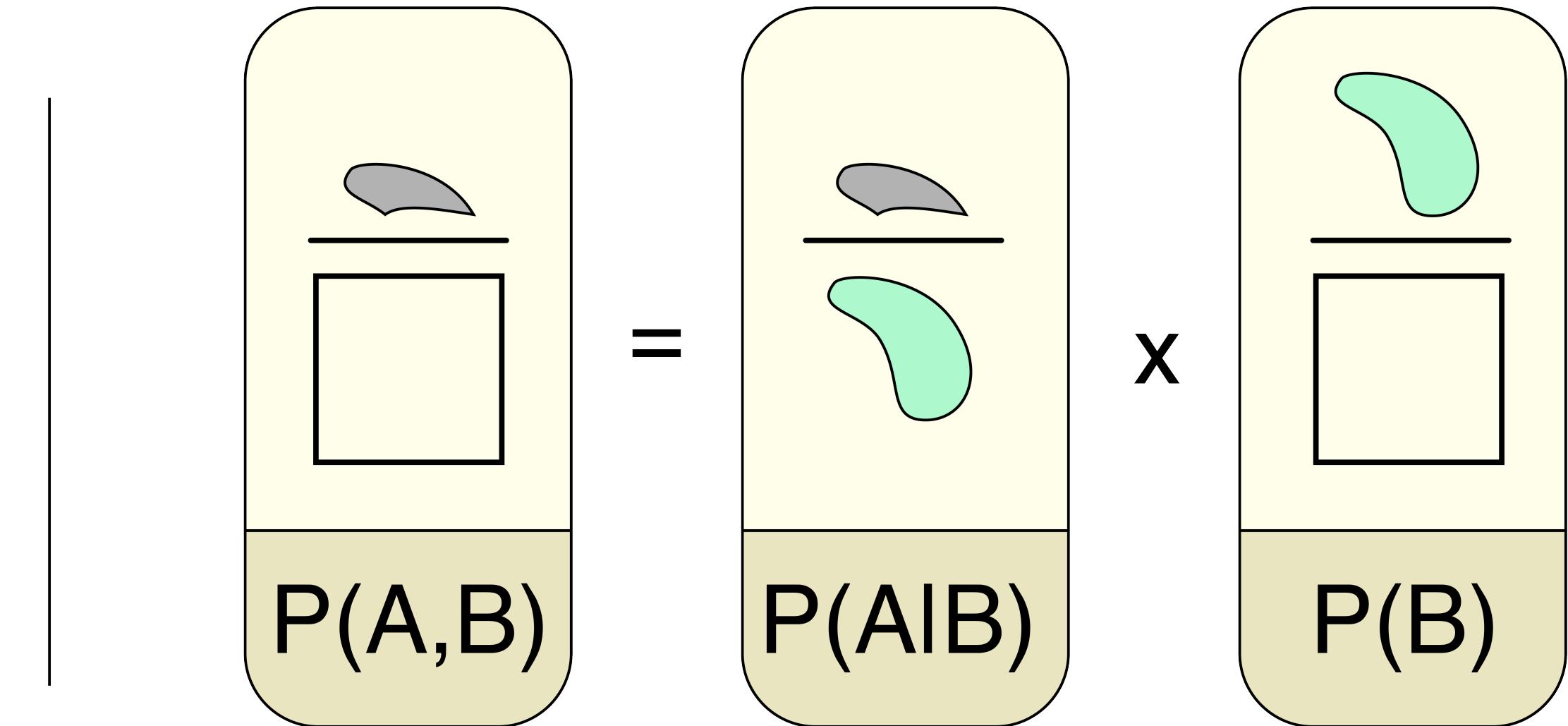
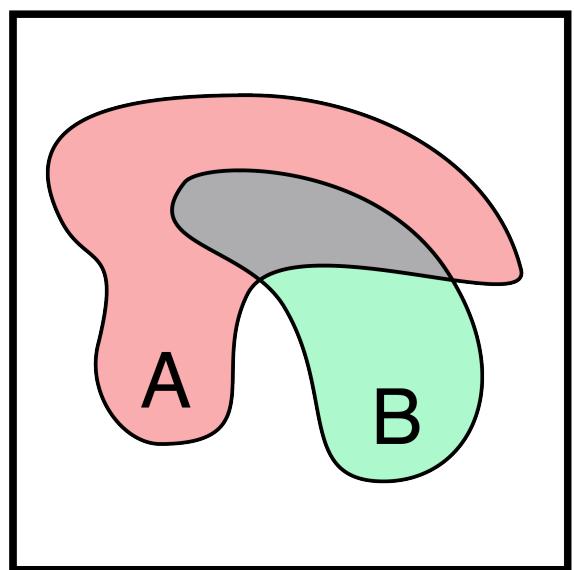
Rules

$$1. P(A, B) = P(A | B)P(B)$$

$$2. P(A) = \sum_B P(A, B) = \sum_B P(A | B)P(B)$$

$P(A)$ is called the **marginal** distribution of A, obtained by summing or marginalizing over B .

Conditional Rule



Marginal Rule

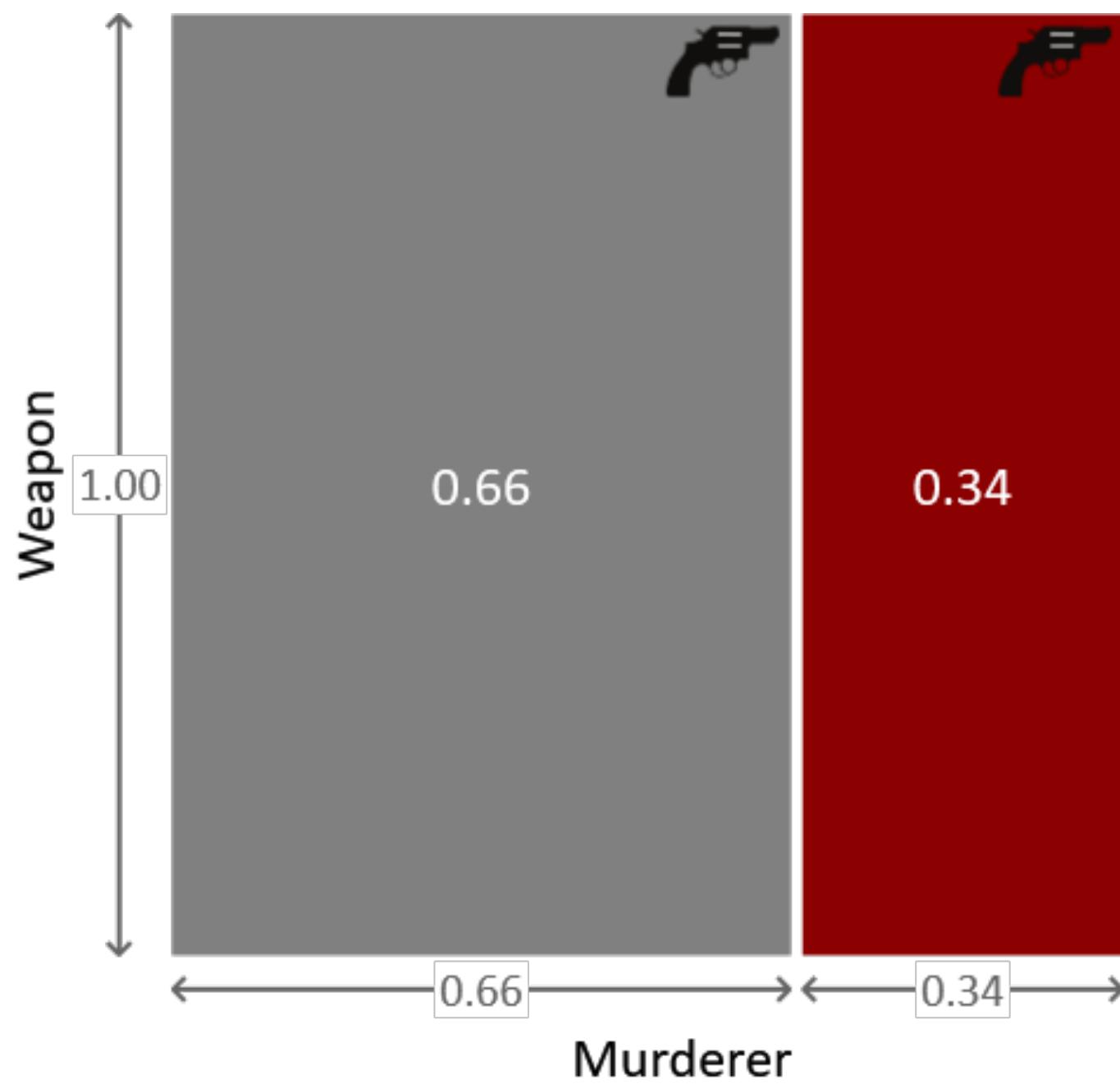
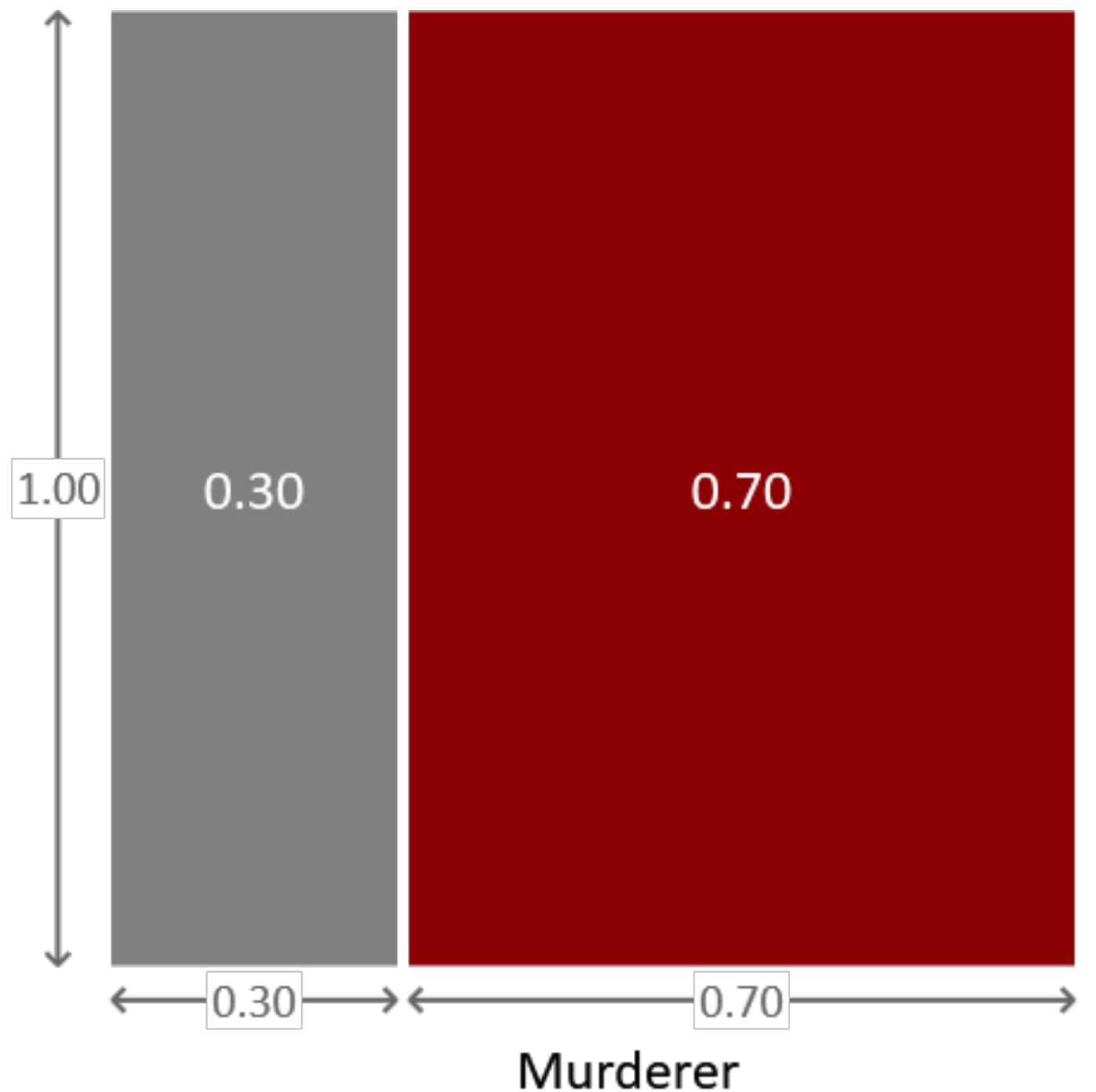
	Vanilla	Chocolate	
Cone	40	60	$P(\text{Cone}) = 100/150 \approx 0.66$
Cup	20	30	$P(\text{Cup}) = 50/150 \approx 0.33$
	$P(\text{Vanilla}) = 60/150 = 0.4$	$P(\text{Chocolate}) = 90/150 = 0.6$	

Observation and Inference

- Dr Bayes spots a bullet lodged in the book case.

The process of computing revised probability distributions after we have observed the values of some the random variables, is called inference.

- a principled way from prior to posterior



Bayes Theorem: Inference without computing the joint distribution

Why? The joint can be computationally hard. Sometimes there are two many "factors"

$$p(y \mid x) = \frac{p(x \mid y) p(y)}{p(x)} = \frac{p(x \mid y) p(y)}{\sum_{y'} p(x, y')} = \frac{p(x \mid y) p(y)}{\sum_{y'} p(x \mid y') p(y')}$$

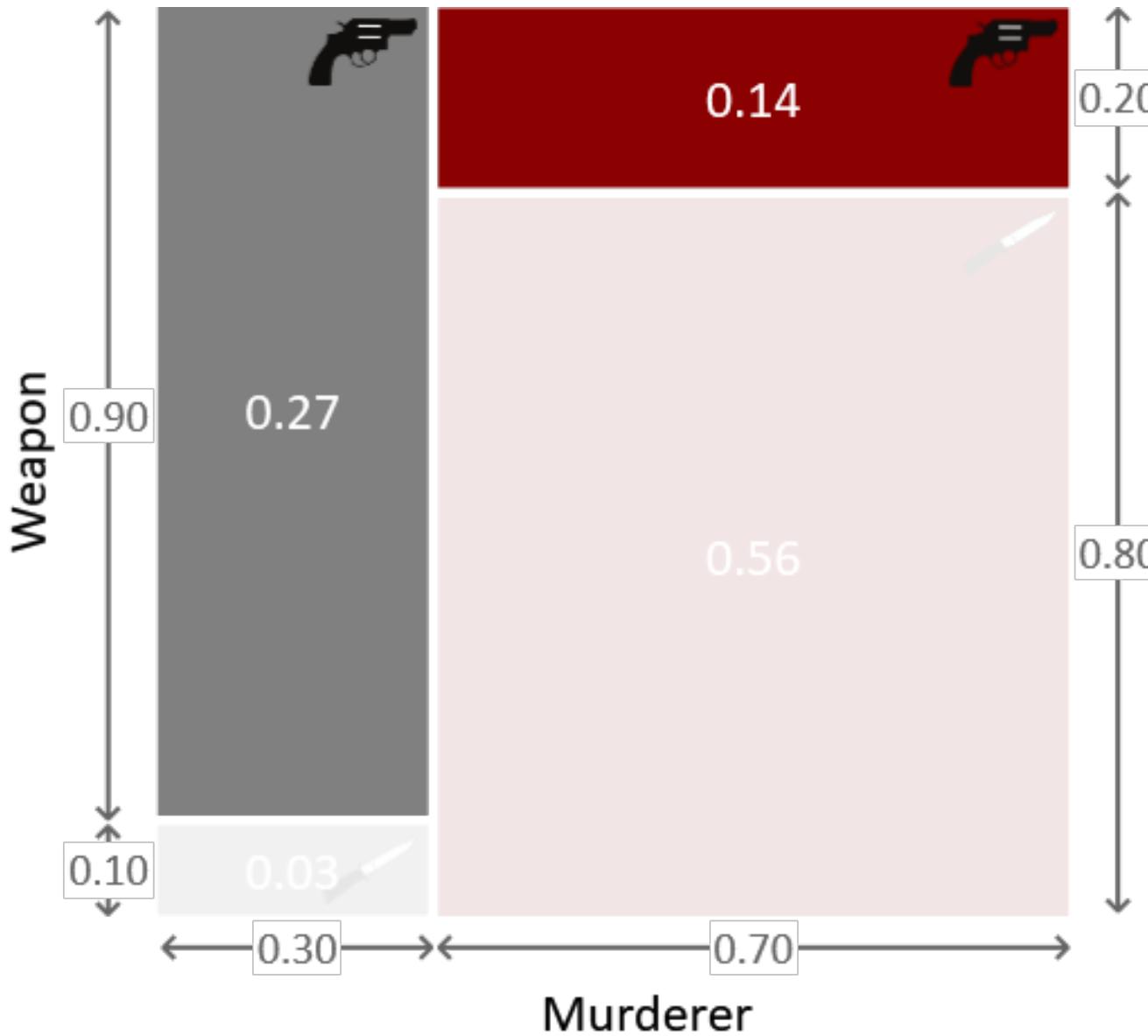
$$P(\text{murderer}|\text{weapon}) = \frac{P(\text{weapon}|\text{murderer})P(\text{murderer})}{P(\text{weapon})}.$$

$$P(\text{weapon}) = \sum_{\text{murderer}} P(\text{weapon}|\text{murderer})P(\text{murderer})$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

The evidence is just a normalizer and can often be ignored.

The likelihood function is NOT a probability distribution over weapon (which is known!). It is a function of the random variable murderer.



Just ignore the fact that we are in a square!

Lets get precise

Cumulative distribution Function

The **cumulative distribution function**, or the **CDF**, is a function

$$F_X : \mathbb{R} \rightarrow [0, 1],$$

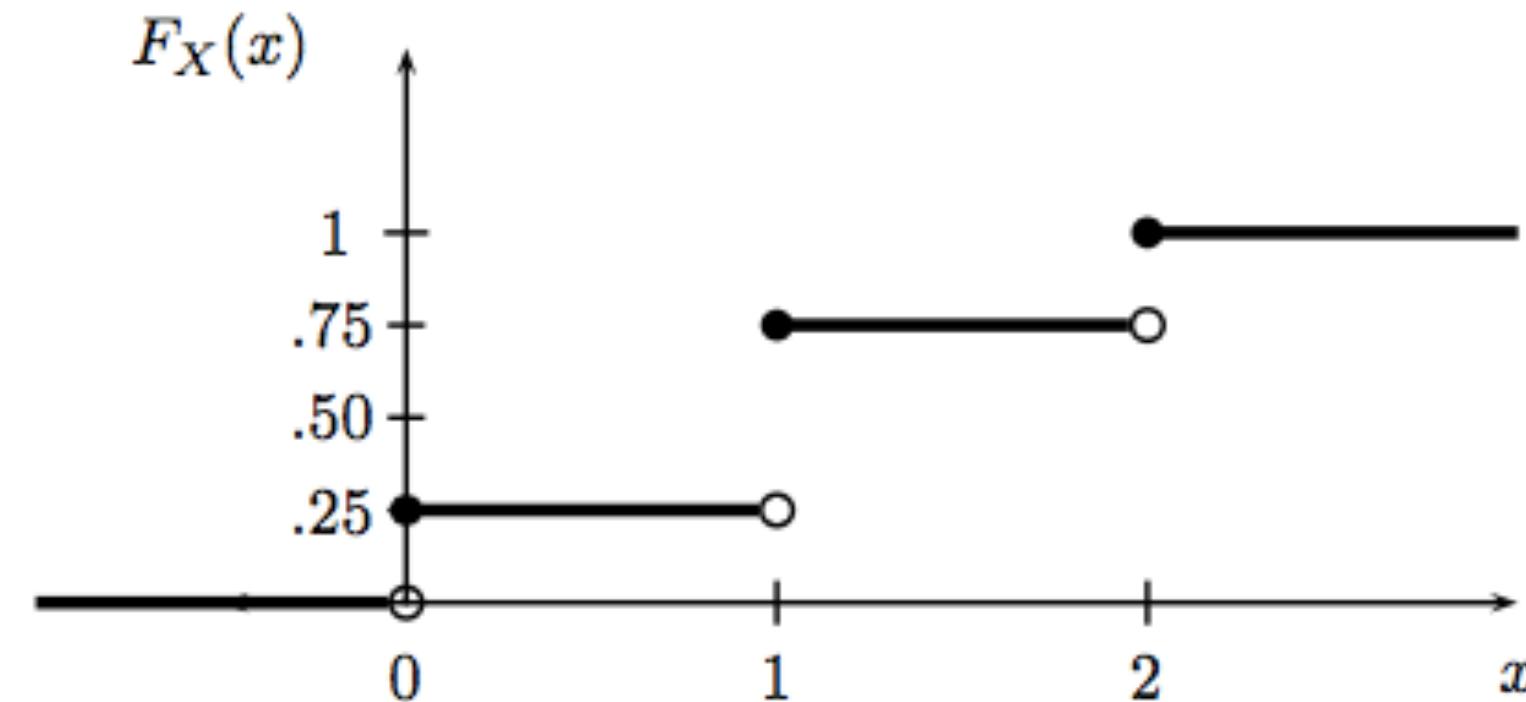
defined by

$$F_X(x) = p(X \leq x).$$

Sometimes also just called *distribution*.

Let X be the random variable representing the number of heads in two coin tosses. Then $x = 0, 1$ or 2 .

CDF:



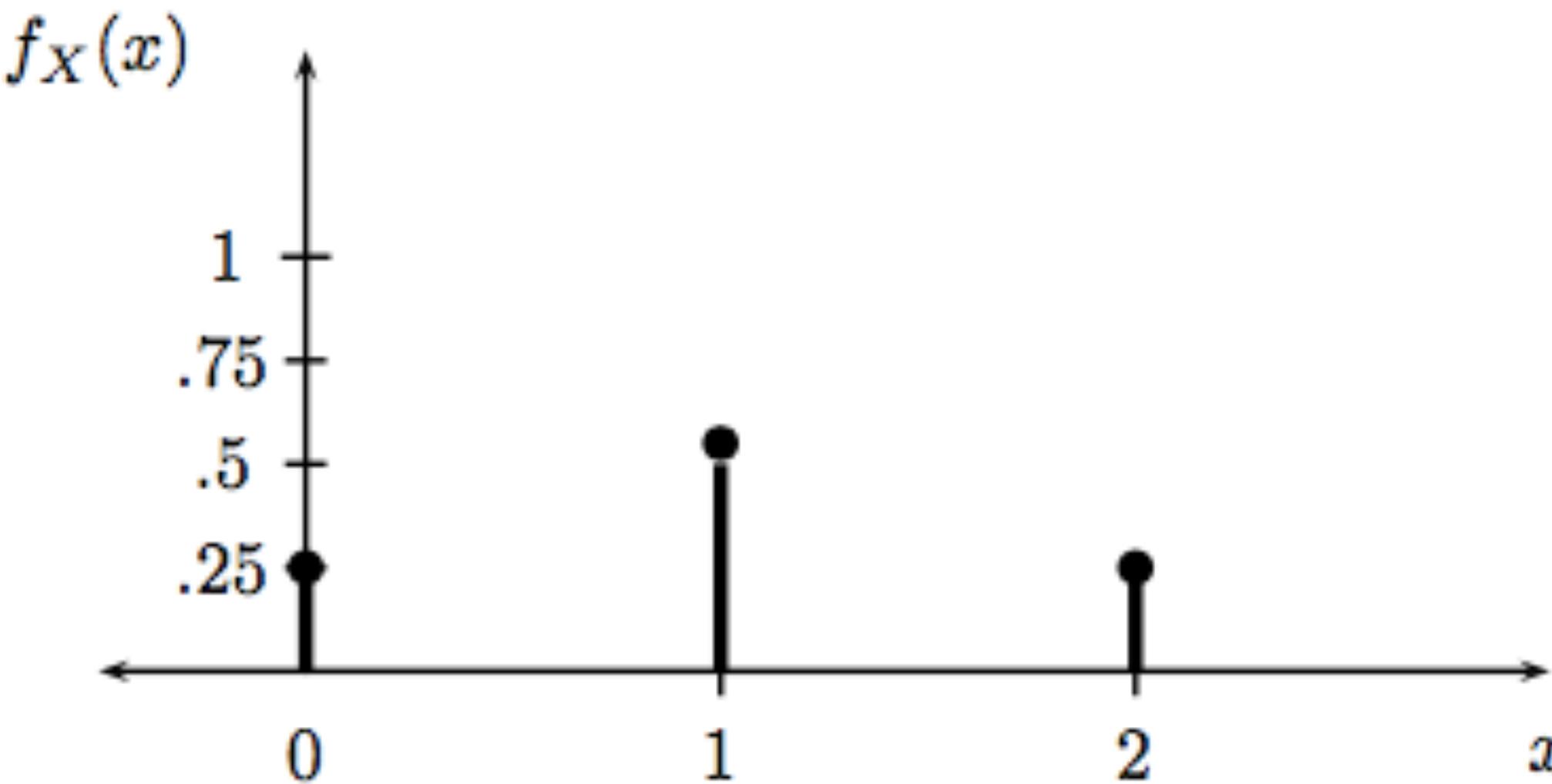
Probability Mass Function

X is called a **discrete random variable** if it takes countably many values $\{x_1, x_2, \dots\}$.

We define the **probability function** or the **probability mass function (pmf)** for X by:

$$f_X(x) = p(X = x)$$

The pmf for the number of heads in two coin tosses:



Probability Density function (pdf)

A random variable is called a **continuous random variable** if there exists a function f_X such that $f_X(x) \geq 0$ for all x ,

$$\int_{-\infty}^{\infty} f_X(x)dx = 1 \text{ and for every } a \leq b,$$

$$p(a < X < b) = \int_a^b f_X(x)dx$$

$= F(b) - F(a)$
 F is the CDF of the distribution

Note: $p(X = x) = 0$ for every x . Confusing!

CDF for continuous random variables

$$F_X(x) = \int_{-\infty}^x f_X(t)dt$$

and $f_X(x) = \frac{dF_X(x)}{dx}$ at all points x at which F_X is differentiable.

Continuous pdfs can be > 1 . cdfs bounded in $[0,1]$.

A continuous example: the Uniform(0,1) Distribution

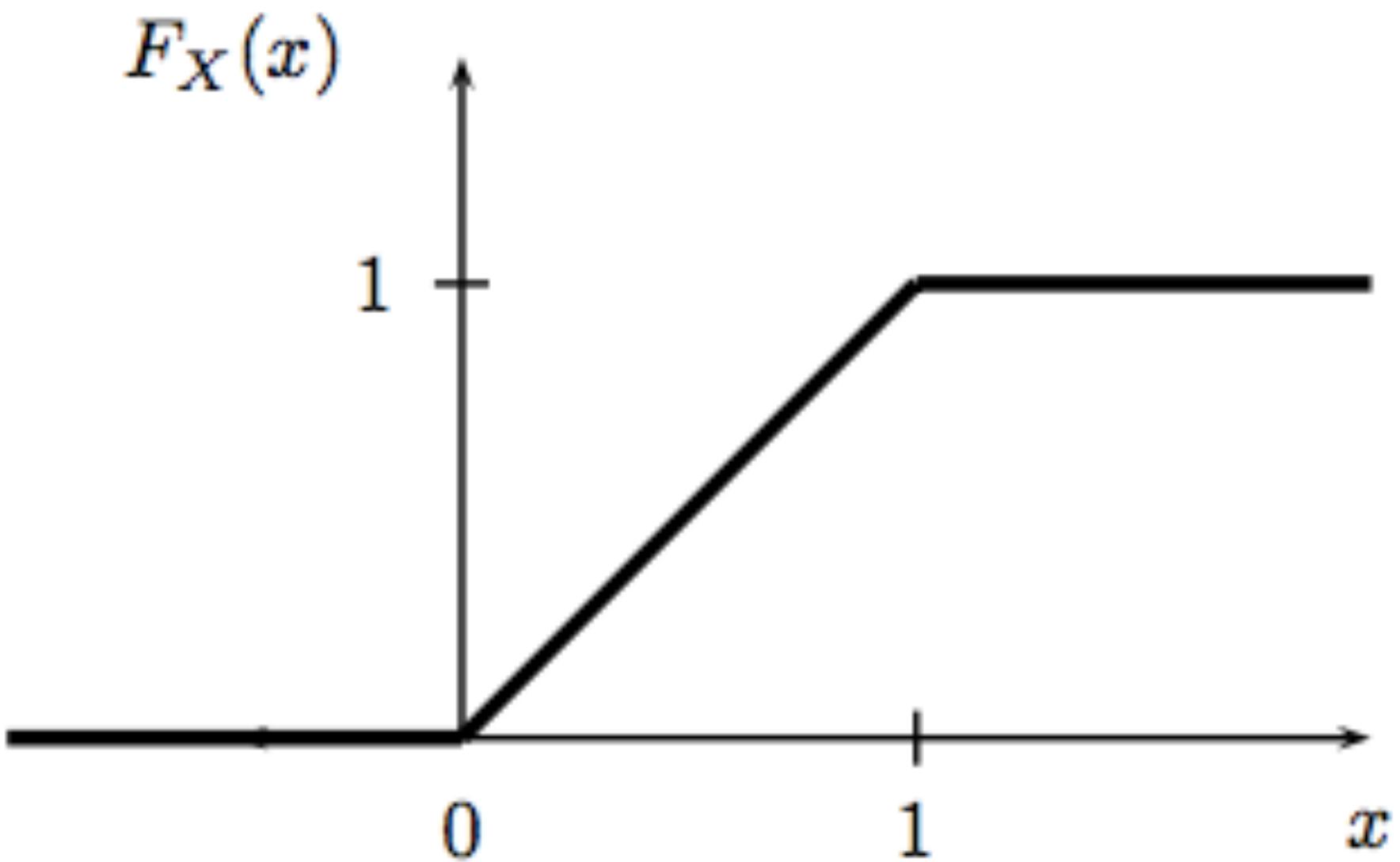
pdf:

$$f_X(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

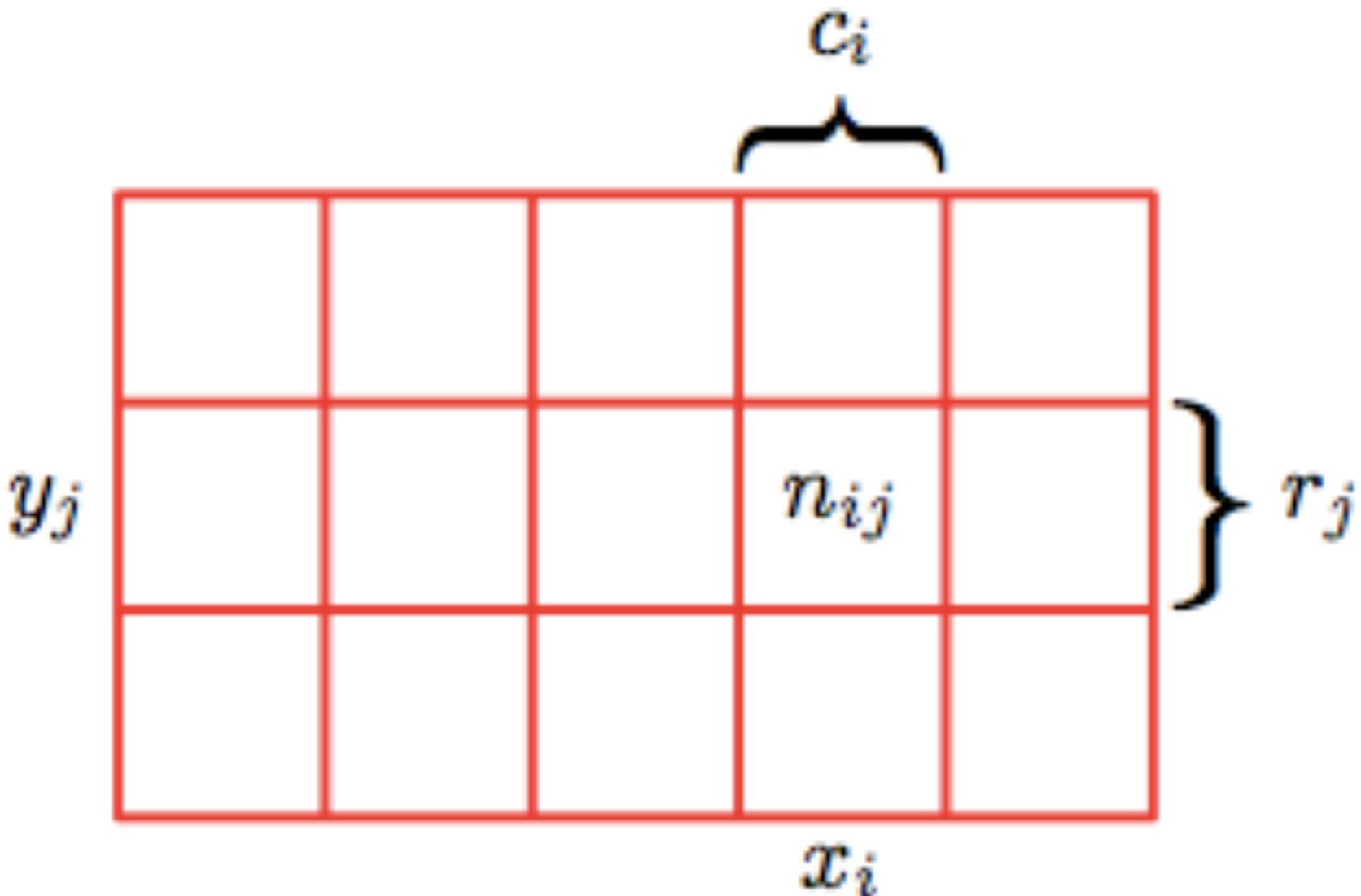
cdf:

$$F_X(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1. \end{cases}$$

cdf:



Marginals and Conditionals



$$p(X = x_i) = \sum_j p(X = x_i, Y = y_j)$$

$$p(Y = y_j \mid X = x_i) \times p(X = x_i) = p(X = x_i, Y = y_j).$$

More generally for hidden variables z :

$$p(x) = \sum_z p(x, z) = \sum_z p(x|z)p(z)$$

Marginals

Marginal mass functions are defined in analog to probabilities:

$$f_X(x) = p(X = x) = \sum_y f(x, y); \quad f_Y(y) = p(Y = y) = \sum_x f(x, y).$$

Marginal densities are defined using integrals:

$$f_X(x) = \int dy f(x, y); \quad f_Y(y) = \int dx f(x, y).$$

Conditionals

Conditional mass function is a conditional probability:

$$f_{X|Y}(x | y) = p(X = x | Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)} = \frac{f_{XY}(x, y)}{f_Y(y)}$$

The same formula holds for densities with some additional requirements $f_Y(y) > 0$ and interpretation:

$$p(X \in A | Y = y) = \int_{x \in A} f_{X|Y}(x, y) dx.$$

Bernoulli pmf:

$$f(x) = \begin{cases} 1 - p & x = 0 \\ p & x = 1. \end{cases}$$

for p in the range 0 to 1.

$$f(x) = p^x (1 - p)^{1-x}$$

for x in the set $\{0,1\}$.

What is the cdf?

The big Ideas
create and simulate a data story
perform inference using data story

Data story

- a story of how the data came to be.
- may be a causal story, or a descriptive one (correlational, associative).
- The story must be sufficient to specify an algorithm to simulate new data*.
- a formal **probability model**.

tossing a globe in the air experiment

- toss and catch it. When you catch it, see what's under index finger
- mark W for water, L for land.
- figure how much of the earth is covered in water
- thus the "data" is the fraction of W tosses

Probabilistic Model

1. The true proportion of water is p .
2. Bernoulli probability for each globe toss, where p is thus the probability that you get a W. This assumption is one of being **Identically Distributed**.
3. Each globe toss is **Independent** of the other.

Assumptions 2 and 3 taken together are called **IID**, or **Independent and Identically Distributed** Data.

Expectations, LLN, Monte Carlo, and the CLT

- Expectations and some notation
- The Law of large numbers
- Simulation and Monte Carlo for Integration
- Sampling and the CLT
- Errors in Monte Carlo

Expectation $E_f[X]$

Why calculate it?

- we'll see it corresponds to the frequentist notion of probability
- we often want point estimates

Expectations are always with respect to a pmf or density. Often just called the **mean** of the mass function or density. More weight to more probable values.

For the discrete random variable X :

$$E_f[X] = \sum_x x f(x).$$

Continuous case:

$$E_f[X] = \int x f(x)dx = \int x dF(x),$$

Notation

The expected value, or mean, or first moment, of X is defined to be

$$E_f X = \int x dF(x) = \begin{cases} \sum_x x f(x) & \text{if } X \text{ is discrete} \\ \int x f(x) dx & \text{if } X \text{ is continuous} \end{cases}$$

assuming that the sum (or integral) is well defined.

The discrete sum can be said to be an integral with respect to a counting measure.

LOTUS: Law of the unconscious statistician

Also known as **The rule of the lazy statistician.**

Theorem:

if $Y = r(X)$,

$$E[Y] = \int r(x)dF(x)$$

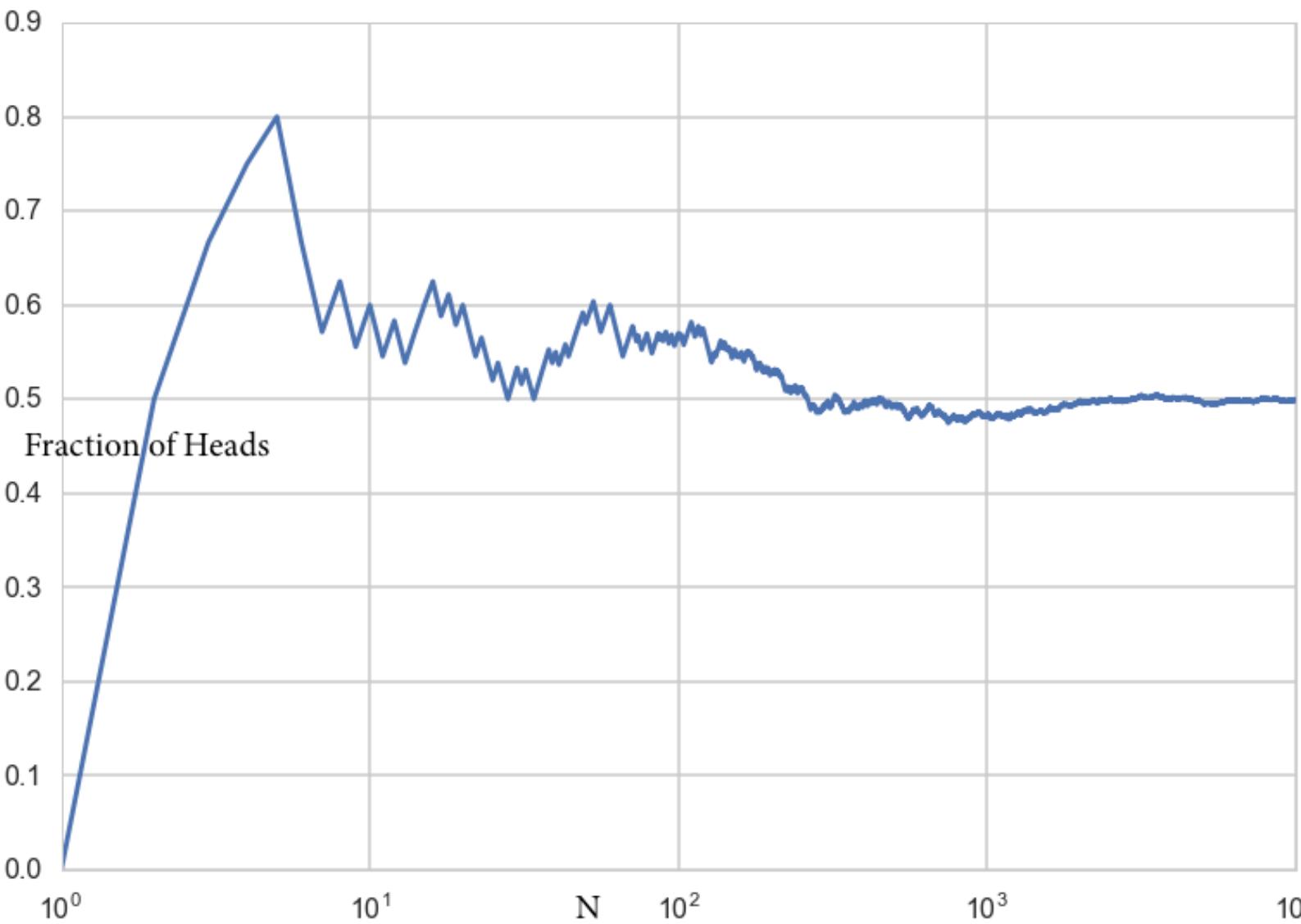
Application: Probability as Expectation

Let A be an event and let $r(x) = I_A(x)$ (Indicator for event A)

Then:

$$E_f[I_A(X)] = \int I_A(x)dF(x) = \int_A f_X(x)dx = p(X \in A)$$

Ever longer sequences for means



Law of Large numbers

Let x_1, x_2, \dots, x_n be a sequence of IID values from random variable X , which has finite mean μ . Let:

$$S_n = \frac{1}{n} \sum_{i=1}^n x_i,$$

Then:

$$S_n \rightarrow \mu \text{ as } n \rightarrow \infty.$$

Frequentist Interpretation of probability

$$E_F[I_A(X)] = p(X \in A)$$

Suppose $Z = I_A(X) \sim \text{Bernoulli}(p = P(A))$.

Now if we take a long sequence seq=10010011100... from Z , then

$$P(A) = \text{mean}(\text{seq}) \text{ as } \text{length}(\text{seq}) \rightarrow \infty$$

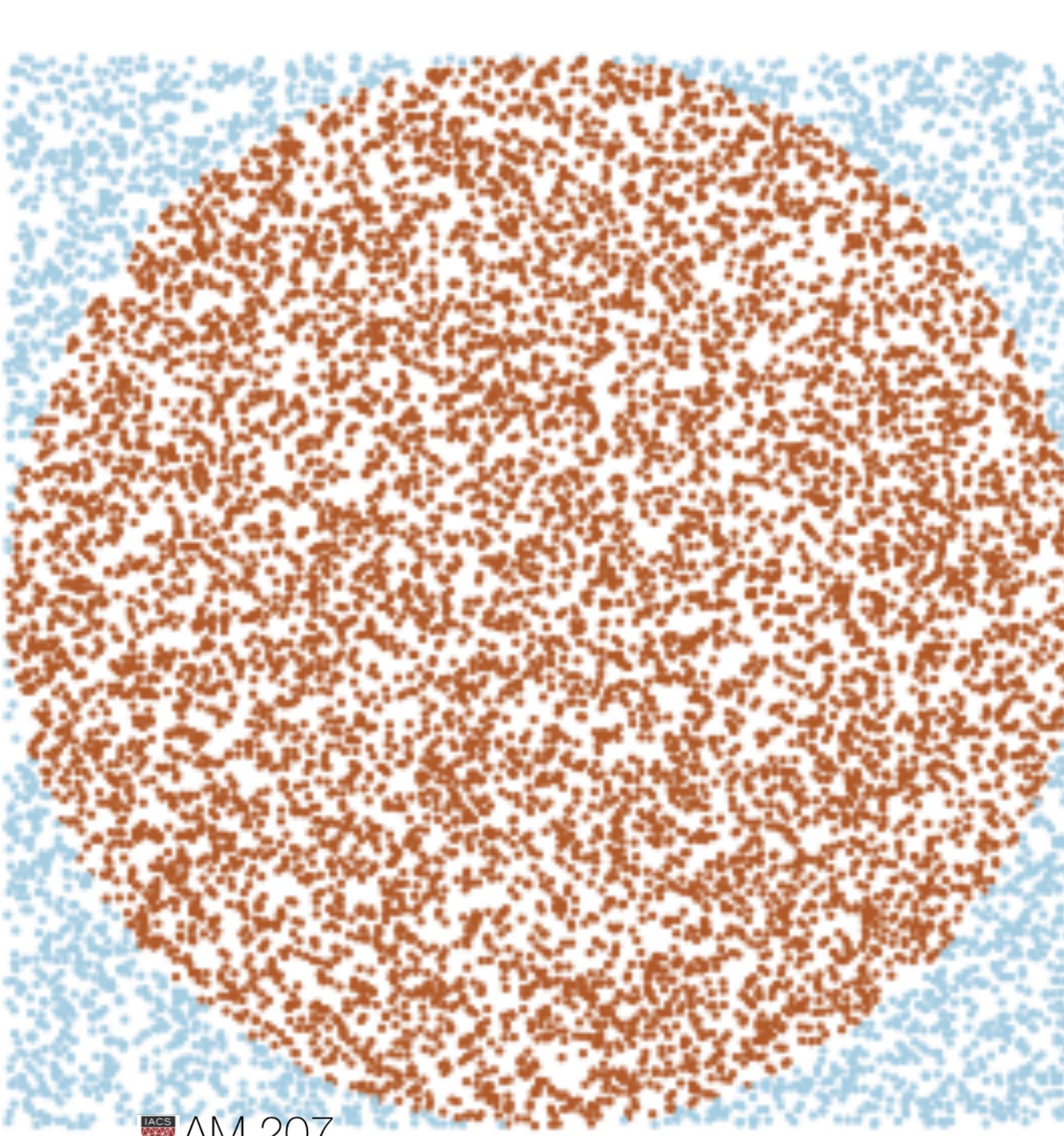
Monte Carlo Algorithm

- use randomness to solve what is often a deterministic problem
- application of the law of large numbers
- integrals, expectations, marginalization
- we'll study optimization, integration, and obtaining draws from a probability distribution

...I wondered whether a more practical method than “abstract thinking” might not be to lay it out say one hundred times and simply observe and count the number of successful plays

*...and more generally how to change
processes described by certain
differential equations into an
equivalent form interpretable as a
succession of random operations*

– Stanislaw Ulam



estimating π

$$A = \int_x \int_y I_{\in C}(x, y) dx dy = \int \int_{\in C} dx dy$$

$$\begin{aligned} E_f[I_{\in C}(X, Y)] &= \int I_{\in C}(X, Y) dF(X, Y) \\ &= \int \int_{\in C} f_{X,Y}(x, y) dx dy = p(X, Y \in C) \end{aligned}$$

If $f_{X,Y}(x, y) \sim Uniform(V)$:

$$= \frac{1}{V} \int \int_{\in C} dx dy = \frac{A}{V}$$

Formalize Monte Carlo Integration idea

For Uniform pdf: $U_{ab}(x) = 1/V = 1/(b - a)$

$$J = \int_a^b f(x) U_{ab}(x) dx = \int_a^b f(x) dx / V = I/V$$

From LOTUS and the law of large numbers:

$$I = V \times J = V \times E_U[f] = V \times \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{x_i \sim U} f(x_i)$$

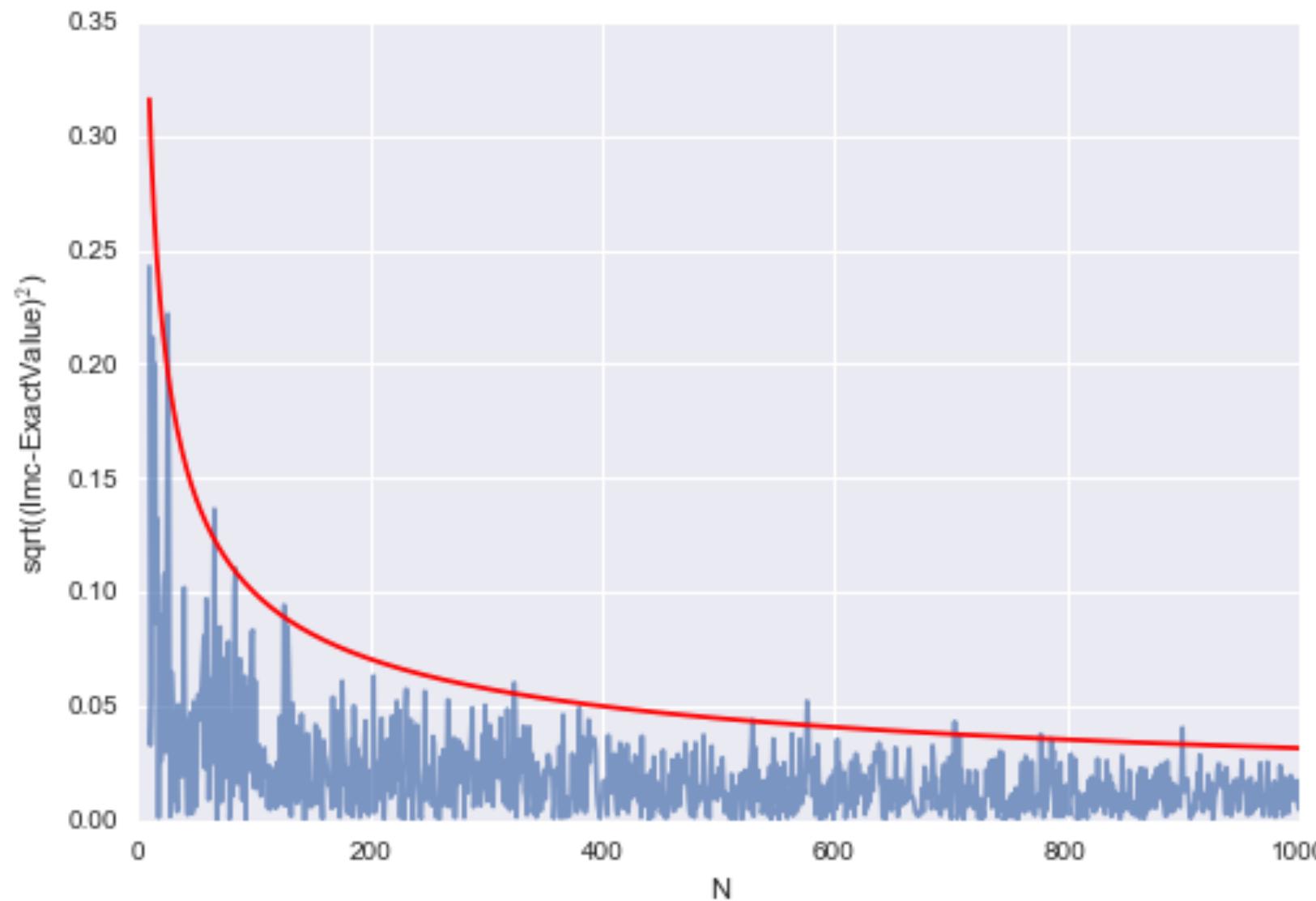
Example

$$I = \int_2^3 [x^2 + 4x \sin(x)] dx.$$

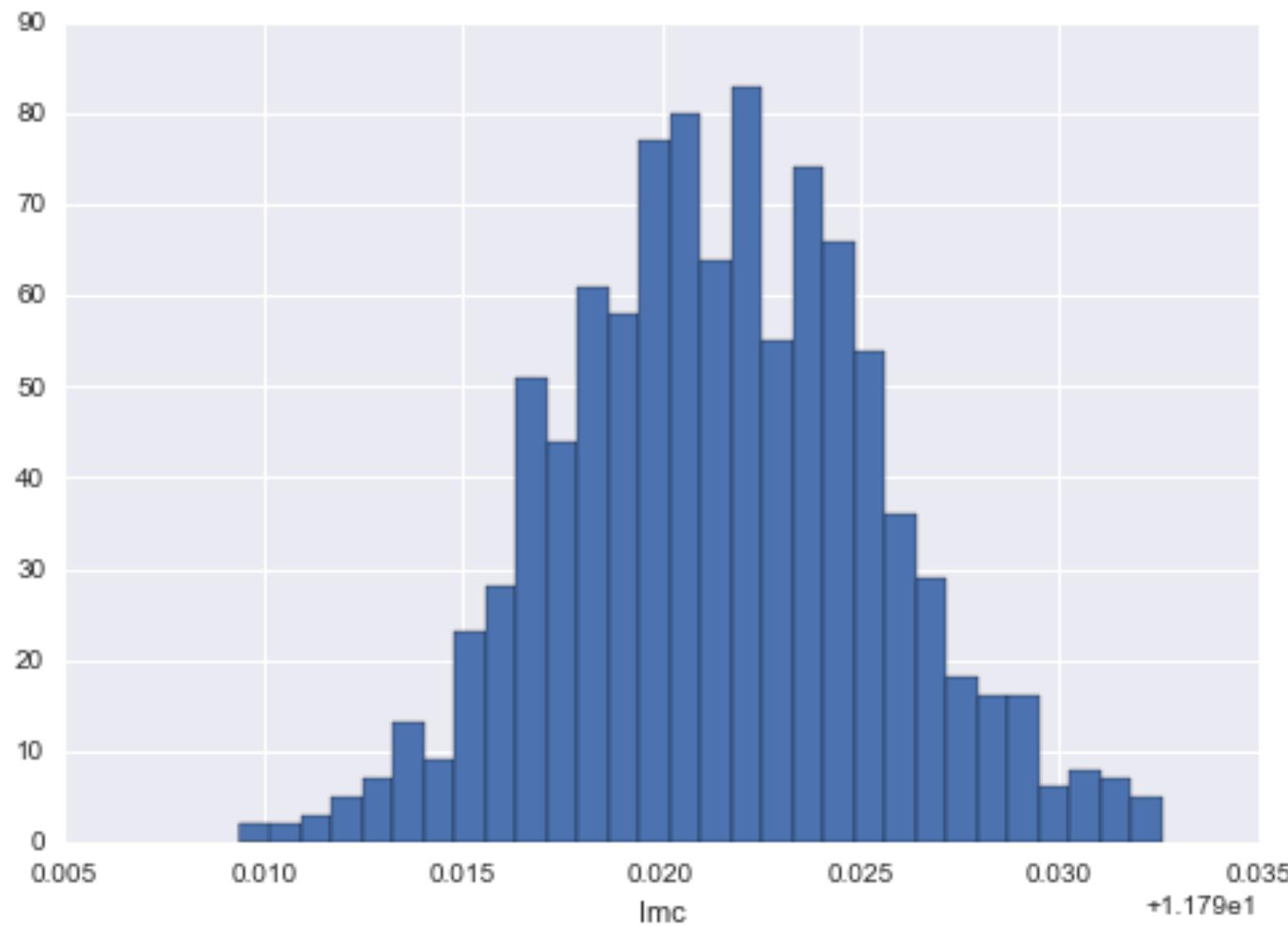
```
def f(x):
    return x**2 + 4*x*np.sin(x)
def intf(x):
    return x**3/3.0+4.0*np.sin(x) - 4.0*x*np.cos(x)
a = 2;
b = 3;
N= 10000
X = np.random.uniform(low=a, high=b, size=N)
Y =f(X)
V = b-a
Imc= V * np.sum(Y)/ N;
exactval=intf(b)-intf(a)
print("Monte Carlo estimation=",Imc, "Exact number=", intf(b)-intf(a))
```

Monte Carlo estimation= 11.8120823531 Exact number= 11.8113589251

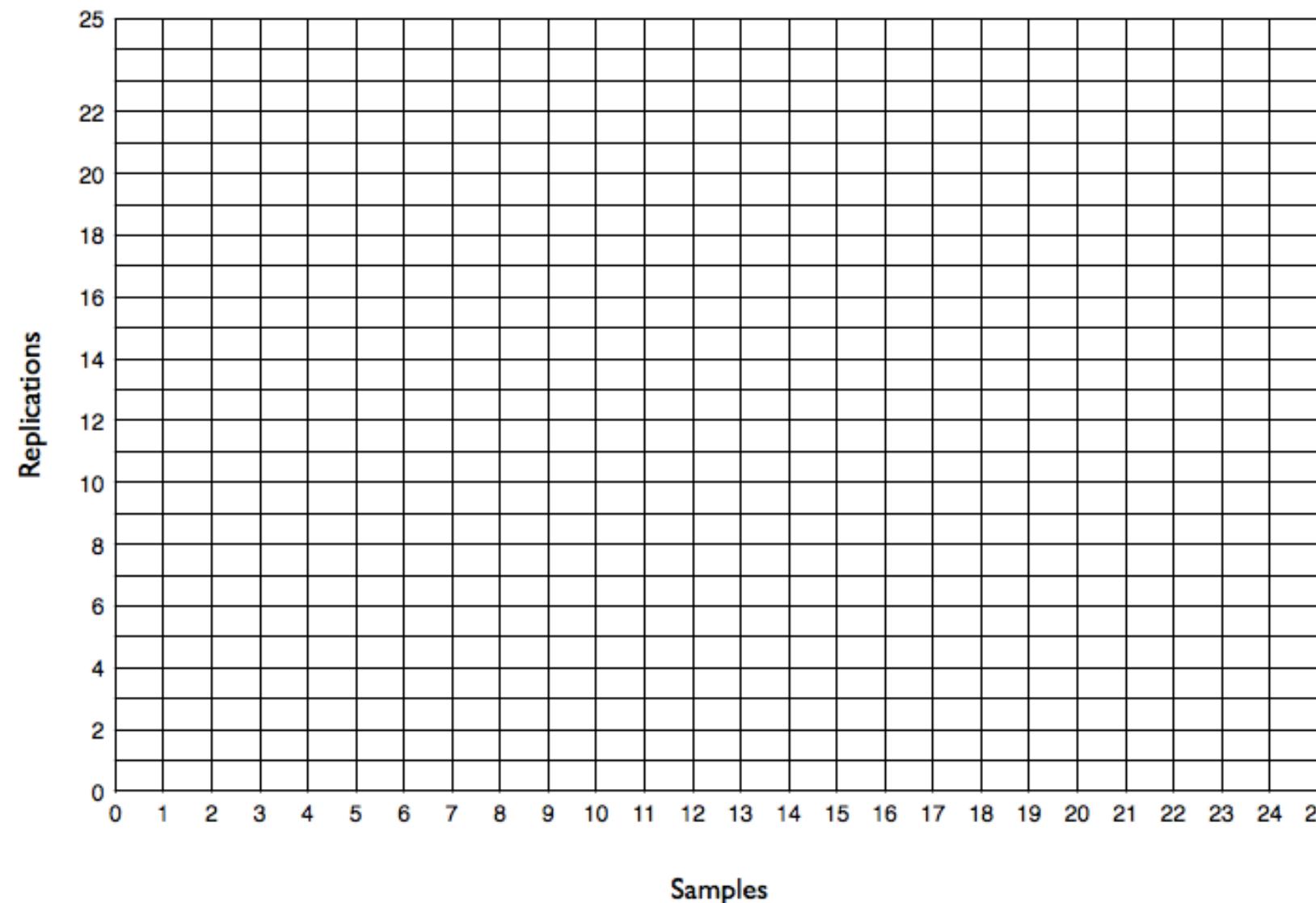
Accuracy as a function of the number of samples



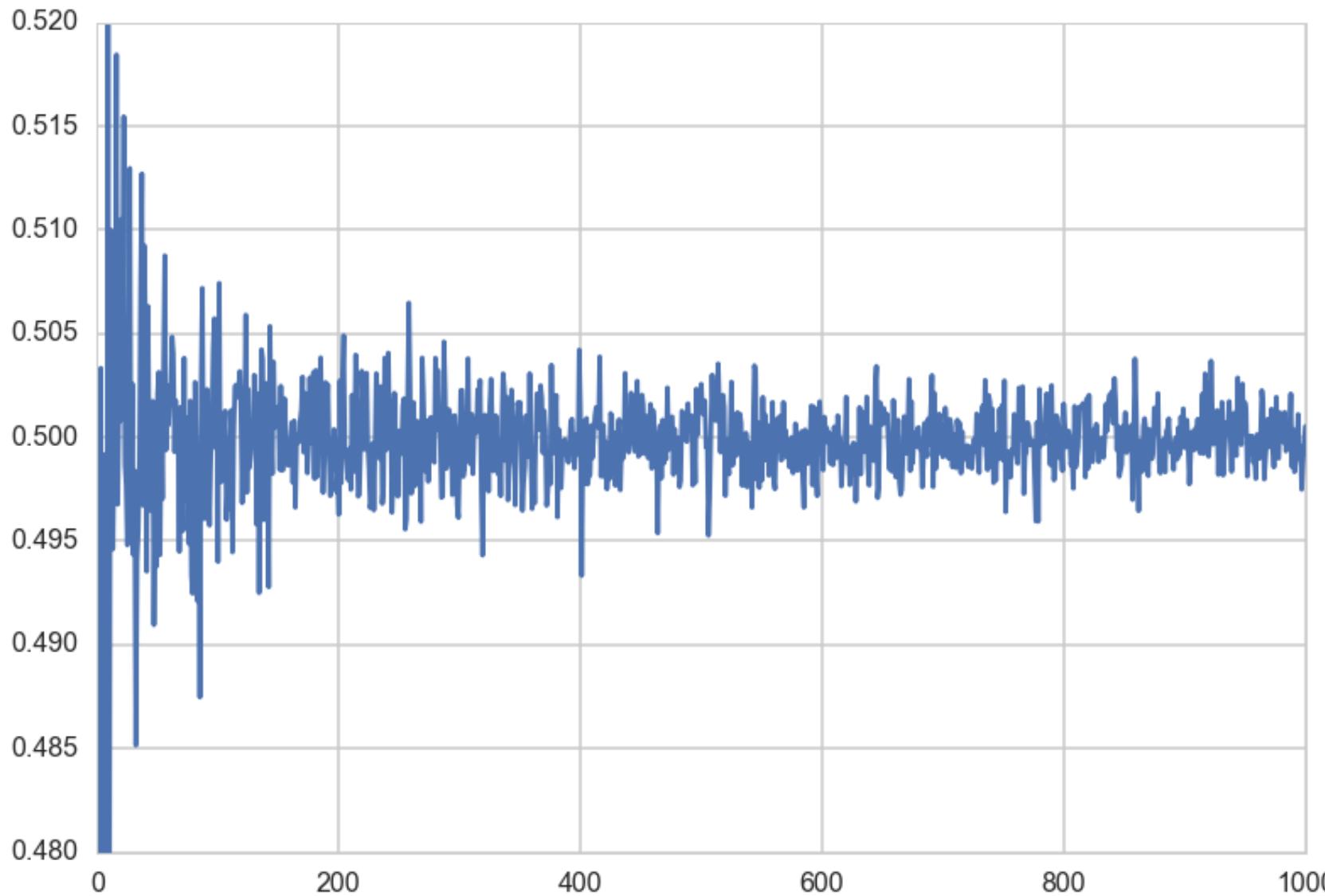
Variance of the estimate



M replications of N coin tosses



sample means: 200 replications of N coin tosses



$$E_{\{R\}}(N \bar{x}) = E_{\{R\}}(x_1 + x_2 + \dots + x_N) = E_{\{R\}}(x_1) + E_{\{R\}}(x_2) + \dots + E_{\{R\}}(x_N)$$

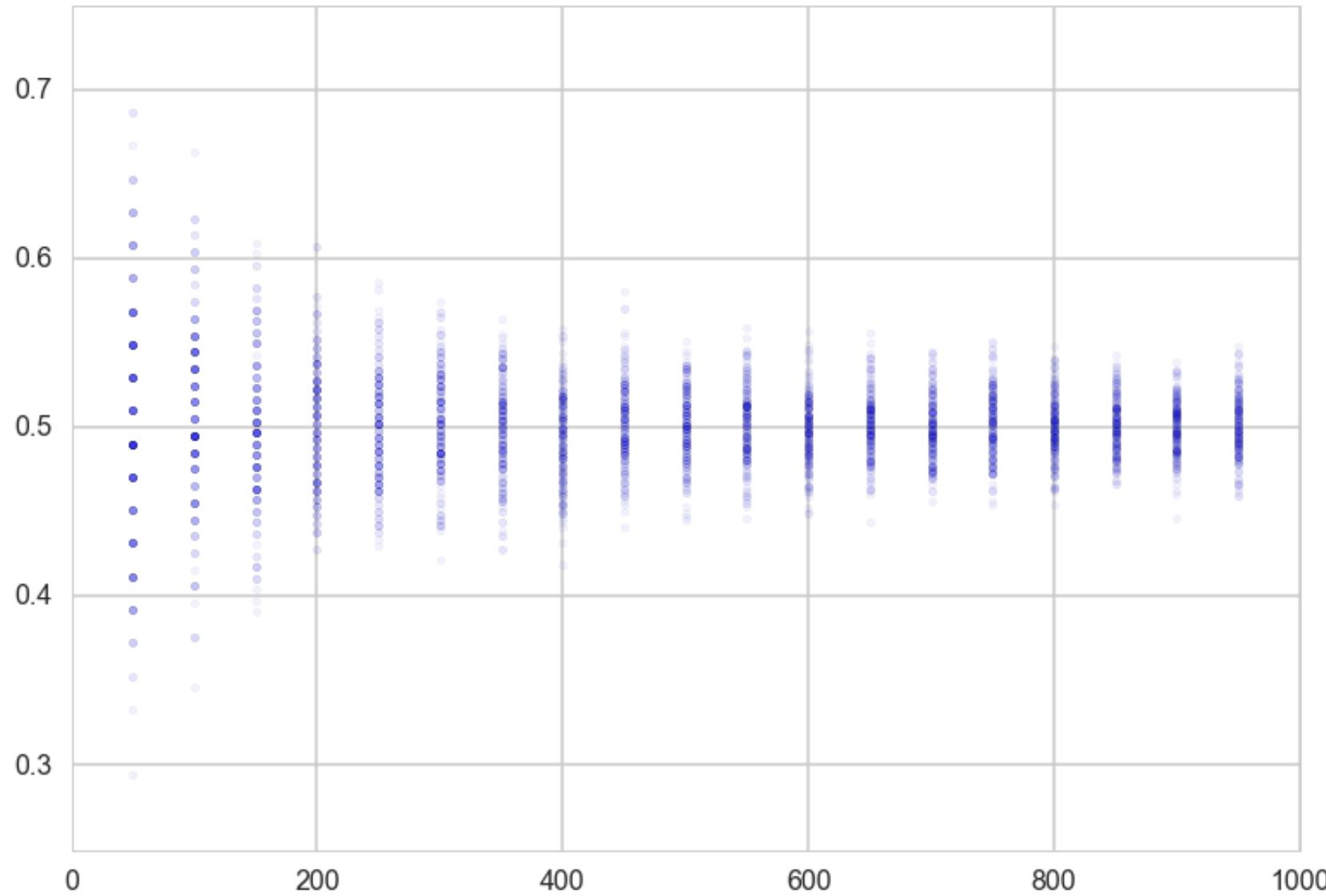
In limit $M \rightarrow \infty$ of replications, each of the expectations in RHS can be replaced by the population mean μ using the law of large numbers! Thus:

$$E_{\{R\}}(N \bar{x}) = N \mu$$

$$E_{\{R\}}(\bar{x}) = \mu$$

In limit $M \rightarrow \infty$ of replications the expectation value of the sample means converges to the population mean.

Distribution of Sample Means



Now let underlying distribution have well defined mean μ AND a well defined variance σ^2 .

$$V_{\{R\}}(N \bar{x}) = V_{\{R\}}(x_1 + x_2 + \dots + x_N) = V_{\{R\}}(x_1) + V_{\{R\}}(x_2) + \dots + V_{\{R\}}(x_N)$$

Now in limit $M \rightarrow \infty$, each of the variances in the RHS can be replaced by the population variance using the law of large numbers!
Thus:

$$V_{\{R\}}(N \bar{x}) = N \sigma^2$$

$$V(\bar{x}) = \frac{\sigma^2}{N}$$

The Central Limit Theorem (CLT)

Let x_1, x_2, \dots, x_n be a sequence of IID values from a random variable X . Suppose that X has the finite mean μ AND finite variance σ^2 . Then:

$$S_n = \frac{1}{n} \sum_{i=1}^n x_i, \text{ converges to}$$

$$S_n \sim N\left(\mu, \frac{\sigma^2}{n}\right) \text{ as } n \rightarrow \infty.$$

Meaning

- weight-watchers' study of 1000 people, average weight is 150 lbs with σ of 30lbs.
- Randomly choose many samples of 100 people each, the mean weights of those samples would cluster around 150lbs with a standard error of 3lbs.
- a different sample of 100 people with an average weight of 170lbs would be more than 6 standard errors beyond the population mean.

Back to Monte Carlo

We want to calculate:

$$S_n(f) = \frac{1}{n} \sum_{i=1}^n f(x_i)$$

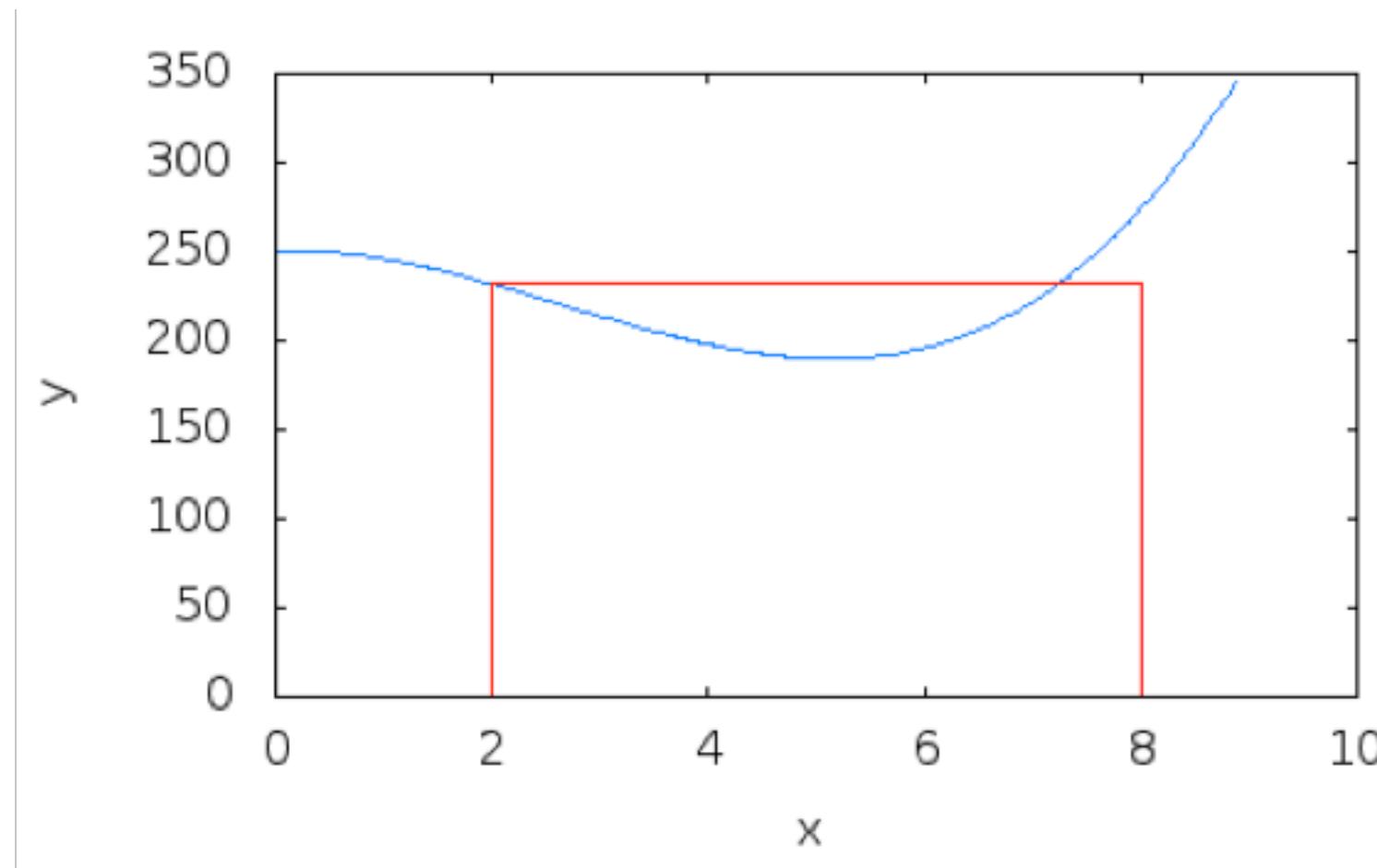
- Whatever $V[f(X)]$ is, the variance of the sampling distribution of the mean goes down as $1/n$
- Thus s goes down as $1/\sqrt{n}$

Why is this important?

- In higher dimensions d , the CLT still holds and the error still scales as $\frac{1}{\sqrt{n}}$.
- How does this compete with numerical integration? For $n = N^{1/d}$:
 - left or right rule: $\propto 1/n$, Midpoint rule: $\propto 1/n^2$
 - Trapezoid: $\propto 1/n^2$, Simpson: $\propto 1/n^4$

Basic Numerical Integration idea

(from wikipedia)

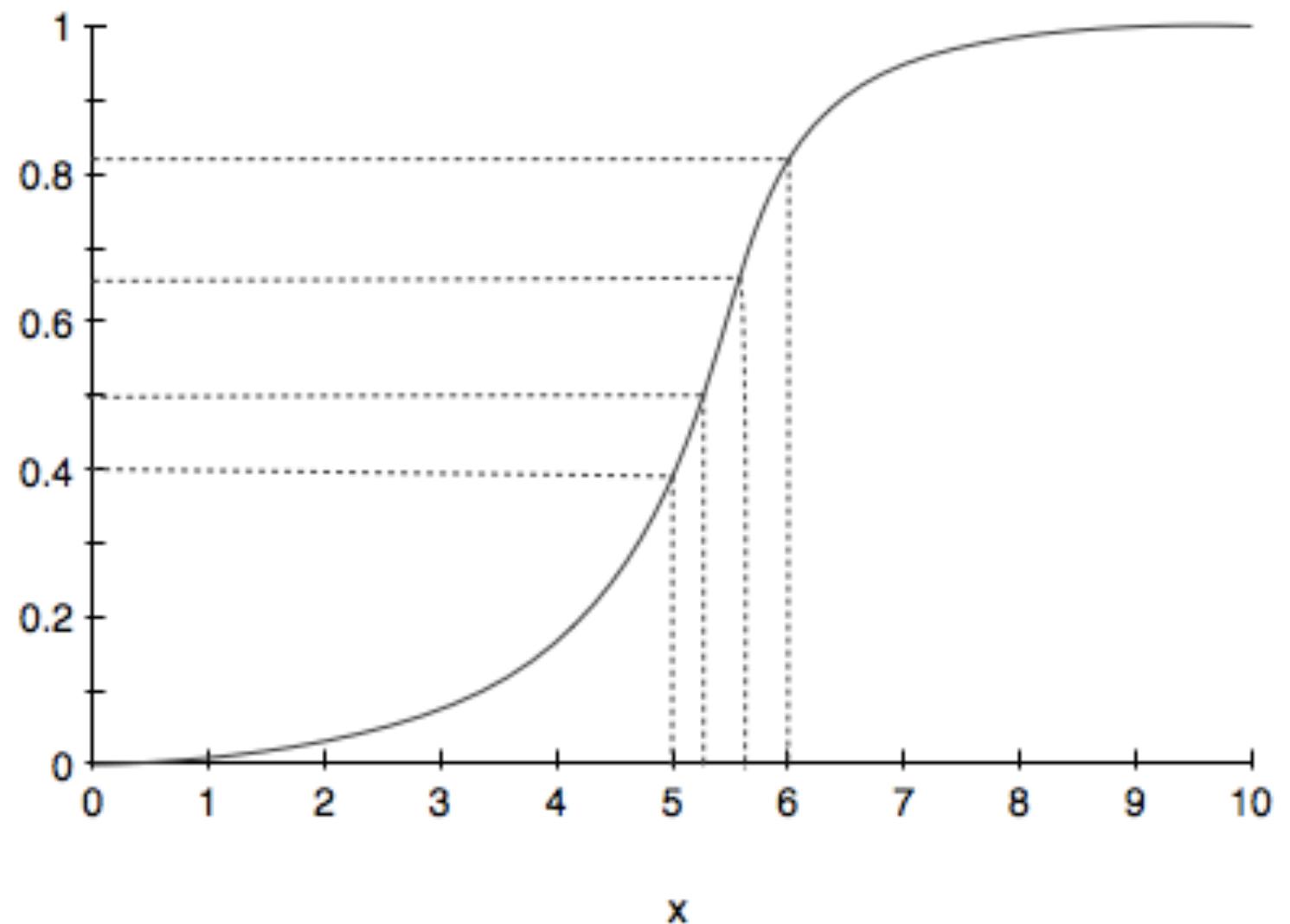
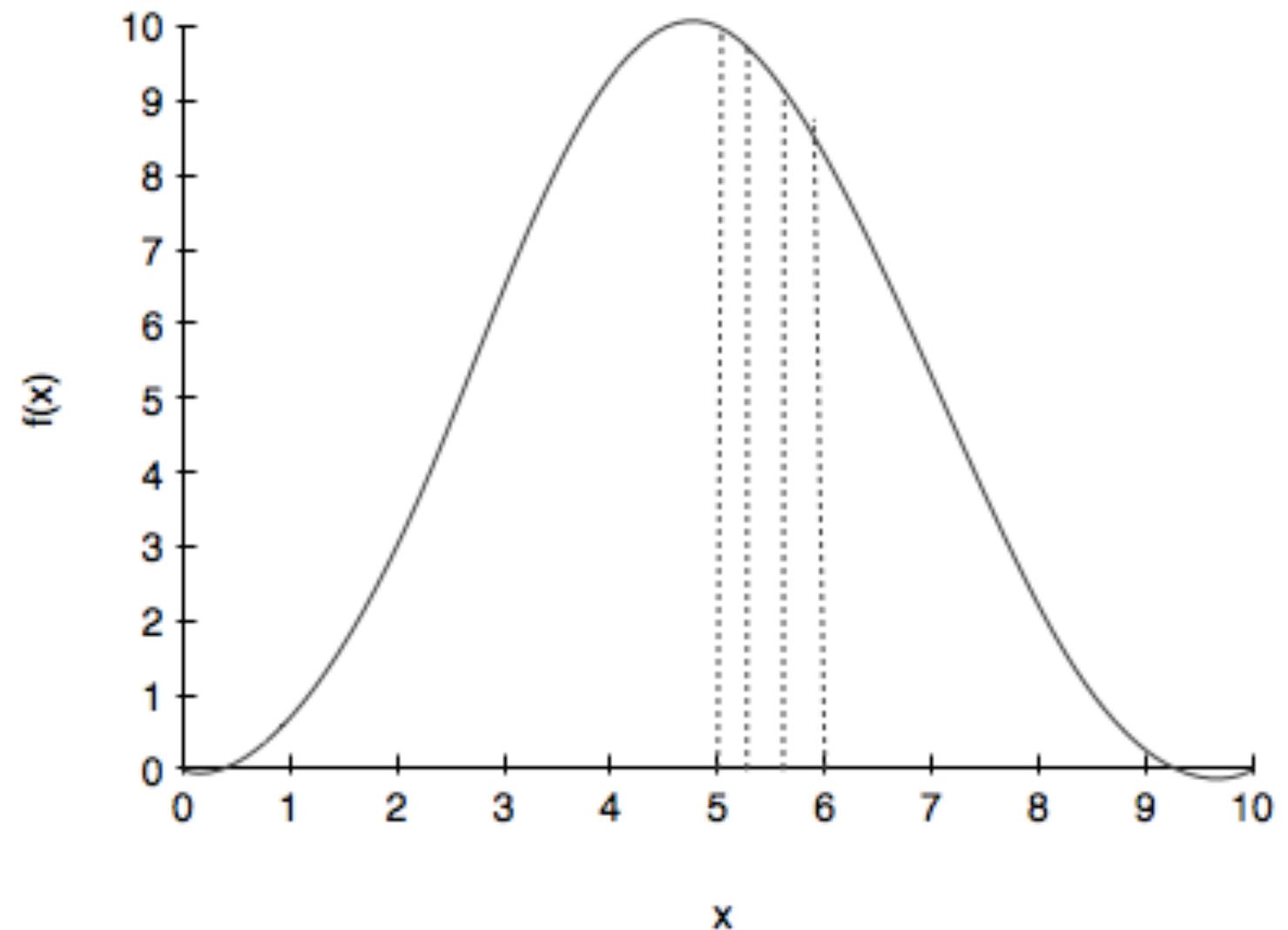


Soon

In order to calculate expectations, do integrals, and do statistics,
we must learn how to do

SAMPLING

A taste: Inverse transform



algorithm

The CDF F must be invertible!

1. get a uniform sample u from $\text{Unif}(0, 1)$
2. solve for x yielding a new equation $x = F^{-1}(u)$ where F is the CDF of the distribution we desire.
3. repeat.

Example: exponential

pdf: $f(x) = \frac{1}{\lambda}e^{-x/\lambda}$ for $x \geq 0$ and $f(x) = 0$ otherwise.

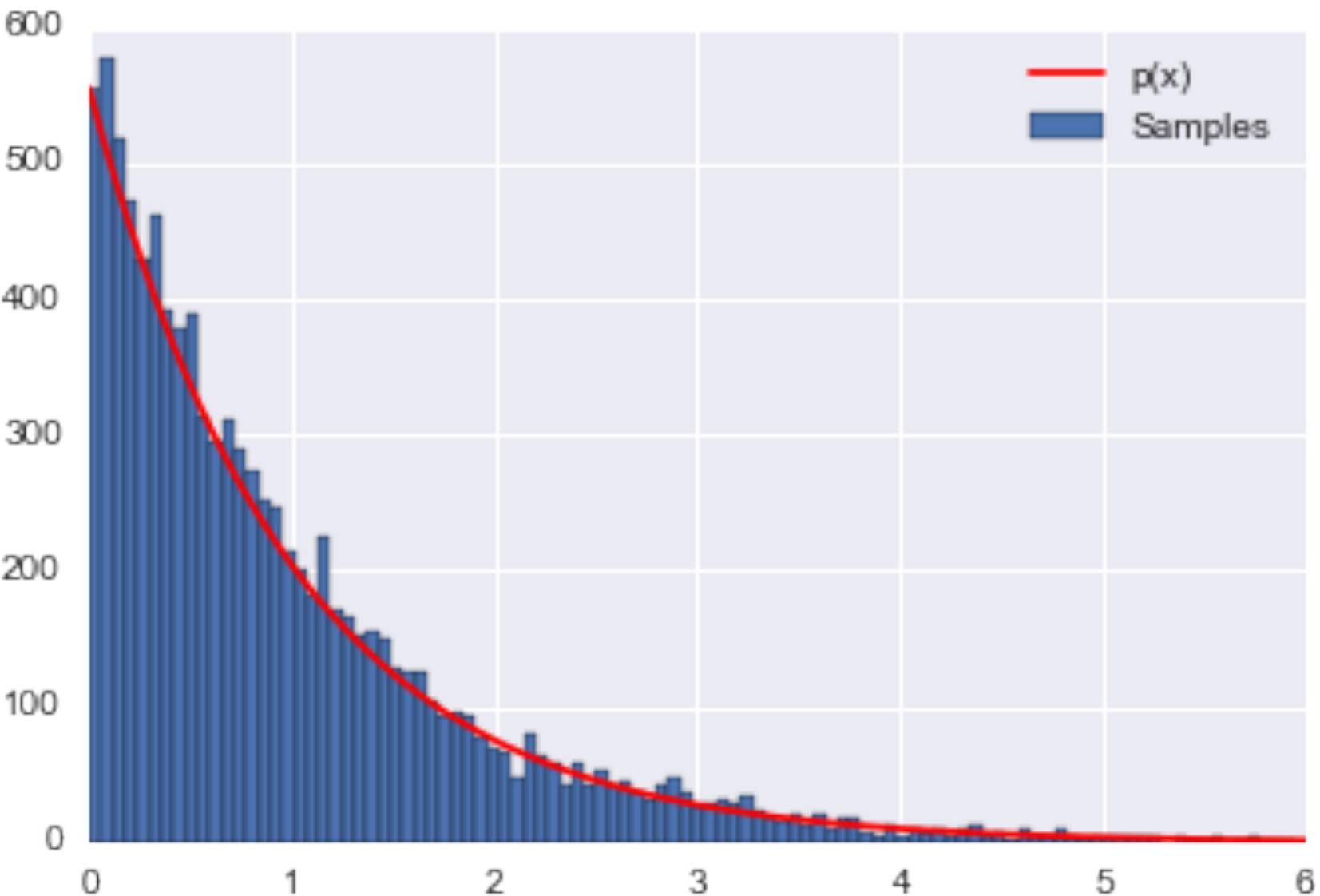
$$u = \int_0^x \frac{1}{\lambda}e^{-x'/\lambda} dx' = 1 - e^{-x/\lambda}$$

Solving for x

$$x = -\lambda \ln(1 - u)$$

code

```
p = lambda x: np.exp(-x)
CDF = lambda x: 1-np.exp(-x)
invCDF = lambda r: -np.log(1-r) # invert the CDF
xmin = 0 # the lower limit of our domain
xmax = 6 # the upper limit of our domain
rmin = CDF(xmin)
rmax = CDF(xmax)
N = 10000
# generate uniform samples in our range then invert the CDF
# to get samples of our target distribution
R = np.random.uniform(rmin, rmax, N)
X = invCDF(R)
hinfo = np.histogram(X, 100)
plt.hist(X, bins=100, label=u'Samples');
# plot our (normalized) function
xvals=np.linspace(xmin, xmax, 1000)
plt.plot(xvals, hinfo[0][0]*p(xvals), 'r', label=u'p(x)')
plt.legend()
```



Hit or miss

- Generate samples from a uniform distribution with support on the rectangle
- See how many fall below $y(x)$ at a specific x sliver.

This is the basic idea behind rejection sampling