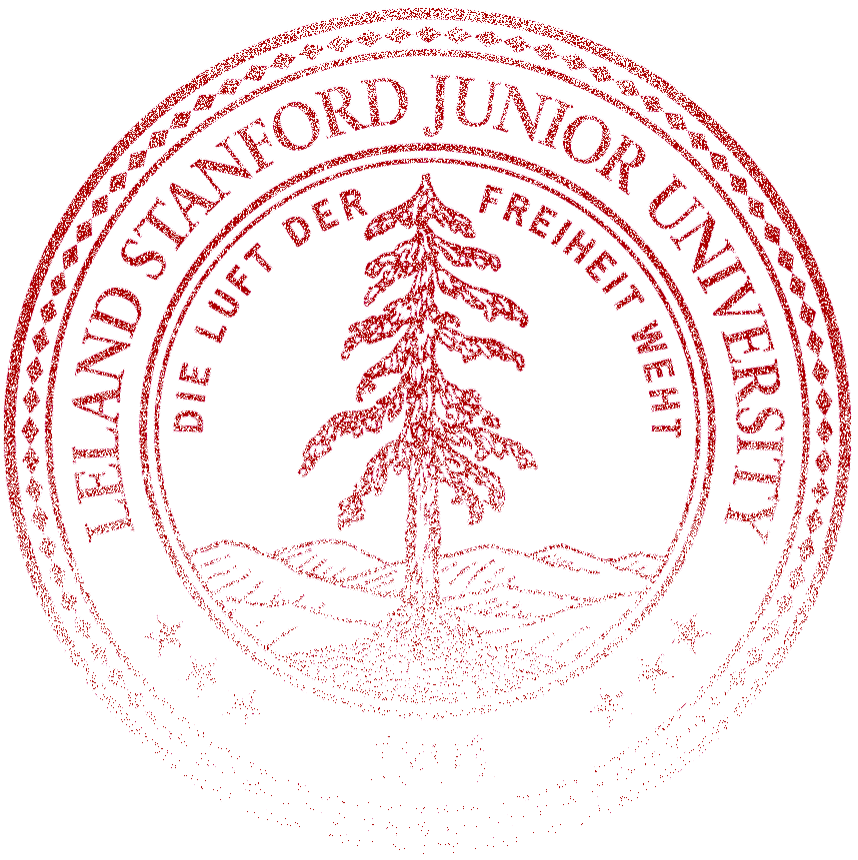
Course Reader for CS109



CS109
Department of Computer Science
Stanford University
Jan 2023
V 0.9

Get Started

Notable Updates Fall 2023:

1. General Inclusion-Exclusion. *Oct 7th 2023*
2. Core Probability Reference. *Oct 7th 2023*

Acknowledgements: This book was written by Chris Piech for Stanford's CS109 course, Probability for Computer scientists. The course was originally designed by Mehran Sahami and followed the Sheldon Ross book Probability Theory from which we take inspiration. The course has since been taught by Lisa Yan, Jerry Cain and David Varodayan and their ideas and feedback have improved this reader.

This course reader is open to contributions. Want to make your mark? Keen to fix a typo? Download the github project and publish a pull request. We will credit all contributors.

Folks who have contributed to editing the book: GitHub Contributors. This includes Logan Bhamidipaty, Jonatan Pérez, Bobby Abraham, Tim Gianitsos, Yogi the Curious, Thanawan "Ly-Ly" Atchariyachanvanit, Kunal

Notation Reference

Core Probability

Notation	Meaning
E or F	Capital letters can denote events
A or B	Sometimes they denote sets
$\|E\|$	Size of an event or set
E^C	Complement of an event or set
EF	And of events (aka intersection)
E and F	And of events (aka intersection)
$E \cap F$	And of events (aka intersection)
E or F	Or of events (aka union)
$E \cup F$	Or of events (aka union)
$\text{count}(E)$	The number of times that E occurs
$\text{P}(E)$	The probability of an event E
$\text{P}(E\|F)$	The conditional probability of an event E given F
$\text{P}(E, F)$	The probability of event E and F
$\text{P}(E\|F, G)$	The conditional probability of an event E given both F and G
$n!$	n factorial
$\binom{n}{k}$	Binomial coefficient
$\binom{n}{r_1, r_2, r_3}$	Multinomial coefficient

Random Variables

Notation	Meaning
x or y or i	Lower case letters denote regular variables
X or Y	Capital letters are used to denote random variables
K	Capital K is reserved for constants
$\text{E}[X]$	Expectation of X

Notation	Meaning
$\text{Var}(X)$	Variance of X
$\text{P}(X = x)$	Probability mass function (PMF) of X, evaluated at x
$\text{P}(x)$	Probability mass function (PMF) of X, evaluated at x
$f(X = x)$	Probability density function (PDF) of X, evaluated at x
$f(x)$	Probability density function (PDF) of X, evaluated at x
$f(X = x, Y = y)$	Joint probability density
$f(X = x \vert Y = y)$	Conditional probability density
$F_X(x)$ or $F(x)$	Cumulative distribution function (CDF) of X
IID	Independent and Identically Distributed

Parametric Distributions

Notation	Meaning
$X \sim \text{Bern}(p)$	X is a Bernoulli random variable
$X \sim \text{Bin}(n, p)$	X is a Binomial random variable
$X \sim \text{Poi}(p)$	X is a Poisson random variable
$X \sim \text{Geo}(p)$	X is a Geometric random variable
$X \sim \text{NegBin}(r, p)$	X is a Negative Binomial random variable
$X \sim \text{Uni}(a, b)$	X is a Uniform random variable
$X \sim \text{Exp}(\lambda)$	X is a Exponential random variable
$X \sim \text{Beta}(a, b)$	X is a Beta random variable

Core Probability Reference

Definition: Empirical Definition of Probability

The probability of any event E can be defined as:

$$\mathrm{P}(E) = \lim_{n \to \infty} \frac{\mathrm{count}(E)}{n}$$

Where $\mathrm{count}(E)$ is the number of times that E occured in n experiments.

Definition: Core Identities

For an event E and a sample space S

$0 \leq \mathrm{P}(E) \leq 1$	All probabilities are numbers between 0 and 1.
$\mathrm{P}(S) = 1$	All outcomes must be from the Sample Space.
$\mathrm{P}(E) = 1 - \mathrm{P}(E^{\mathrm{C}})$	The probability of an event from its complement.

Definition: Probability of Equally Likely Outcomes

If S is a sample space with equally likely outcomes, for an event E that is a subset of the outcomes in S:

$$\mathrm{P}(E) = \frac{\text{number of outcomes in } E}{\text{number of outcomes in } S} = \frac{|E|}{|S|}$$

Definition: Conditional Probability.
The probability of E given that (aka conditioned on) event F already happened:

$$\mathrm{P}(E|F) = \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)}$$

Definition: Probability of **or** with Mututally Exclusive Events
If two events E and F are mutually exclusive then the probability of E ***or*** F occurring is:

$$\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F)$$

For n events $E_1, E_2, \ldots E_n$ where each event is mutually exclusive of one another (in other words, no outcome is in more than one event). Then:

$$\mathrm{P}(E_1 \text{ or } E_2 \text{ or } \ldots \text{ or } E_n) = \mathrm{P}(E_1) + \mathrm{P}(E_2) + \cdots + \mathrm{P}(E_n) = \sum_{i=1}^{n} \mathrm{P}(E_i)$$

Definition: General Probability of **or** (Inclusion-Exclusion)
For any two events E and F:

$$\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F) - \mathrm{P}(E \text{ and } F)$$

For three events, E, F, and G the formula is:

$$\begin{aligned}\mathrm{P}(E \text{ or } F \text{ or } G) = \mathrm{P}(E) + \mathrm{P}(F) + \mathrm{P}(G) \\ - \mathrm{P}(E \text{ and } F) - \mathrm{P}(E \text{ and } G) - P(F \text{ and } G) \\ + \mathrm{P}(E \text{ and } F \text{ and } G)\end{aligned}$$

For more than three events see the chapter of probability of or.

Definition: Probability of **and** for Independent Events.
If two events: E, F are independent then the probability of E ***and*** F occurring is:

$$\mathrm{P}(E \text{ and } F) = \mathrm{P}(E) \cdot \mathrm{P}(F)$$

For n events $E_1, E_2, \ldots E_n$ that are independent of one another:

$$\mathrm{P}(E_1 \text{ and } E_2 \text{ and } \ldots \text{ and } E_n) = \prod_{i=1}^{n} \mathrm{P}(E_i)$$

Definition: General Probability of **and** (The Chain Rule)
For any two events E and F:

$$\mathrm{P}(E \text{ and } F) = \mathrm{P}(E|F) \cdot \mathrm{P}(F)$$

For n events $E_1, E_2, \ldots E_n$:

$$\begin{aligned}\mathrm{P}(E_1 \text{ and } E_2 \text{ and } \ldots \text{ and } E_n) = &\mathrm{P}(E_1) \cdot \mathrm{P}(E_2|E_1) \cdot \mathrm{P}(E_3|E_1 \text{ and } E_2) \ldots \\ &\mathrm{P}(E_n|E_1 \ldots E_{n-1})\end{aligned}$$

Definition: The Law of Total Probability
For any two events E and F:

$$\begin{aligned}\mathrm{P}(E) &= \mathrm{P}(E \text{ and } F) + \mathrm{P}(E \text{ and } F^{\mathrm{C}}) \\ &= \mathrm{P}(E|F)\,\mathrm{P}(F) + \mathrm{P}(E|F^{\mathrm{C}})\,\mathrm{P}(F^{\mathrm{C}})\end{aligned}$$

For mutually exclusive events: $B_1, B_2, \ldots B_n$ such that every outcome in the sample space falls into one of those events:

$$\begin{aligned}\mathrm{P}(E) &= \sum_{i=1}^{n} \mathrm{P}(E \text{ and } B_i) && \text{Extension of our observation} \\ &= \sum_{i=1}^{n} \mathrm{P}(E|B_i)\,\mathrm{P}(B_i) && \text{Using chain rule on each term}\end{aligned}$$

Definition: Bayes' Theorem
The most common form of Bayes' Theorem is **Bayes' Theorem Classic**:

$$\mathrm{P}(B|E) = \frac{\mathrm{P}(E|B) \cdot \mathrm{P}(B)}{\mathrm{P}(E)}$$

Bayes' Theorem combined with the Law of Total Probability:

$$\mathrm{P}(B|E) = \frac{\mathrm{P}(E|B) \cdot \mathrm{P}(B)}{\mathrm{P}(E|B) \cdot \mathrm{P}(B) + \mathrm{P}(E|B^{\mathrm{C}}) \cdot \mathrm{P}(B^{\mathrm{C}})}$$

Random Variable Reference

Discrete Random Variables

Bernoulli Random Variable

Notation: $X \sim \text{Bern}(p)$

Description: A boolean variable that is 1 with probability p

Parameters: p, the probability that $X = 1$.

Support: x is either 0 or 1

PMF equation: $$\text{P}(X = x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}$$

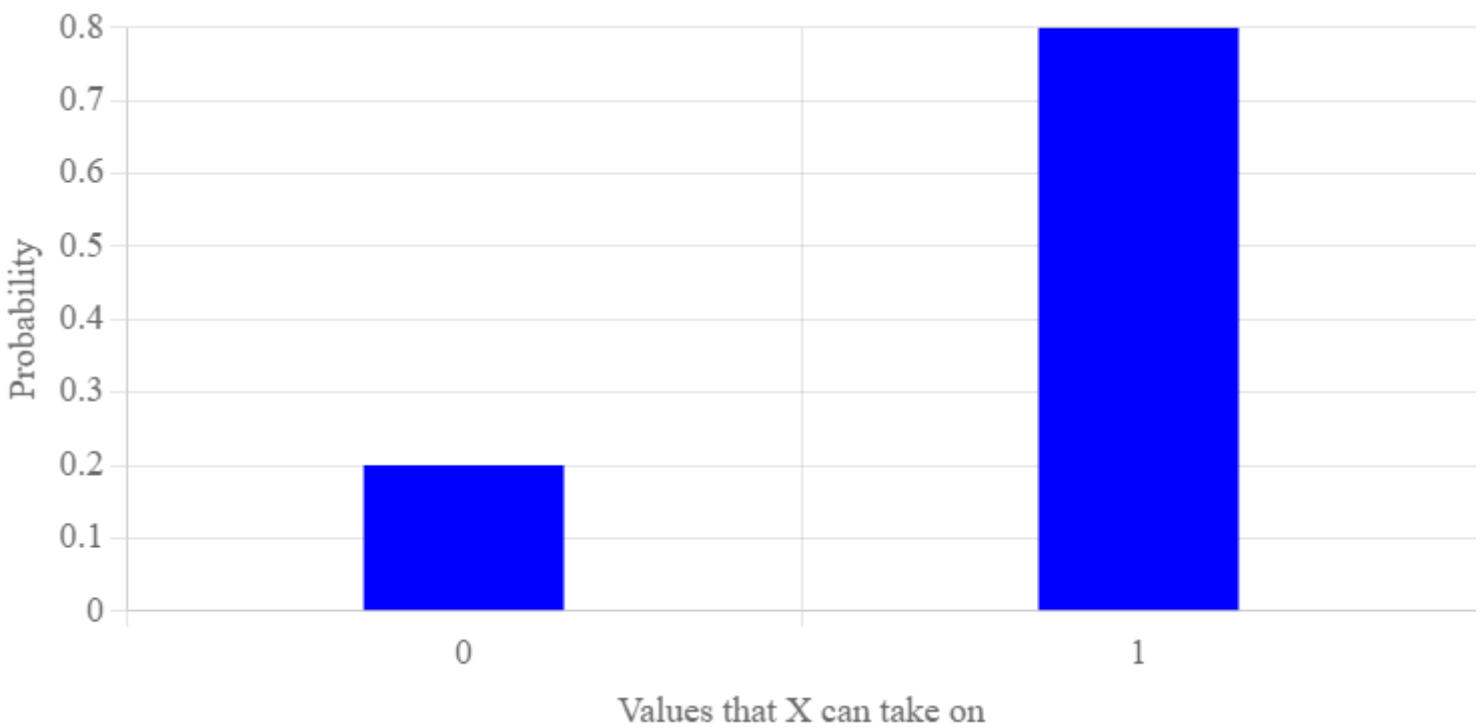
PMF (smooth): $\text{P}(X = x) = p^x (1-p)^{1-x}$

Expectation: $\text{E}[X] = p$

Variance: $\text{Var}(X) = p(1-p)$

PMF graph:

Parameter p: 0.80



Binomial Random Variable

Notation: $X \sim \text{Bin}(n, p)$

Description: Number of "successes" in n identical, independent experiments each with probability of success p.

Parameters: $n \in \{0, 1, \ldots\}$, the number of experiments.
$p \in [0, 1]$, the probability that a single experiment gives a "success".

Support: $x \in \{0, 1, \ldots, n\}$

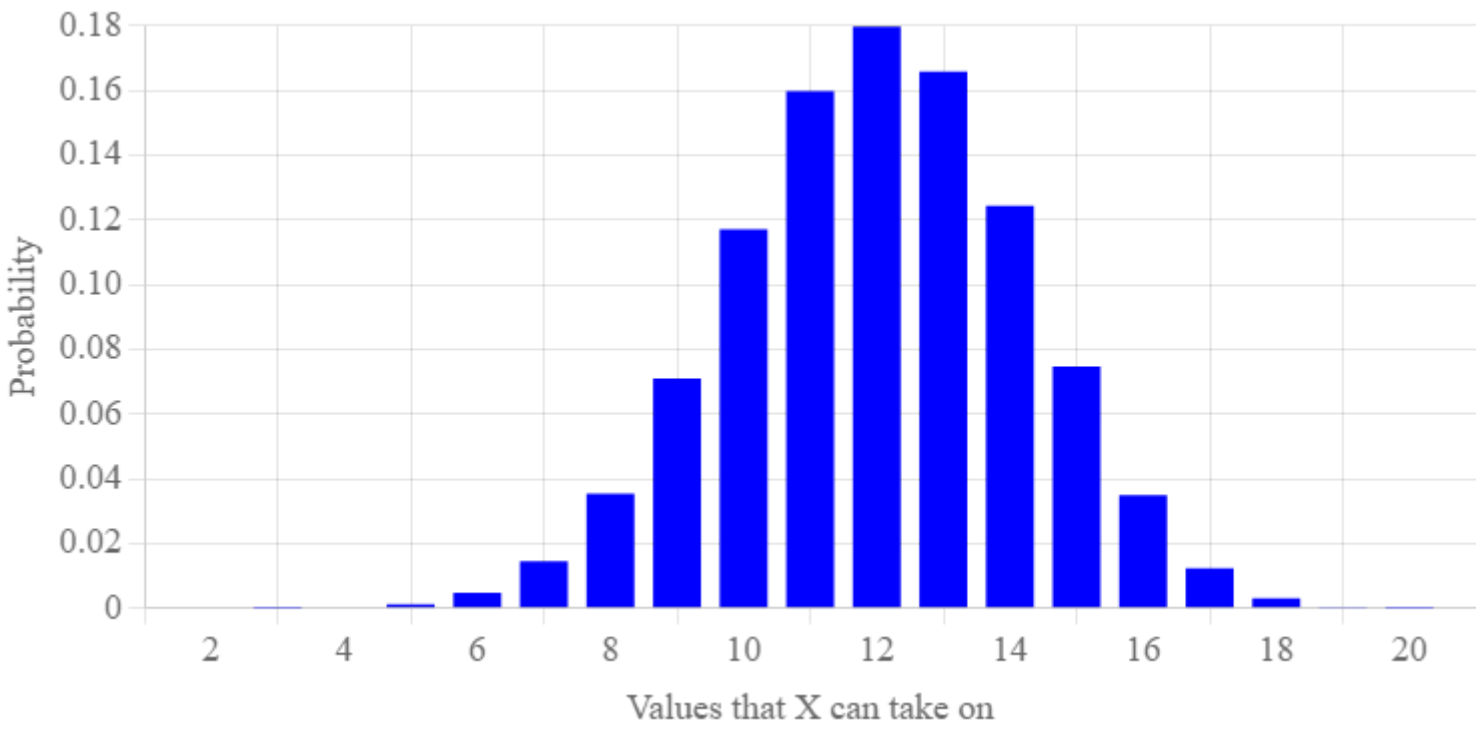
PMF equation: $$\text{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

Expectation: $\text{E}[X] = n \cdot p$

Variance: $\text{Var}(X) = n \cdot p \cdot (1-p)$

PMF graph:

Parameter n: 20 Parameter p: 0.60



Poisson Random Variable

Notation: $X \sim \text{Poi}(\lambda)$

Description: Number of events in a fixed time frame if (a) the events occur with a constant mean rate and (b) they occur independently of time since last event.

Parameters: $\lambda \in \{0, 1, \ldots\}$, the constant average rate.

Support: $x \in \{0, 1, \ldots\}$

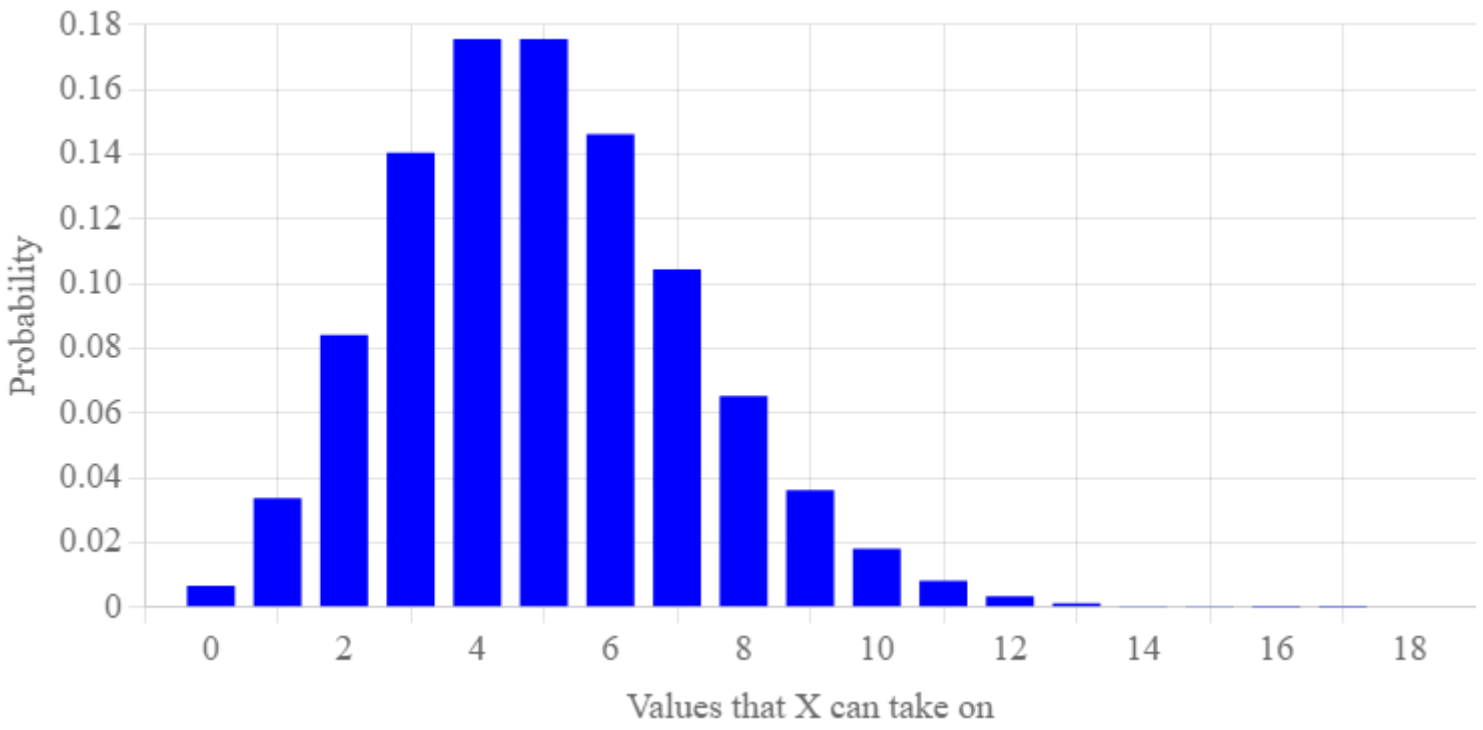
PMF equation: $\text{P}(X = x) = \dfrac{\lambda^x e^{-\lambda}}{x!}$

Expectation: $\text{E}[X] = \lambda$

Variance: $\text{Var}(X) = \lambda$

PMF graph:

Parameter λ: 5



Geometric Random Variable

Notation: $X \sim \text{Geo}(p)$

Description: Number of experiments until a success. Assumes independent experiments each with probability of success p.

Parameters: $p \in [0, 1]$, the probability that a single experiment gives a "success".

Support: $x \in \{1, \ldots, \infty\}$

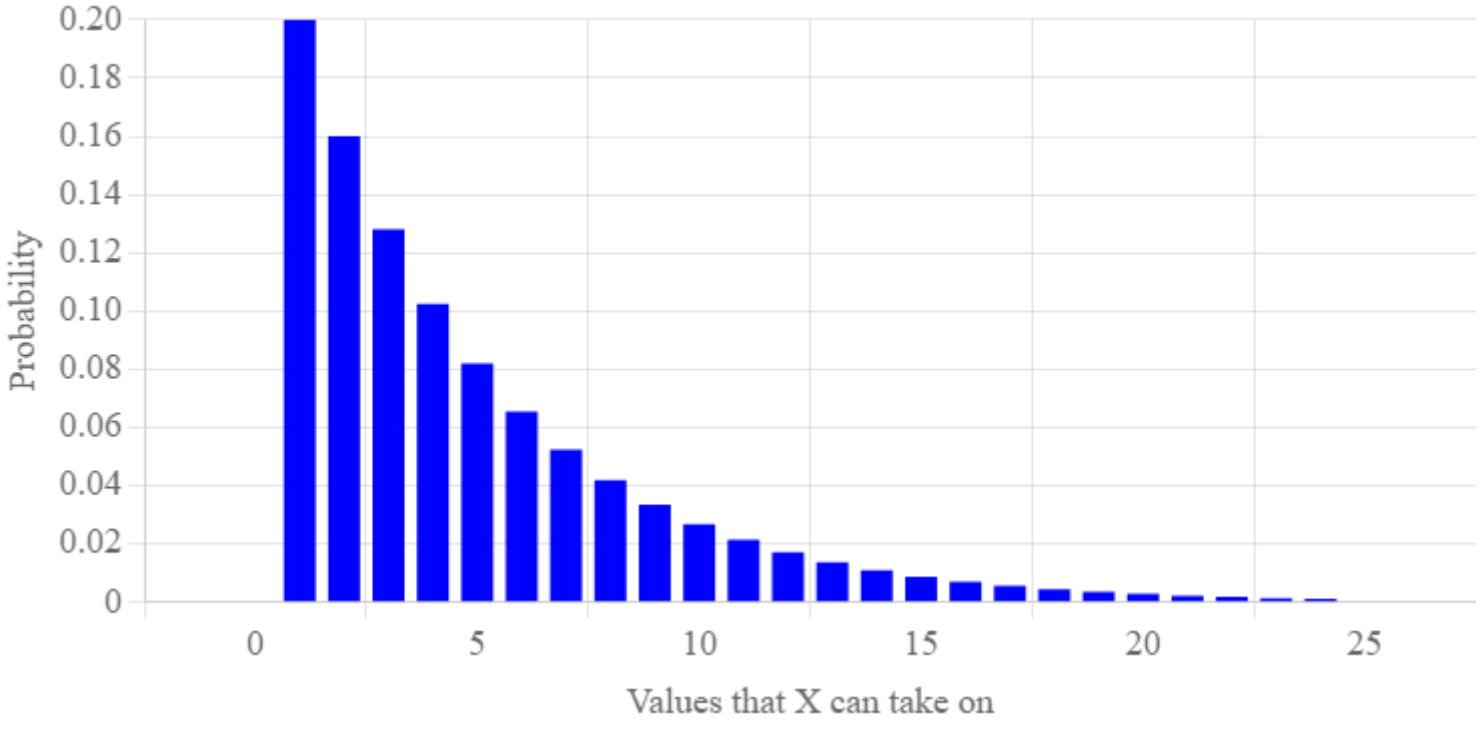
PMF equation: $\text{P}(X = x) = (1 - p)^{x-1} p$

Expectation: $\text{E}[X] = \frac{1}{p}$

Variance: $\text{Var}(X) = \frac{1-p}{p^2}$

PMF graph:

Parameter p: 0.20

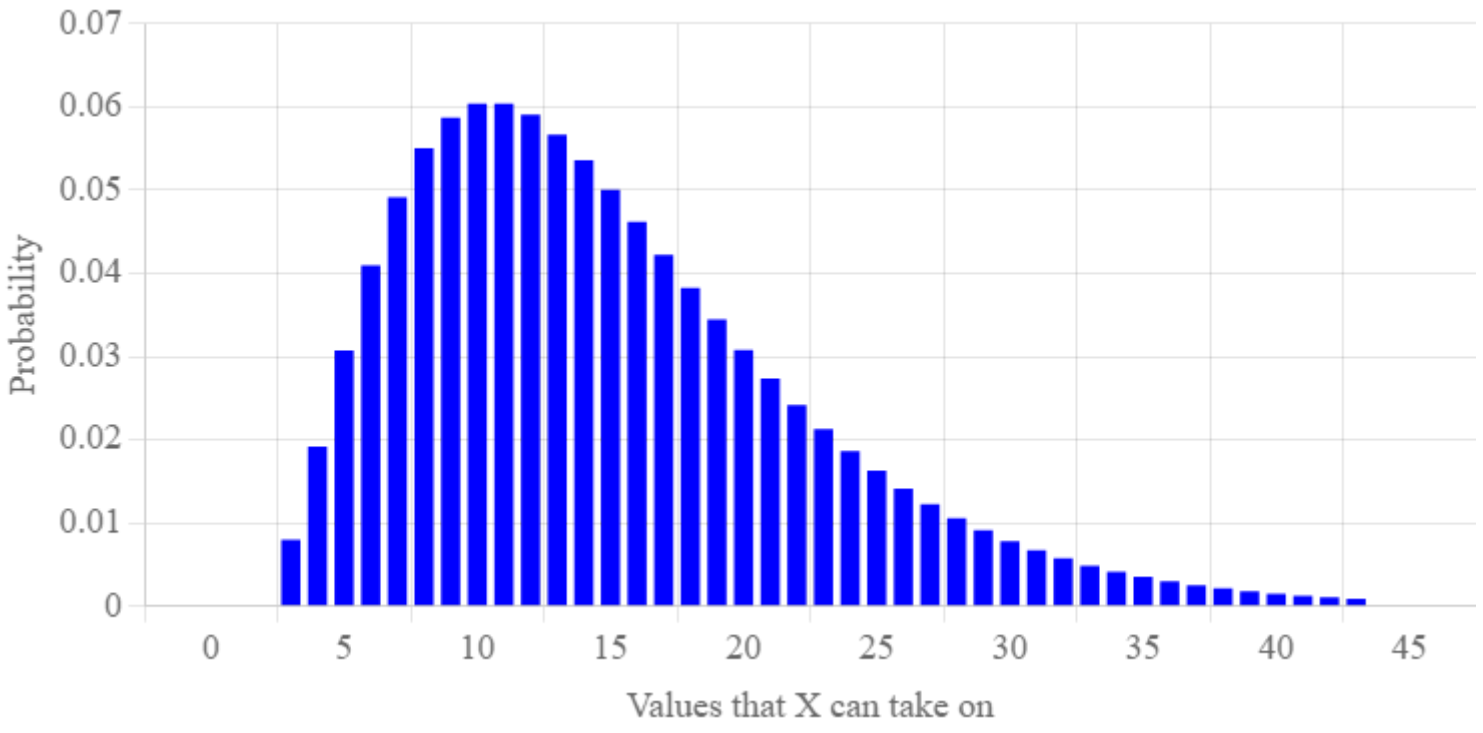


Negative Binomial Random Variable

Notation:	$X \sim \text{NegBin}(r, p)$
Description:	Number of experiments until r successes. Assumes each experiment is independent with probability of success p.
Parameters:	$r > 0$, the number of success we are waiting for. $p \in [0, 1]$, the probability that a single experiment gives a "success".
Support:	$x \in \{r, \ldots, \infty\}$
PMF equation:	$\text{P}(X = x) = \binom{x-1}{r-1} p^r (1-p)^{x-r}$
Expectation:	$\text{E}[X] = \frac{r}{p}$
Variance:	$\text{Var}(X) = \frac{r \cdot (1-p)}{p^2}$

PMF graph:

Parameter r: 3 Parameter p: 0.20



Continuous Random Variables

Uniform Random Variable

Notation:	$X \sim \text{Uni}(\alpha, \beta)$
Description:	A continuous random variable that takes on values, with equal likelihood, between α and β

Parameters: $\alpha \in \mathbb{R}$, the minimum value of the variable.
$\beta \in \mathbb{R}, \beta > \alpha$, the maximum value of the variable.

Support: $x \in [\alpha, \beta]$

PDF equation: $f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{else} \end{cases}$

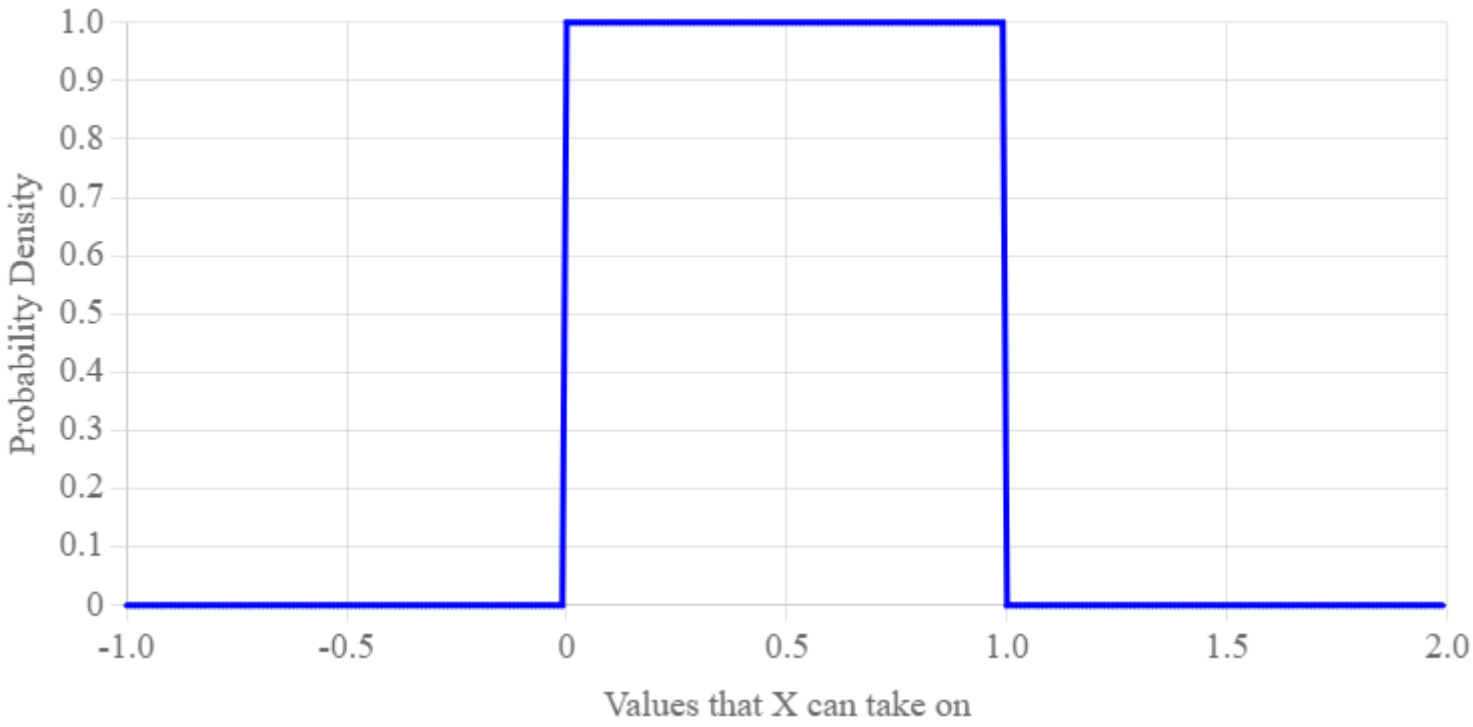
CDF equation: $F(x) = \begin{cases} \frac{x - \alpha}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{for } x < \alpha \\ 1 & \text{for } x > \beta \end{cases}$

Expectation: $\mathrm{E}[X] = \frac{1}{2}(\alpha + \beta)$

Variance: $\mathrm{Var}(X) = \frac{1}{12}(\beta - \alpha)^2$

PDF graph:

Parameter α: 0 Parameter β: 1



Exponential Random Variable

Notation: $X \sim \mathrm{Exp}(\lambda)$

Description: Time until next events if (a) the events occur with a constant mean rate and (b) they occur independently of time since last event.

Parameters: $\lambda \in \{0, 1, \ldots\}$, the constant average rate.

Support: $x \in \mathbb{R}^+$

PDF equation: $f(x) = \lambda e^{-\lambda x}$

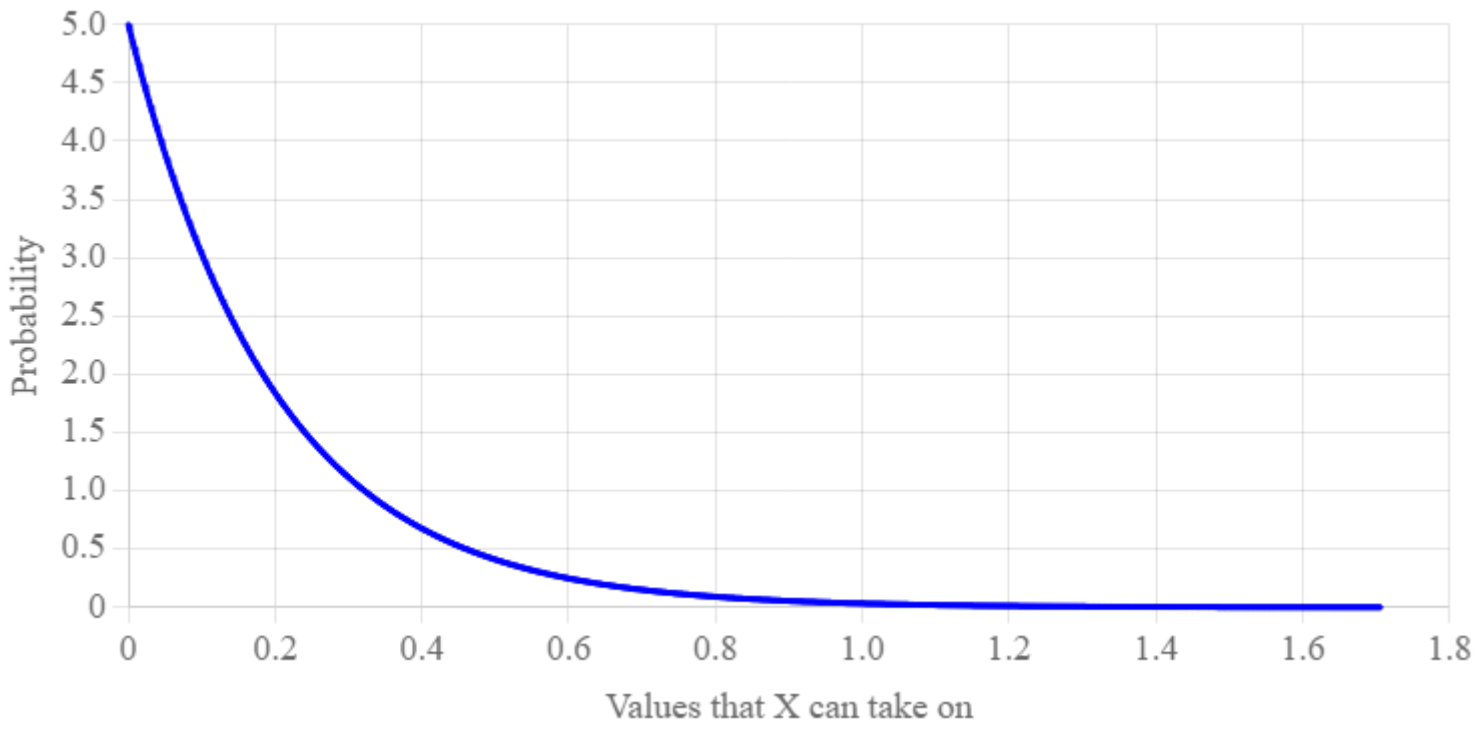
CDF equation: $F(x) = 1 - e^{-\lambda x}$

Expectation: $\mathrm{E}[X] = 1/\lambda$

Variance: $\mathrm{Var}(X) = 1/\lambda^2$

PDF graph:

Parameter λ: 5



Normal (aka Gaussian) Random Variable

Notation: $X \sim \mathrm{N}(\mu, \sigma^2)$

Description: A common, naturally occurring distribution.

Parameters: $\mu \in \mathbb{R}$, the mean.

$\sigma^2 \in \mathbb{R}$, the variance.

Support: $x \in \mathbb{R}$

PDF equation: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

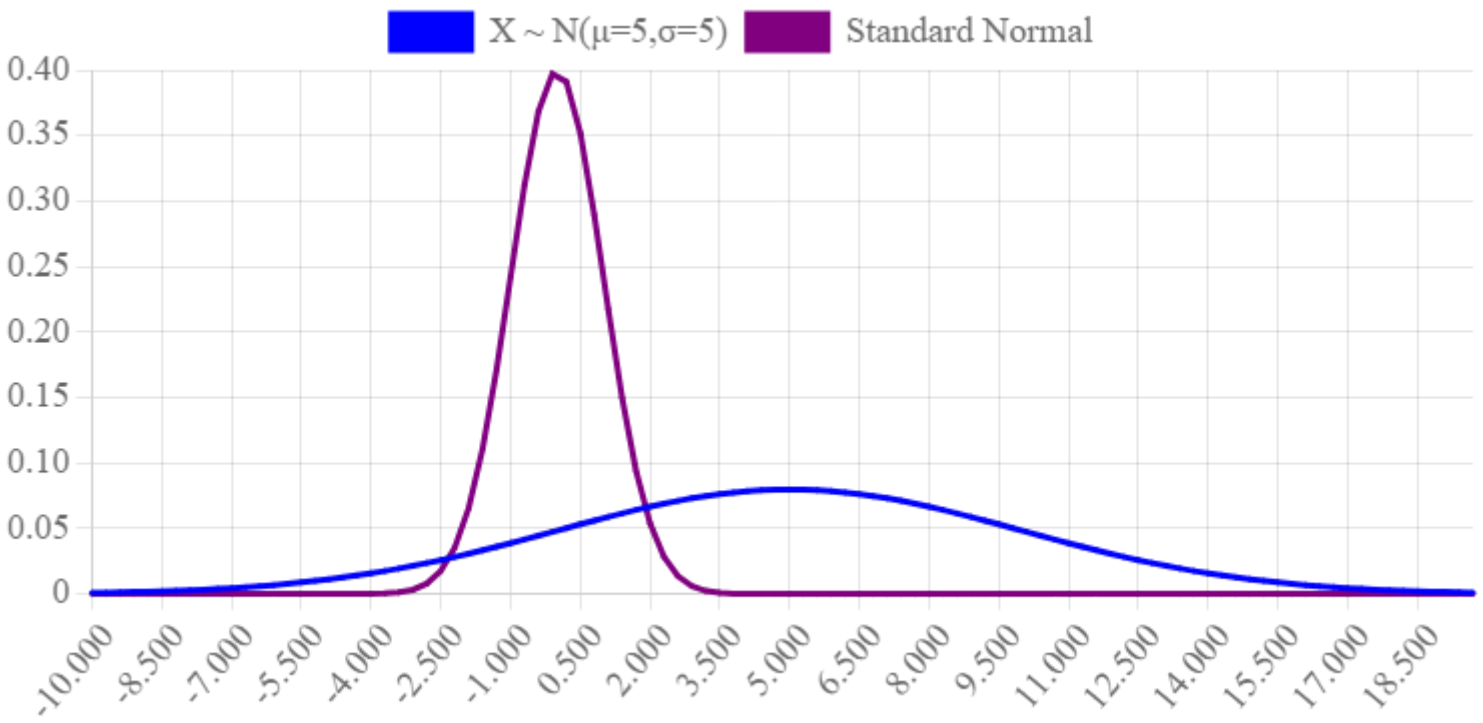
CDF equation: $F(x) = \phi(\frac{x-\mu}{\sigma})$ Where ϕ is the CDF of the standard normal

Expectation: $\mathrm{E}[X] = \mu$

Variance: $\mathrm{Var}(X) = \sigma^2$

PDF graph:

Parameter μ: 5 Parameter σ: 5



Beta Random Variable

Notation: $X \sim \mathrm{Beta}(a, b)$

Description: A belief distribution over the value of a probability p from a Binomial distribution after observing $a - 1$ successes and $b - 1$ fails.

Parameters: $a > 0$, the number successes + 1

$b > 0$, the number of fails + 1

Support: $x \in [0, 1]$

PDF equation: $f(x) = B \cdot x^{a-1} \cdot (1-x)^{b-1}$

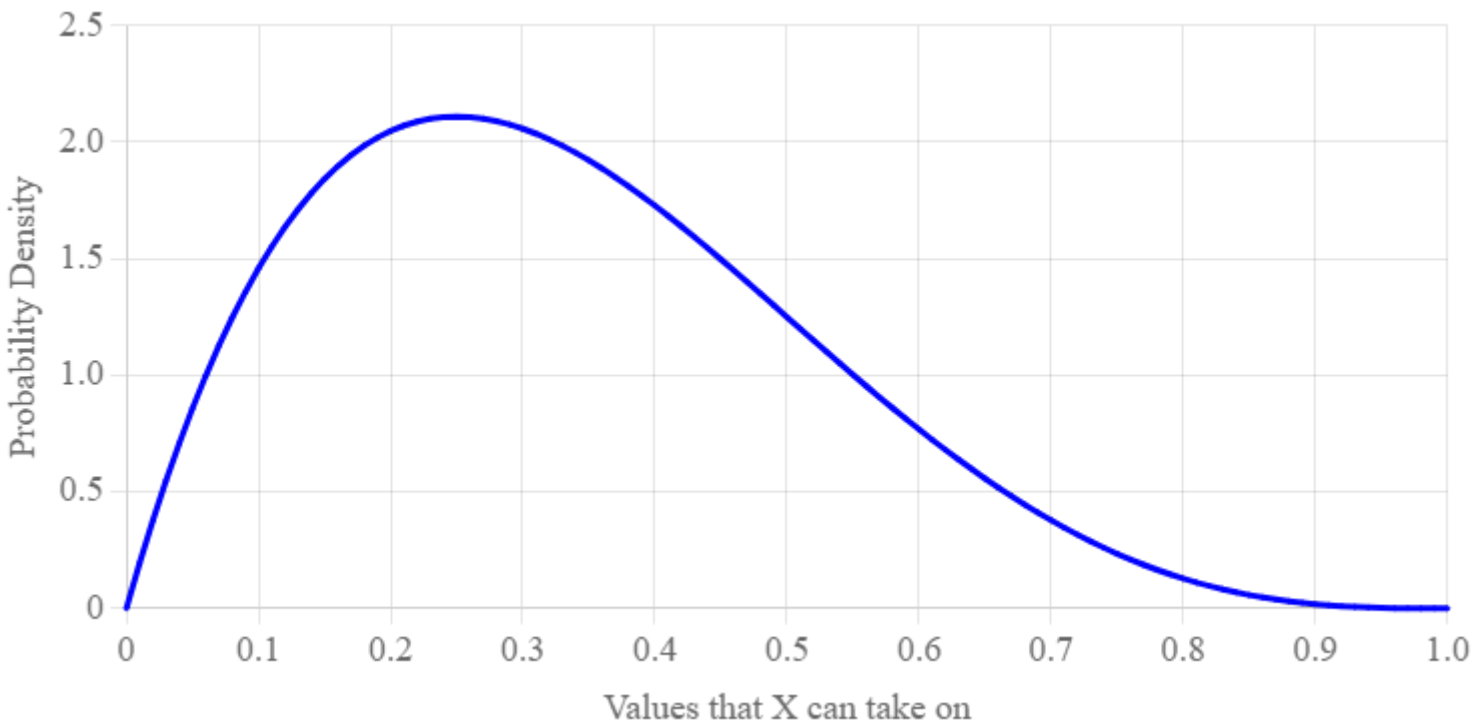
CDF equation: No closed form

Expectation: $\mathrm{E}[X] = \frac{a}{a+b}$

Variance: $\mathrm{Var}(X) = \frac{ab}{(a+b)^2(a+b+1)}$

PDF graph:

Parameter a: 2 Parameter b: 4



Python Reference

Factorial

Compute $n!$ as an integer. This example computes 20!:

```
import math
print(math.factorial(20))
```

Choose

As of Python 3.8, you can compute $\binom{n}{m}$ from the math module. This example computes $\binom{10}{5}$:

```
import math
print(math.comb(10, 5))
```

Natural Exponent

Calculate e^x. For example this computes e^3

```
import math
print(math.exp(3))
```

SciPy Stats Library

SciPy is a free and open source library for scientific computing that is built on top of NumPy. You may find it helpful to use SciPy to check the answers you obtain in the written section of your problem sets. NumPy has the capability of drawing samples from many common distributions (type `help(np.random)` in the python interpreter), but SciPy has the added capability of computing the probability of observing events, and it can perform computations directly on the probability mass/density functions.

Binomial

Make a Binomial Random variable X and compute its probability mass function (PMF) or cumulative density function (CDF). We love the scipy stats library because it defines all the functions you would care about for a random variable, including expectation, variance, and even things we haven't talked about in CS109, like entropy. This example declares $X \sim \text{Bin}(n = 10, p = 0.2)$. It calculates a few statistics on X. It then calculates $P(X = 3)$ and $P(X \leq 4)$. Finally it generates a few random samples from X:

```
from scipy import stats
X = stats.binom(10, 0.2) # Declare X to be a binomial random variable
print(X.pmf(3))          # P(X = 3)
print(X.cdf(4))          # P(X <= 4)
print(X.mean())          # E[X]
print(X.var())           # Var(X)
print(X.std())           # Std(X)
print(X.rvs())           # Get a random sample from X
print(X.rvs(10))         # Get 10 random samples form X
```

From a **terminal** you can always use the "help" command to see a full list of methods defined on a variable (or for a package):

```
from scipy import stats
X = stats.binom(10, 0.2) # Declare X to be a binomial random variable
help(X)                  # List all methods defined for X
```

Poisson

Make a Poisson Random variable Y. This example declares $Y \sim \text{Poi}(\lambda = 2)$. It then calculates $P(Y = 3)$:

```
from scipy import stats
Y = stats.poisson(2)  # Declare Y to be a poisson random variable
print(Y.pmf(3))       # P(Y = 3)
print(Y.rvs())        # Get a random sample from Y
```

Geometric

Make a Geometric Random variable X, the number of trials until a success. This example declares $X \sim \text{Geo}(p = 0.75)$:

```
from scipy import stats
X = stats.geom(0.75)  # Declare X to be a geometric random variable
print(X.pmf(3))       # P(X = 3)
print(X.rvs())        # Get a random sample from Y
```

Normal

Make a Normal Random variable A. This example declares $A \sim N(\mu = 3, \sigma^2 = 16)$. It then calculates $f_Y(0)$ and $F_Y(0)$. **Very Important!!!** In class, the second parameter to a normal was the variance (σ^2). In the scipy library, the second parameter is the standard deviation (σ):

```
import math
from scipy import stats
A = stats.norm(3, math.sqrt(16)) # Declare A to be a normal random variable
print(A.pdf(4))                  # f(3), the probability density at 3
print(A.cdf(2))                  # F(2), which is also P(Y < 2)
print(A.rvs())                   # Get a random sample from A
```

Exponential

Make an Exponential Random variable B. This example declares $B \sim \text{Exp}(\lambda = 4)$:

```
from scipy import stats
# `λ` is a common parameterization for the exponential,
#  but `scipy` uses `scale` which is `1/λ`
B = stats.expon(scale=1/4)
print(B.pdf(1))             # f(1), the probability density at 1
print(B.cdf(2))             # F(2) which is also P(B < 2)
print(B.rvs())              # Get a random sample from B
```

Beta

Make an Beta Random variable X. This example declares $X \sim \text{Beta}(\alpha = 1, \beta = 3)$:

```
from scipy import stats
X = stats.beta(1, 3)  # Declare X to be a beta random variable
print(X.pdf(0.5))     # f(0.5), the probability density at 1
print(X.cdf(0.7))     # F(0.7) which is also P(X < 0.7)
print(X.rvs())        # Get a random sample from X
```

Calculators

Factorial Calculator $n!$

n 10

`factorial(n)`

Combination Calculator $\binom{n}{k}$

n 10

k 6

`combination(n,k)`

Phi Calculator, $\Phi(x)$

x 0.7

`phi(x)`

Inverse Phi Calculator, $\Phi^{-1}(y)$

y 0.7

`inverse_phi(y)`

Norm CDF Calculator

x 0.0

mu 0

std 1

`norm.cdf(x, mu, std)`

Beta CDF Calculator

x 0.5

a 3

b 4

```
beta.cdf(x, a, b)
```

Counting

Although you may have thought you had a pretty good grasp on the notion of counting at the age of three, it turns out that you had to wait until now to learn how to really count. Aren't you glad you took this class now?! But seriously, counting is like the foundation of a house (where the house is all the great things we will do later in this book, such as machine learning). Houses are awesome. Foundations, on the other hand, are pretty much just concrete in a hole. But don't make a house without a foundation. It won't turn out well.

Counting with Steps

Definition: Step Rule of Counting (aka Product Rule of Counting)

If an experiment has two parts, where the first part can result in one of m outcomes and the second part can result in one of n outcomes regardless of the outcome of the first part, then the total number of outcomes for the experiment is $m \cdot n$.

Rewritten using set notation, the Step Rule of Counting states that if an experiment with two parts has an outcome from set A in the first part, where $|A| = m$, and an outcome from set B in the second part (where the number of outcomes in B is the same regardless of the outcome of the first part), where $|B| = n$, then the total number of outcomes of the experiment is $|A||B| = m \cdot n$.

Simple Example: Consider a hash table with 100 buckets. Two arbitrary strings are independently hashed and added to the table. How many possible ways are there for the strings to be stored in the table? Each string can be hashed to one of 100 buckets. Since the results of hashing the first string do not impact the hash of the second, there are 100 * 100 = 10,000 ways that the two strings may be stored in the hash table.

Peter Norvig, the author of the cannonical text book "Artificial Intelligence" made the following compelling point on why computer scientists need to know how to count. To start, lets set a baseline for a really big number: The number of atoms in the observable universe, often estimated to be around 10 to the 80th power (10^{80}). There certainly are a lot of atoms in the universe. As a leading expert said,

> "Space is big. Really big. You just won't believe how vastly, hugely, mind-bogglingly big it is. I mean, you may think it's a long way down the road to the chemist, but that's just peanuts to space." - Douglas Adams

This number is often used to demonstrate tasks that computers will never be able to solve. Problems can quickly grow to an absurd size, and we can understand why using the Step Rule of Counting.

There is an art project to display every possible picture. Surely that would take a long time, because there must be many possible pictures. But how many? We will assume the color model known as True Color, in which each pixel can be one of $2^{24} \approx 17$ million distinct colors.

How many distinct pictures can you generate from (a) a smart phone camera shown with 12 million pixels, (b) a grid with 300 pixels, and (c) a grid with just 12 pixels?

(a) 12 million pixels

(b) 300 pixels

(c) 12 pixels

Answer: We can use the step rule of counting. An image can be created one pixel at a time, step by step. Each time we choose a pixel you can select its color out of 17 million choices. An array of n pixels produces (17 million)^n different pictures. (17 million)$^{12} \approx 10^{86}$, so the tiny 12-pixel grid produces a million times more pictures than the number of atoms in the universe! How about the 300 pixel array? It can produce 10^{2167} pictures. You may think the number of atoms in the universe is big, but that's just peanuts to the number of pictures in a 300-pixel array. And 12M pixels? $10^{86696638}$ pictures.

Example: Unique states of Go

For example a Go board has 19 × 19 points where a user can place a stone. Each of the points can be empty or occupied by black or white stone. By the Step Rule of Counting, we can compute the number of unique board configurations.

In go there are 19x19 points. Each point can have a black stone, white stone, or no stone at all.

Here we are going to construct the board one point at a time, step by step. Each time we add a point we have a unique choice where we can decide to make the point one of three options: {Black, White, No Stone}. Using this construction we can apply the Step Rule of Counting. If there was only one point, there would be three unique board configurations. If there were four points you would have $3 \cdot 3 \cdot 3 \cdot 3 = 81$ unique combinations. In Go there are $3^{(19\times19)} \approx 10^{172}$ possible board positions. The way we constructed our board didn't take into account which ones were illegal by the rules of Go. It turns out that "only" about 10^{170} of those positions are legal. That is about the square of the number of atoms in the universe. In other words: if there was another universe of atoms for every single atom, only then would there be as many atoms in the universe as there are unique configurations of a Go board.

As a computer scientist this sort of result can be very important. While computers are powerful, an algorithm which needed to store each configuration of the board would not be a reasonable approach. No computer can store more information than atoms in the universe squared!

The above argument might leave you feeling like some problems are incredibly hard as a result of the product rule of counting. Let's take a moment to talk about how the product rule of counting can help! Most logarithmic time algorithms leverage this principle.

Imagine you are building a machine learning system that needs to learn from data and you want to synthetically generate 10 million unique data points for it. How many steps would you need to encode to get to 10 million? Assuming that at each step you have a binary choice, the number of unique data points you produce will be 2^n by the Step Rule of counting. If we chose n such that $\log_2 10,000,000 < n$. You would only need to encode $n = 24$ binary decisions.

Example: Rolling two dice. Two 6-sided dice, with faces numbered 1 through 6, are rolled. How many possible outcomes of the roll are there?

Solution: Note that we are not concerned with the total value of the two die ("die" is the singular form of "dice"), but rather the set of all explicit outcomes of the rolls. Since the first die can come up with 6 possible values and the second die similarly can have 6 possible values (regardless of what appeared on the first die), the total number of potential outcomes is 36 (= 6 × 6). These possible outcomes are explicitly listed below as a series of pairs, denoting the values rolled on the pair of dice:

(1, 1) (1, 2) (1, 3) (1, 4) (1, 5) (1, 6)
(2, 1) (2, 2) (2, 3) (2, 4) (2, 5) (2, 6)
(3, 1) (3, 2) (3, 3) (3, 4) (3, 5) (3, 6)
(4, 1) (4, 2) (4, 3) (4, 4) (4, 5) (4, 6)
(5, 1) (5, 2) (5, 3) (5, 4) (5, 5) (5, 6)
(6, 1) (6, 2) (6, 3) (6, 4) (6, 5) (6, 6)

Counting with **or**

If you want to consider the total number of unique outcomes, when outcomes can come from source A **or** source B, then the equation you use depends on whether or not there are outcomes which are both in A and B. If not, you can use the simpler "Mutually Exclusive Counting" rule. Otherwise you need to use the slightly more involved Inclusion Exclusion rule.

Definition: Mutually Exclusive Counting

If the outcome of an experiment can either be drawn from set A or set B, where none of the outcomes in set A are the same as any of the outcomes in set B (called mutual exclusion), then there are $|A \text{ or } B| = |A| + |B|$ possible outcomes of the experiment.

Example: Sum of Routes. A route finding algorithm needs to find routes from Nairobi to Dar Es Salaam. It finds routes that either pass through Mt Kilimanjaro or Mombasa. There are 20 routes that pass through Mt Kilimanjaro, 15 routes that pass through Mombasa and 0 routes which pass through both Mt Kilimanjaro and Mombasa. How many routes are there total?

Solution: Routes can come from either Mt Kilimanjaro **or** Mombasa. The two sets of routes are mutually exclusive as there are zero routes which are in both groups. As such the total number of routes is addition: 20 + 15 = 35.

If you can show that two groups are mutually exclusive counting becomes simple addition. Of course not all sets are mutually exclusive. In the example above, imagine there had been a single route which went through both Mt Kilimanjaro and Mombasa. We would have double counted that route because it would be included in both the sets. If sets are not mutually exclusive, counting the **or** is still addition, we simply need to take into account any double counting.

Definition: Inclusion Exclusion Counting

If the outcome of an experiment can either be drawn from set A or set B, and sets A and B may potentially overlap (i.e., it is not the case that A and B are mutually exclusive), then the number of outcomes of the experiment is $|A \text{ or } B| = |A| + |B| - |A \text{ and } B|$.

Note that the Inclusion-Exclusion Principle generalizes the Sum Rule of Counting for arbitrary sets A and B. In the case where A and $B = \emptyset$, the Inclusion-Exclusion Principle gives the same result as the Sum Rule of Counting since $|A \text{ and } B| = 0$.

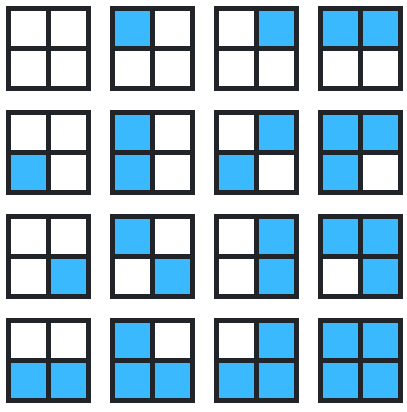
Example: An 8-bit string (one byte) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

Solution: The potential bit strings that match the receiver's criteria can either be the 64 strings that start with "01" (since that last 6 bits are left unspecified, allowing for $2^6 = 64$ possibilities) or the 64 strings that end with "10" (since the first 6 bits are unspecified). Of course, these two sets overlap, since strings that start with "01" and end with "10" are in both sets. There are $2^4 = 16$ such strings (since the middle 4 bits can be arbitrary). Casting this description into corresponding set notation, we have: $|A| = 64$, $|B| = 64$, and $|A \text{ and } B| = 16$, so by the Inclusion-Exclusion Principle, there are $64 + 64 - 16 = 112$ strings that match the specified receiver's criteria.

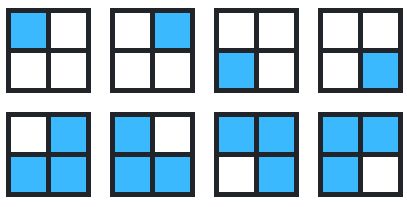
Overcounting and Correcting

One strategy for counting is sometimes to overcount a solution and then correct for any duplicates. This is especially common when it is easier to generate all outcomes under some relaxed assumptions, or someone introduces contraints. If you can argue that you have over-counted each element the same multiple number of times, you can simply correct by using division. If you can count exactly how many elements were over-counted you can correct using subtraction.

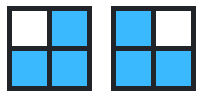
As a simple example to demonstrate the point, lets revisit the problem of generating all images, but this time lets just have 4 pixels (2x2) and each pixel can only be **blue** or white. How many unique images are there? Generating any image is a four step process where you choose each pixel one at a time. Since each pixel has two choices there are $2^4 = 16$ unique images (they are not exactly Picasso — but hey, it's 4 pixels):



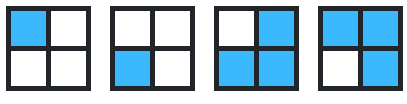
Now lets say we add in new "constraint" that we only want to accept pictures which have an odd number of pixels turned blue. There are two ways of getting to the answer. You could start out with the original 16 and work out that you need to subtract off 8 images that have either 0, 2 or 4 blue pixels (which is easier to work out after the next chapter). Or you could have counted up using Mutually Exclusive Counting: there are 4 ways of making an image with 1 pixel and 4 ways of making an image with 3. Both approaches lead to the same answer, 8.



Next lets add a much harder constraint: mirror indistinction. If you can flip any image horizontally to create another, they are no longer considered unique. For example these two both show up in our set of 8 odd-blue pixel images, but they are now considered to be the same (they are indistinct after a horizontal flip):



How many images have an odd number of pixels taking into account mirror indistinction? The answer is that for each unique image with odd numbers of blue pixels, under this new constraint, you have counted it twice: itself and its horizonal flip. To convince yourself that each image has been counted *exactly* twice you can look at all of the examples in the set of 8 images with an odd number of blue pixels. Each image is next to one which is indistinct after a horizontal flip. Since each image was counted exactly twice in the set of 8, we can divide by two to get the updated count. If we list them out we can confirm that there are 8/2=4 images left after this last constraint:



Applying any math (counting included) to novel contexts can be as much an art as it is a science. In the next chapter we will build a useful toolset from the basic first principles of counting by steps, and counting by "or".

Combinatorics

Counting problems can be approached from the basic building blocks described in the first section: Counting. However some counting problems are so ubiquitous in the world of probability that it is worth knowing a few higher level counting abstractions. When solving problems, if you can find the analogy from these canonical examples you can build off of the corresponding combinatorics formulas:

1. Permutations of Distinct Objects
2. Permutations with Indistinct Objects
3. Combinations with Distinct Objects
4. Bucketing with Distinct Objects
5. Bucketing with Indistinct Objects
6. Bucketing into Fixed Sized Containers

While these are by no means the only common counting paradigms, it is a helpful set.

Permutations of Distinct Objects

Definition: Permutation Rule

A permutation is an ordered arrangement of n distinct object. Those n objects can be permuted in $n \cdot (n–1) \cdot (n–2) \cdots 2 \cdot 1 = n!$ ways.

This changes slightly if you are permuting a subset of distinct objects, or if some of your objects are indistinct. We will handle those cases shortly! Note that unique is a synonym for distinct.

Example: How many unique orderings of characters are possible for the string "BAYES"? **Solution**: Since the order of characters is important, we are considering all permutations of the 5 distinct characters B, A, Y, E, and S: $5! = 120$. Here is the full list:

BAYES, BAYSE, BAEYS, BAESY, BASYE, BASEY, BYAES, BYASE, BYEAS, BYESA, BYSAE, BYSEA, BEAYS, BEASY, BEYAS, BEYSA, BESAY, BESYA, BSAYE, BSAEY, BSYAE, BSYEA, BSEAY, BSEYA, ABYES, ABYSE, ABEYS, ABESY, ABSYE, ABSEY, AYBES, AYBSE, AYEBS, AYESB, AYSBE, AYSEB, AEBYS, AEBSY, AEYBS, AEYSB, AESBY, AESYB, ASBYE, ASBEY, ASYBE, ASYEB, ASEBY, ASEYB, YBAES, YBASE, YBEAS, YBESA, YBSAE, YBSEA, YABES, YABSE, YAEBS, YAESB, YASBE, YASEB, YEBAS, YEBSA, YEABS, YEASB, YESBA, YESAB, YSBAE, YSBEA, YSABE, YSAEB, YSEBA, YSEAB, EBAYS, EBASY, EBYAS, EBYSA, EBSAY, EBSYA, EABYS, EABSY, EAYBS, EAYSB, EASBY, EASYB, EYBAS, EYBSA, EYABS, EYASB, EYSBA, EYSAB, ESBAY, ESBYA, ESABY, ESAYB, ESYBA, ESYAB, SBAYE, SBAEY, SBYAE, SBYEA, SBEAY, SBEYA, SABYE, SABEY, SAYBE, SAYEB, SAEBY, SAEYB, SYBAE, SYBEA, SYABE, SYAEB, SYEBA, SYEAB, SEBAY, SEBYA, SEABY, SEAYB, SEYBA, SEYAB

Example: a smart-phone has a 4-digit passcode. Suppose there are 4 smudges over 4 digits on the screen. How many distinct passcodes are possible?
Solution: Since the order of digits in the code is important, we should use permutations. And since there are exactly four smudges we know that each number in the passcode is distinct. Thus, we can plug in the permutation formula: 4! = 24.

Permutations of Indistinct Objects

Definition: Permutations of In-Distinct Objects

Generally when there are n objects and:

n_1 are the same (indistinguishable) and
n_2 are the same and
...
n_r are the same, then the number of distinct permutations is:

$$\text{Number of unique orderings} = \frac{n!}{n_1!n_2!\cdots n_r!}$$

Example: How many distinct bit strings can be formed from three 0's and two 1's?
Solution: 5 total digits would give 5! permutations. But that is assuming the 0's and 1's are distinguishable (to make that explicit, let's give each one a subscript). Here are the $3! \cdot 2! = 12$ different ways that we could have arrived at the identical string "01100" if we thought of each 0 and 1 as unique.

$$\begin{array}{ccccc} 0_1 & 1_0 & 1_1 & 0_2 & 0_3 \\ 0_1 & 1_0 & 1_1 & 0_3 & 0_2 \\ 0_2 & 1_0 & 1_1 & 0_1 & 0_3 \\ 0_2 & 1_0 & 1_1 & 0_3 & 0_1 \\ 0_3 & 1_0 & 1_1 & 0_1 & 0_2 \\ 0_3 & 1_0 & 1_1 & 0_2 & 0_1 \end{array}$$

$$\begin{array}{ccccc} 0_1 & 1_1 & 1_0 & 0_2 & 0_3 \\ 0_1 & 1_1 & 1_0 & 0_3 & 0_2 \\ 0_2 & 1_1 & 1_0 & 0_1 & 0_3 \\ 0_2 & 1_1 & 1_0 & 0_3 & 0_1 \\ 0_3 & 1_1 & 1_0 & 0_1 & 0_2 \\ 0_3 & 1_1 & 1_0 & 0_2 & 0_1 \end{array}$$

Since identical digits are indistinguishable, all the listed permutations are the same. For any given permutation, there are 3! ways of rearranging the 0's and 2! ways of rearranging the 1's (resulting in indistinguishable strings). We have over-counted. Using the formula for permutations of indistinct objects, we can correct for the over-counting:

$$\text{Total} = \frac{5!}{3! \cdot 2!} = \frac{120}{6 \cdot 2} = 10$$

Example: How many *distinct* orderings of characters are possible for the string "MISSISSIPPI"?

Solution: In the case of the string "MISSISSIPPI", we should separate the characters into four distinct groups of indistinct characters: one "M", four "I"s, four "S"s, and two "P"s. The number of distinct orderings are:

$$\frac{11!}{1!4!4!2!} = 34,650$$

Example: Consider the 4-digit passcode smart-phone from before. How many distinct passcodes are possible if there are 3 smudges over 3 digits on the screen?

Solution: One of 3 digits is repeated, but we don't know which one. We can solve this by making three cases, one for each digit that could be repeated (each with the same number of permutations). Let A, B, C represent the 3 digits, with C repeated twice. We can initially pretend the two C's are distinct $[A, B, C_1, C_2]$. Then each case will have 4! permutations: However, then we need to eliminate the double-counting of the permutations of the identical digits (one A, one B, and two C's):

$$\frac{4!}{2! \cdot 1! \cdot 1!}$$

Adding up the three cases for the different repeated digits gives

$$3 \cdot \frac{4!}{2! \cdot 1! \cdot 1!} = 3 \cdot 12 = 36$$

Part B: What if there are 2 smudges over 2 digits on the screen?

Solution: There are two possibilities: 2 digits used twice each, or 1 digit used 3 times, and other digit used once.

$$\frac{4!}{2! \cdot 2!} + 2 \cdot \frac{4!}{3! \cdot 1!} = 6 + (2 \cdot 4) = 6 + 8 = 14$$

You can use the power of computers to enumerate all permutations. Here is sample python code which uses the built in itertools library:

```
>>> import itertools

# get all 4! = 24 permutations of 1,2,3,4 as a list:
>>> list(itertools.permutations([1,2,3,4]))
[(1, 2, 3, 4), (1, 2, 4, 3), (1, 3, 2, 4), (1, 3, 4, 2), (1, 4, 2, 3), (1, 4, 3, 2), (2, 1, 3,
4), (2, 1, 4, 3), (2, 3, 1, 4), (2, 3, 4, 1), (2, 4, 1, 3), (2, 4, 3, 1), (3, 1, 2, 4), (3, 1,
4, 2), (3, 2, 1, 4), (3, 2, 4, 1), (3, 4, 1, 2), (3, 4, 2, 1), (4, 1, 2, 3), (4, 1, 3, 2), (4,
2, 1, 3), (4, 2, 3, 1), (4, 3, 1, 2), (4, 3, 2, 1)]

# get all 3!/2! = 3 unique permutations of 1,1,2 as a set:
>>> set(itertools.permutations([1,1,2]))
{(1, 2, 1), (2, 1, 1), (1, 1, 2)}
```

Combinations of Distinct Objects

Definition: Combinations

A combination is an unordered selection of r objects from a set of n objects. If all objects are distinct, and objects are not "replaced" once selected, then the number of ways of making the selection is:

$$\text{Number of unique selections} = \frac{n!}{r!(n-r)!} = \binom{n}{r}$$

Here are all the $10 = \binom{5}{3}$ ways of choosing three items from a list of 5 unique numbers:

```
# Get all ways of chosing three numbers from [1,2,3,4,5]
>>> list(itertools.combinations([1,2,3,4,5], 3))
[(1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5), (1, 4, 5), (2, 3, 4), (2, 3, 5), (2,
4, 5), (3, 4, 5)]
```

Notice how order doesn't matter. Since (1, 2, 3) is in the set of combinations, we don't also include (3, 2, 1) as this is considered to be the same selection. Note that this formula does not work if some of the objects are indistinct from one another.

How did we get the formula $\frac{n!}{r!(n-r)!}$? Consider this general way to select r unordered objects from a set of n objects, e.g., "7 choose 3":

1. First consider permutations of all n objects. There are $n!$ ways to do that.
2. Then select the first r in the permutation. There is one way to do that.
3. Note that the order of r selected objects is irrelevant. There are $r!$ ways to permute them. The selection remains unchanged.
4. Note that the order of $(n-r)$ unselected objects is irrelevant. There are $(n-r)!$ ways to permute them. The selection remains unchanged.

$$\text{Total} = \frac{n!}{r! \cdot (n-r)!} = \binom{n}{r}$$

Example: In the Hunger Games, how many ways are there of choosing 2 villagers from district 12, which has a population of 8,000?

Solution: This is a straightforward combinations problem. $\binom{8000}{2}$ =31,996,000.

Part A: How many ways are there to select 3 books from a set of 6?
Solution: If each of the books are distinct, then this is another straightforward combination problem. There are $\binom{6}{3} = \frac{6!}{3!3!} = 20$ ways.

Part B: How many ways are there to select 3 books if there are two books that should not both be chosen together? For example, if you are choosing 3 out of 6 probability books, don't choose both the 8th and 9th edition of the Ross textbook.
Solution: This problem is easier to solve if we split it up into cases. Consider the following three different cases:
Case 1: Select the 8th Ed. and 2 other non-9th Ed. books: There are $\binom{4}{2}$ ways of doing so.
Case 2: Select the 9th Ed. and 2 other non-8th Ed. books: There are $\binom{4}{2}$ ways of doing so.
Case 3: Select 3 books from the 4 remaining books that are neither the 8th nor the 9th edition: There are $\binom{4}{3}$ ways of doing so.
Using our old friend the Sum Rule of Counting, we can add the cases:

$$\text{Total} = 2 \cdot \binom{4}{2} + \binom{4}{3} = 16$$

Alternatively, we could have calculated all the ways of selecting 3 books from 6, and then subtract the "forbidden" ones (i.e., the selections that break the constraint).
Forbidden Case: Select 8th edition and 9th edition and 1 other book. There are $\binom{4}{1}$ ways of doing so (which equals 4). Total = All possibilities - forbidden = 20 - 4 = 16 Two different ways to get the same right answer!

Bucketing with Distinct Objects

In this section we are going to be counting the many different ways that we can think of stuffing elements into containers. (It turns out that Jacob Bernoulli was into voting and ancient Rome. And in ancient Rome they used urns for ballot boxes. For this reason many books introduce this through counting ways to put balls in urns.) This "bucketing" or "group assignment" process is a useful metaphor for many counting problems.

The most common case that we will want to consider is when all of the items you are putting into buckets are distinct. In that case you can think of bucketing as a series of steps, and employ the step rule of counting. The first step? You put the first distinct item into a bucket (there are number-of-buckets ways to do this). Second step? You put the second distinct item into a bucket (again, there are number-of-buckets ways to do this).

Bucketing Distinct Items:

Suppose you want to place n distinguishable items into r containers. The number of ways of doing so is:

$$r^n$$

You have n steps (place each item) and for each item you have r choices

Problem: Say you want to put 10 distinguishable balls into 5 urns (No! Wait! Don't say that! Not urns!). Okay, fine. No urns. Say we are going to put 10 different strings into 5 buckets of a hash table. How many possible ways are there of doing this?

Solution: You can think of this as 10 independent experiments each with 5 outcomes. Using our rule for bucketing with distinct items, this comes out to 5^{10}.

Bucketing with Indistinct Objects

While the previous example allowed us to put n distinguishable objects into r distinct groups, the more interesting problem is to work with n indistinguishable objects.

Divider Method:

Suppose you want to place n indistinguishable items into r containers. The divider method works by imagining that you are going to solve this problem by sorting two types of objects, your n original elements and $(r-1)$ dividers. Thus, you are permuting $n+r-1$ objects, n of which are the same (your elements) and $r-1$ of which are the same (the dividers). Thus the total number of outcomes is:

$$\frac{(n+r-1)!}{n!(r-1)!} = \binom{n+r-1}{n} = \binom{n+r-1}{r-1}$$

The divider method can be derived via the "Stars and Bars" method. This is a creative construction where we consider permutations of indistinguishable items, represented by stars `*`, and dividers between our containers, represented by bars `|`. Any distinct permutation of these stars and bars represents a unique assignments of our items to containers.

Imagine we want to separate 5 indistinguishable objects into 3 containers. We can think of the problem as finding the number of ways to order 5 stars and 2 bars `*****||`. Any permutation of these symbols represents a unique assignment. Here are a few examples:

`**|*|**` represents 2 items in the first bucket, 1 item in the second and 2 items in the third.

`****||*` represents 4 items in the first bucket, 0 item in the second and 1 items in the third.

`||*****` represents 0 items in the first bucket, 0 item in the second and 5 items in the third.

Why are there only 2 dividers when there are 3 buckets? This is an example of a fence-post-problem". With 2 dividers you have created three containers. We already have a method for counting permutations with some indistinct items. For the example above where we have seven elements in our permutation ($n = 5$ stars and $r - 1 = 2$ bars):

$$\text{Number of unique orderings} = \frac{n!}{n_1!n_2!} = \frac{(n+r-1)!}{n!(r-1)!} = \frac{7!}{5!2!} = 21$$

Part A: Say you are a startup incubator and you have \$10 million to invest in 4 companies (in \$1 million increments). How many ways can you allocate this money?
Solution: This is just like putting 10 balls into 4 urns. Using the Divider Method we get:

$$\text{Total ways} = \binom{10+4-1}{10} = \binom{13}{10} = 286$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where x_i represents the investment in company i such that $x_i \geq 0$ for all $i = 1, 2, 3, 4$.

Part B: What if you know you want to invest at least \$3 million in Company 1?
Solution: There is one way to give \$3 million to Company 1. The number of ways of investing the remaining money is the same as putting 7 balls into 4 urns.

$$\text{Total Ways} = \binom{7+4-1}{7} = \binom{10}{7} = 120$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 = 10$, where $x_1 \geq 3$ and $x_2, x_3, x_4 \geq 0$. To translate this problem into the integer solution equation that we can solve via the divider method, we need to adjust the bounds on x_1 such that the problem becomes
$x_1 + x_2 + x_3 + x_4 = 7$, where x_i is defined as in Part A.

Part C: What if you don't have to invest all \$10 M? (The economy is tight, say, and you might want to save your money.)
Solution: Imagine that you have an extra company: yourself. Now you are investing \$10 million in 5 companies. Thus, the answer is the same as putting 10 balls into 5 urns.

$$\text{Total} = \binom{10+5-1}{10} = \binom{14}{10} = 1001$$

This problem is analogous to solving the integer equation $x_1 + x_2 + x_3 + x_4 + x_5 = 10$, such that $x_i \geq 0$ for all $i = 1, 2, 3, 4, 5$.

Bucketing into Fixed Sized Containers

Bucketing into Fixed Sized Containers:

If n objects are distinct, then the number of ways of putting them into r groups of objects, such that group i has size n_i, and $\sum_{i=1}^{r} n_i = n$, is:

$$\frac{n!}{n_1! n_2! \cdots n_r!} = \binom{n}{n_1, n_2, \ldots, n_r}$$

where $\binom{n}{n_1, n_2, \ldots, n_r}$ is special notation called the multinomial coefficient.

You may have noticed that this is the exact same formula as "Permutations With Indistinct Objects". There is a deep parallel. One way to imagine assigning objects into their groups would be to imagine the groups themselves as objects. You have one object per "slot" in a group. So if there were two slots in group 1, three slots in group 2, and one slot in group 3 you could have six objects (1, 1, 2, 2, 2, 3). Each unique permutation can be used to make a unique assignment.

Problem:

Company Camazon has 13 distinct new servers that they would like to assign to 3 datacenters, where Datacenter A, B, and C have 6, 4, and 3 empty server racks, respectively. How many different divisions of the servers are possible?

Solution: This is a straightforward application of our multinomial coefficient representation. Setting $n_1 = 6, n_2 = 4, n_3 = 3$, $\binom{13}{6,4,3} = 60,060$.

Another way to do this problem would be from first principles of combinations as a multipart experiment. We first select the 6 servers to be assigned to Datacenter A, in $\binom{13}{6}$ ways. Now out of the 7 servers remaining, we select the 4 servers to be assigned to Datacenter B, in $\binom{7}{4}$ ways. Finally, we select the 3 servers out of the remaining 3 servers, in $\binom{3}{3}$ ways. By the Product Rule of Counting, the total number of ways to assign all servers would be $\binom{13}{6}\binom{7}{4}\binom{3}{3} = \frac{13!}{6!4!3!} = 60,060$.

Definition of Probability

What does it mean when someone makes a claim like "the probability that you find a pearl in an oyster is 1 in 5,000?" or "the probability that it will rain tomorrow is 52%"?

Events and Experiments

When we speak about probabilities, there is always an implied context, which we formally call the "experiment". For example: flipping two coins is something that probability folks would call an experiment. In order to precisely speak about probability, we must first define two sets: the set of all possible outcomes of an experiment, and the subset that we consider to be our event (what is a set?).

Definition: Sample Space, S

A Sample Space is the set of all possible outcomes of an experiment. For example:

- Coin flip: S = {Heads, Tails}
- Flipping two coins: S = {(H, H), (H, T), (T, H), (T, T)}
- Roll of 6-sided die: S = {1, 2, 3, 4, 5, 6}
- The number of emails you receive in a day: $S = \{x | x \in \mathbb{Z}, x \geq 0\}$ (non-neg. ints)
- YouTube hours in a day: $S = \{x | x \in \mathbb{R}, 0 \leq x \leq 24\}$

Definition: Event, E

An Event is some subset of S that we ascribe meaning to. In set notation ($E \subseteq S$).For example:

- Coin flip is heads: E = {Heads}
- At least 1 head on 2 coin flips = {(H, H), (H, T), (T, H)}
- Roll of die is 3 or less: E = {1, 2, 3}
- You receive less than 20 emails in a day: $E = \{x | x \in Z, 0 \leq x < 20\}$ (non-neg. ints)
- Wasted day (≥ 5 YouTube hours): $E = \{x | x \in R, 5 \leq x \leq 24\}$

Events can be represented as capital letters such as E or F.

[todo] In the world of probability, events are binary: they either happen or they don't.

Definition of Probability

It wasn't until the 20th century that humans figured out a way to precisely define what the word probability means:

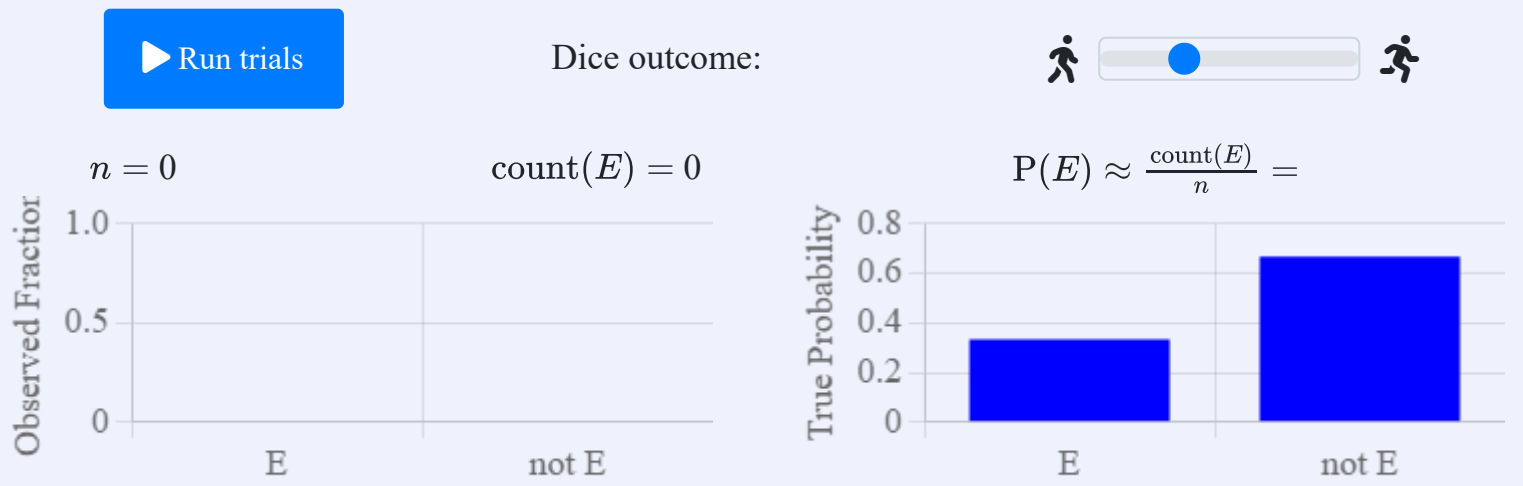
$$\mathrm{P}(\text{Event}) = \lim_{n \to \infty} \frac{\text{count}(\text{Event})}{n}$$

In English this reads: lets say you perform n trials of an "experiment" which could result in a particular "Event" occurring. The probability of the event occurring, $\mathrm{P}(\text{Event})$, is the ratio of trials that result in the event, written as $\text{count}(\text{Event})$, to the number of trials performed, n. In the limit, as your number of trials approaches infinity, the ratio will converge to the true probability. People also apply other semantics to the concept of a probability. One common meaning ascribed is that $\mathrm{P}(E)$ is a measure of the chance of event E occurring.

Example: Probability in the limit

Here we use the definition of probability to calculate the probability of event E, rolling a "5" or a "6" on a fair six-sided dice. Hit the "Run trials" button to start running trials of the experiment "roll dice". Notice how $\mathrm{P}(E)$, converges to $2/6$ or 0.33 repeating.

Event E: Rolling a 5 or 6 on a six-sided dice.



Measure of uncertainty: It is tempting to think of probability as representing some natural randomness in the world. That might be the case. But perhaps the world isn't random. I propose a deeper way of thinking about probability. There is so much that we as humans don't know, and probability is our robust language for expressing our belief that an event will happen given our limited knowledge. This interpretation acknowledges your own uncertainty of an event. Perhaps if you knew the position of every water molecule, you could perfectly predict tomorrow's weather. But we don't have such knowledge and as such we use probability to talk about the chance of rain tomorrow given the information that we have access to.

Origins of probabilities: The different interpretations of probability are reflected in the many origins of probabilities that you will encounter in the wild (and not so wild) world. Some probabilities are calculated analytically using mathematical proofs. Some probabilities are calculated from data, experiments or simulations. Some probabilities are just made up to represent a belief. Most probabilities are generated from a combination of the above. For example, someone will make up a prior belief, that belief will be mathematically updated using data and evidence. Here is an example of calculating a probability from data:

Probabilities and simulations: Another way to compute probabilities is via simulation. For some complex problems where the probabilities are too hard to compute analytically you can run simulations using your computer. If your simulations generate believable trials from the sample space, then the probability of an event E is approximately equal to the fraction of simulations that produced an outcome from E. Again, by the definition of probability, as your number of simulations approaches infinity, the estimate becomes more accurate.

Probabilities and percentages: You might hear people refer to a probability as a percent. That the probability of rain tomorrow is 32%. The proper way to state this would be to say that 0.32 is the probability of rain. Percentages are simply probabilities multiplied by 100. "Percent" is latin for "out of one hundred".

Problem: Use the definition of probability to approximate the answer to the question: "What is the probability a new-born elephant child is male?" Contrary to what you might think the gender outcomes of a newborn elephant are not equally likely between male and female. You have data from a report in Animal Reproductive Science which states that 3,070 elephants were born in Myanmar of which 2,180 were male [1]. Humans also don't have a 50/50 sex ratio at birth [2].

Answer: The Experiment is: A single elephant birth in Myanmar.
The sample space is the set of possible sexes assigned at birth, {Male, Female, Intersex}.
E is the event that a new-born elephan child is male, which in set notation is the subset {Male} of the sample space. The outcomes are not equally likely.

By the definition of probability, the ratio — of trials that result in the event, to the total number of trials — will tend to our desired probability:

$$
\begin{aligned}
\mathrm{P}(\text{Born Male}) &= \mathrm{P}(E) \\
&= \lim_{n \to \infty} \frac{\text{count}(E)}{n} \\
&\approx \frac{2,180}{3,070} \\
&\approx 0.710
\end{aligned}
$$

Since 3,000 is quite a bit less than infinity, this is an approximation. It turns out, however, to be a rather good one. A few important notes: there is no guarantee that our estimate applies to elephants outside Myanmar. Later in the class we will develop language for "how confident we can be in a number like 0.71 after 3,000 trials?" Using tools from later in class we can say that we have 98% confidence that the true probability is within 0.02 of 0.710.

Axioms of Probability

Here are some basic truths about probabilities that we accept as axioms:

Axiom 1: $0 \leq \mathrm{P}(E) \leq 1$	All probabilities are numbers between 0 and 1.
Axiom 2: $\mathrm{P}(S) = 1$	All outcomes must be from the Sample Space.
Axiom 3: If E and F are mutually exclusive, then $\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F)$	The probability of "or" for mutually exclusive events

These three axioms are formally called the Kolmogorov axioms and they are considered to be the foundation of probability theory. They are also useful identities!

You can convince yourself of the first axiom by thinking about the math definition of probability. As you perform trials of an experiment it is not possible to get more events than trials (thus probabilities are less than 1) and its not possible to get less than 0 occurrences of the event (thus probabilities are greater than 0). The second axiom makes sense too. If your event is the sample space, then each trial must produce the event. This is sort of like saying; the probability of you eating cake (event) if you eat cake (sample space that is the same as the event) is 1. The third axiom is more complex and in this textbook we dedicate an entire chapter to understanding it: Probability of or. It applies to events that have a special property called "mutual exclusion": the events do not share any outcomes.

These axioms have great historical significance. In the early 1900s it was not clear if probability was somehow different than other fields of math -- perhaps the set of techniques and systems of proofs from other fields of mathematics couldn't apply. Kolmogorov's great success was to show to the world that the tools of mathematics did in fact apply to probability. From the foundation provided by this set of axioms mathematicians built the edifice of probability theory.

Provable Identities

We often refer to these as corollaries that are directly provable from the three axioms given above.

Identity 1: $\mathrm{P}(E^{\mathrm{C}}) = 1 - \mathrm{P}(E)$	The probability of event E not happening
Identity 2: If $E \subseteq F$, then $\mathrm{P}(E) \leq \mathrm{P}(F)$	Events which are subsets

This first identity is especially useful. For any event, you can calculate the probability of the event *not* occurring which we write in probability notation as E^{C}, if you know the probability of it occurring -- and vice versa. We can also use this identity to show you what it looks like to prove a theorem in probability.

Proof: $\mathrm{P}(E^{\mathrm{C}}) = 1 - \mathrm{P}(E)$

$\mathrm{P}(S) = \mathrm{P}(E \text{ or } E^{\mathrm{C}})$	E or E^{C} covers every outcome in the sample space
$\mathrm{P}(S) = \mathrm{P}(E) + \mathrm{P}(E^{\mathrm{C}})$	Events E and E^{C} are mututally exclusive
$1 = \mathrm{P}(E) + \mathrm{P}(E^{\mathrm{C}})$	Axiom 2 of probability
$\mathrm{P}(E^{\mathrm{C}}) = 1 - \mathrm{P}(E)$	By re-arranging

Equally Likely Outcomes

Some sample spaces have equally likely outcomes. We like those sample spaces, because there is a way to calculate probability questions about those sample spaces simply by counting. Here are a few examples where there are equally likely outcomes:

- Coin flip: S = {Head, Tails}
- Flipping two coins: S = {(H, H), (H, T), (T, H), (T, T)}
- Roll of 6-sided die: S = {1, 2, 3, 4, 5, 6}

Because every outcome is equally likely, and the probability of the sample space must be 1, we can prove that each outcome must have probability:

$$\mathrm{P}(\text{an outcome}) = \frac{1}{|S|}$$

Where |S| is the size of the sample space, or, put in other words, the total number of outcomes of the experiment. Of course this is only true in the special case where every outcome has the same likelihood.

Definition: Probability of Equally Likely Outcomes

If S is a sample space with equally likely outcomes, for an event E that is a subset of the outcomes in S:

$$\mathrm{P}(E) = \frac{\text{number of outcomes in } E}{\text{number of outcomes in } S} = \frac{|E|}{|S|}$$

There is some art form to setting up a problem to calculate a probability based on the equally likely outcome rule. (1) The first step is to explicitly define your sample space and to argue that all outcomes in your sample space are equally likely. (2) Next, you need to count the number of elements in the sample space and (3) finally you need to count the size of the event space. The event space must be all elements of the sample space that you defined in part (1). The first step leaves you with a lot of choice! For example you can decide to make indistinguishable objects distinct, as long as your calculation of the size of the event space makes the exact same assumptions.

Example: What is the probability that the sum of two die is equal to 7?

Buggy Solution: You could define your sample space to be all the possible sum values of two die (2 through 12). However this sample space fails the “equally likely” test. You are not equally likely to have a sum of 2 as you are to have a sum of 7.

Solution: Consider the sample space from the previous chapter where we thought of the die as distinct and enumerated all of the outcomes in the sample space. The first number is the roll on die 1 and the second number is the roll on die 2. Note that (1, 2) is distinct from (2, 1). Since each outcome is equally likely, and the sample space has exactly 36 outcomes, the likelihood of any one outcome is $\frac{1}{36}$. Here is a visualization of all outcomes:

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	**(1,6)**
(2,1)	(2,2)	(2,3)	(2,4)	**(2,5)**	(2,6)
(3,1)	(3,2)	(3,3)	**(3,4)**	(3,5)	(3,6)
(4,1)	(4,2)	**(4,3)**	(4,4)	(4,5)	(4,6)
(5,1)	**(5,2)**	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

The event (sum of dice is 7) is the subset of the sample space where the sum of the two dice is 7. Each outcome in the event is highlighted in **blue**. There are 6 such outcomes: (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1). Notice that (1, 6) is a different outcome than (6, 1). To make the outcomes equally likely we had to make the die distinct.

$$\begin{aligned} \mathrm{P}(\text{Sum of two dice is } 7) &= \frac{|E|}{|S|} && \text{Since outcomes are equally likely} \\ &= \frac{6}{36} = \frac{1}{6} && \text{There are 6 outcomes in the event} \end{aligned}$$

Interestingly, this idea also applies to continuous sample spaces. Consider the sample space of all the outcomes of the computer function "random" which produces a real valued number between 0 and 1, where all real valued numbers are equally likely. Now consider the event E that the number generated is in the range [0.3 to 0.7]. Since the sample space is equally likely, $\mathrm{P}(E)$ is the ratio of the size of E to the size of S. In this case $\mathrm{P}(E) = \frac{0.4}{1} = 0.4$.

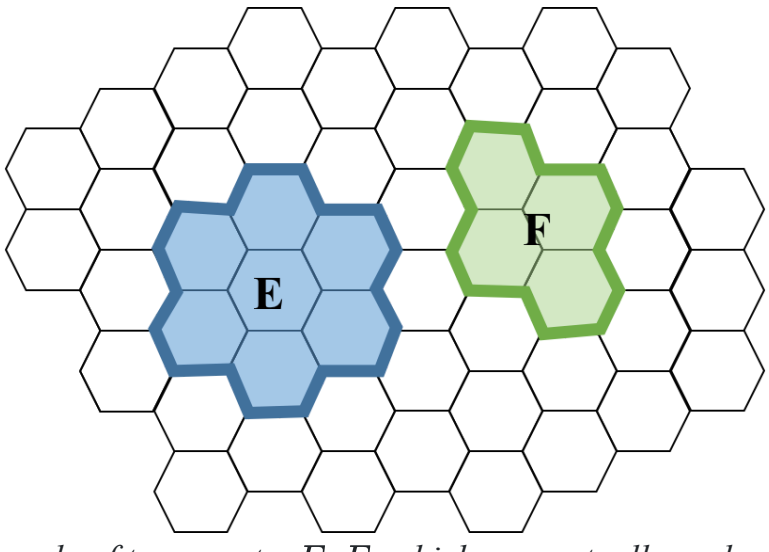
Probability of **or**

The equation for calculating the probability of either event E ***or*** event F happening, written $\mathrm{P}(E \text{ or } F)$ or equivalently as $\mathrm{P}(E \cup F)$, is deeply analogous to counting the size of two sets. As in counting, the equation that you can use depends on whether or not the events are "mutually exclusive". If events are mutually exclusive, it is very straightforward to calculate the probability of either event happening. Otherwise, you need the more complex "inclusion exclusion" formula.

Mutually exclusive events

Two events: E, F are considered to be mutually exclusive (in set notation $E \cap F = \emptyset$) if there are no outcomes that are in both events (recall that an event is a set of outcomes which is a subset of the sample space). In English, mutually exclusive means that two events can't both happen.

Mutual exclusion can be visualized. Consider the following visual sample space where each outcome is a hexagon. The set of all the fifty hexagons is the full sample space:



Example of two events: E, F, which are mutually exclusive.

Both events E and F are subsets of the same sample space. Visually, we can note that the two sets do not overlap. They are mutually exclusive: there is no outcome that is in both sets.

Or with Mutually Exclusive Events

Definition: Probability of **or** for mututally exclusive events
If two events: E, F are mutually exclusive then the probability of E ***or*** F occurring is:

$$\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F)$$

This property applies regardless of how you calculate the probability of E or F. Moreover, the idea extends to more than two events. Lets say you have n events $E_1, E_2, \ldots E_n$ where each event is mutually exclusive of one another (in other words, no outcome is in more than one event). Then:

$$\mathrm{P}(E_1 \text{ or } E_2 \text{ or } \ldots \text{ or } E_n) = \mathrm{P}(E_1) + \mathrm{P}(E_2) + \cdots + \mathrm{P}(E_n) = \sum_{i=1}^{n} \mathrm{P}(E_i)$$

You may have noticed that this is one of the axioms of probability. Though it might seem intuitive, it is one of three rules that we accept without proof.

Caution: Mutual exclusion only makes it easier to calculate the probability of E or F, not other ways of combining events, such as E and F.

At this point we know how to compute the probability of the "or" of events if and only if they have the mutual exclusion property. What if they don't?

Or with Non-Mutually Exclusive Events

Unfortunately, not all events are mutually exclusive. If you want to calculate $\mathrm{P}(E \text{ or } F)$ where the events E and F are ***not*** mutually exclusive you can ***not*** simply add the probabilities. As a simple sanity check, consider the event E: getting heads on a coin flip, where $\mathrm{P}(E) = 0.5$. Now imagine the sample space S, getting either a heads or a tails on a coin flip. These events are not mutually exclusive (the outcome heads is in both). If you incorrectly assumed they were mutually exclusive and tried to calculate $\mathrm{P}(E \text{ or } S)$ you would get this buggy derivation:

Buggy derivation: Incorrectly assuming mutual exclusion

Calculate the probability of E, getting an even number on a dice role (2, 4 or 6), or F, getting three or less (1, 2, 3) on the same dice role.

$$\begin{aligned} \mathrm{P}(E \text{ or } F) &= \mathrm{P}(E) + \mathrm{P}(F) && \text{Incorrectly assumes mutual exclusion} \\ &= 0.5 + 0.5 && \text{substitute the probabilities of } E \text{ and } S \\ &= 1.0 && \text{uh oh!} \end{aligned}$$

The probability can't be one since the outcome 5 is neither three or less nor even. The problem is that we double counted the probability of getting a 2, and the fix is to subtract out the probability of that doubly counted case.

What went wrong? If two events are not mutually exclusive, simply adding their probabilities double counts the probability of any outcome which is in both events. There is a formula for calculating ***or*** of two non-mutually exclusive events: it is called the "inclusion exclusion" principle.

Definition: Inclusion Exclusion principle
For any two events: E, F:

$$\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F) - \mathrm{P}(E \text{ and } F)$$

This formula does have a version for more than two events, but it gets rather complex. For three events, E, F, and G the formula is:

$$\begin{aligned} \mathrm{P}(E \text{ or } F \text{ or } G) = \; & \mathrm{P}(E) + \mathrm{P}(F) + \mathrm{P}(G) \\ & - \mathrm{P}(E \text{ and } F) - \mathrm{P}(E \text{ and } G) - P(F \text{ and } G) \\ & + \mathrm{P}(E \text{ and } F \text{ and } G) \end{aligned}$$

For n events, $E_1, E_2, \ldots E_n$: build a running sum. Add all the probabilities of the events on their own. Then subtract all pairs of events. Then add all subsets of 3 events. Then subtract all subset of 4 events. Continue this process, up until n, adding the subsets if the size of subsets is odd, else subtracting them. The alternating addition and subtraction is where the name inclusion exclusion comes from. This is a complex process and you should first check if there is an easier way to calculate your probability.

Note that the inclusion exclusion principle also applies for mutually exclusive events. If two events are mutually exclusive $\mathrm{P}(E \text{ and } F) = 0$ since its not possible for both E and F to occur. As such the formula $\mathrm{P}(E) + \mathrm{P}(F) - \mathrm{P}(E \text{ and } F)$ reduces to $\mathrm{P}(E) + \mathrm{P}(F)$.

Inclusion-Exclusion with Three Events

What does the inclusion exclusion property look like if we have three events, that are not mutually exclusive, and we want to know the probability of or, $\mathrm{P}(E_1 \text{ or } \mathrm{E}_2 \text{ or } E_3)$?

Recall that if they are mutually exclusive, we simply add the probabilities. If they are not mutually exclusive, you need to use the inclusion exclusion formula for three events:

$$
\begin{aligned}
\mathrm{P}(E_1 \text{or } \mathrm{E}_2 \text{ or } E_3) = \\
&+ \mathrm{P}(E_1) \\
&+ \mathrm{P}(E_2) \\
&+ \mathrm{P}(E_3) \\
&- \mathrm{P}(E_1 \text{ and } E_2) \\
&- \mathrm{P}(E_1 \text{ and } E_3) \\
&- \mathrm{P}(E_2 \text{ and } E_3) \\
&+ \mathrm{P}(E_1 \text{ and } E_2 \text{ and } E_3)
\end{aligned}
$$

In words, to get the probability of three events, you: (1) add the probability of the events on their own. (2) Then you need to subtract off the probability of every *pair* of events co-occuring. (3) Finally, you add in the probability of all three events co-occuring.

Inclusion-Exclusion with n Events

Before we explore the general formula, lets look at one more example. Inclusion-exclusion with four events:

$$
\begin{aligned}
\mathrm{P}(E_1 \text{ or } E_2 \text{ or } E_3 \text{ or } E_4) = \\
&+ \mathrm{P}(E_1) \\
&+ \mathrm{P}(E_2) \\
&+ \mathrm{P}(E_3) \\
&+ \mathrm{P}(E_4) \\
&- \mathrm{P}(E_1 \text{ and } E_2) \\
&- \mathrm{P}(E_1 \text{ and } E_3) \\
&- \mathrm{P}(E_1 \text{ and } E_4) \\
&- \mathrm{P}(E_2 \text{ and } E_3) \\
&- \mathrm{P}(E_2 \text{ and } E_4) \\
&- \mathrm{P}(E_3 \text{ and } E_4) \\
&+ \mathrm{P}(E_1 \text{ and } E_2 \text{ and } E_3) \\
&+ \mathrm{P}(E_1 \text{ and } E_2 \text{ and } E_4) \\
&+ \mathrm{P}(E_1 \text{ and } E_3 \text{ and } E_4) \\
&+ \mathrm{P}(E_2 \text{ and } E_3 \text{ and } E_4) \\
&- \mathrm{P}(E_1 \text{ and } E_2 \text{ and } E_3 \text{ and } E_4)
\end{aligned}
$$

Do you see the pattern? For every possible subset of events from our n events, we calculate the probability of the "and" of the subset. If the subset has an odd number of events, we add its probability mass, otherwise we subtract the probability. This can be written up mathematically — but it is a rather hard pattern to express in notation:

$$
\begin{aligned}
\mathrm{P}(E_1 \text{ or } E_2 \text{ or } \cdots \text{ or } E_n) &= \sum_{r-1}^{n} (-1)^{r+1} Y_r \\
\text{s.t. } Y_r &= \sum_{1 \leq i_1 < \cdots < i_r \leq n} \mathrm{P}(E_{i_1} \text{ and } \cdots \text{ and } E_{i_r})
\end{aligned}
$$

The notation for Y_r is especially hard to parse. Y_r sums over all ways of selecting a subset of r events. For each selection of r events, calculate the probability of the intersection of those events. $(-1)^{r+1}$ is saying: alternate between addition and subtraction, starting with addition.

It is not especially important to follow the math notation here. The main take away is that the general inclusion exclusion principle gets incredibly complex with multiple events. Often, the way to make progress in this situation is to find a way to solve your problem using another method.

The formulas for calculating the ***or*** of events that are not mutually exclusive often require calculating the probability of the ***and*** of events. Learn more about the probability of and in the next section.

Conditional Probability

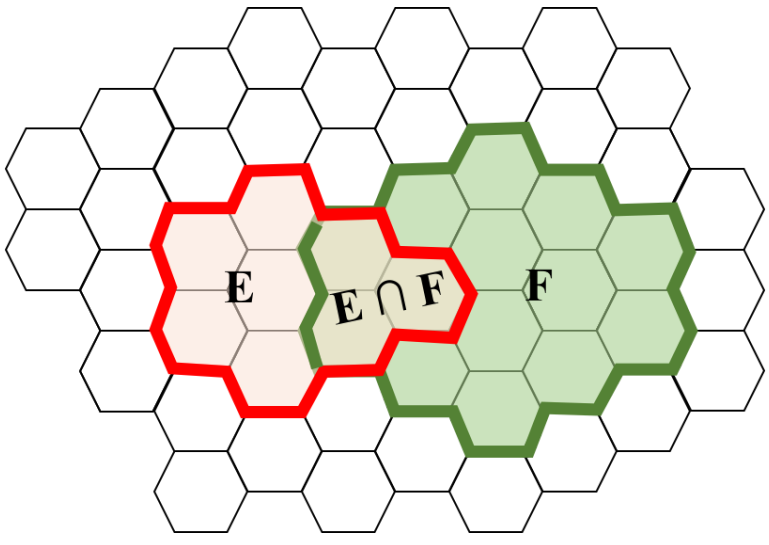
In English, a conditional probability states "what is the chance of an event E happening given that I have already observed some other event F". It is a critical idea in machine learning and probability because it allows us to update our probabilities in the face of new evidence.

When you condition on an event happening you are entering the universe where that event has taken place. Formally, once you condition on F the only outcomes that are now possible are the ones which are consistent with F. In other words your sample space will now be reduced to F. As an aside, in the universe where F has taken place, all rules of probability still hold!

Definition: Conditional Probability.
The probability of E given that (aka conditioned on) event F already happened:

$$\mathrm{P}(E|F) = \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)}$$

Let's use a visualization to get an intuition for why the conditional probability formula is true. Again consider events E and F which have outcomes that are subsets of a sample space with 50 equally likely outcomes, each one drawn as a hexagon:



Conditioning on F means that we have entered the world where F has happened (and F, which has 14 equally likely outcomes, has become our new sample space). Given that event F has occurred, the conditional probability that event E occurs is the subset of the outcomes of E that are consistent with F. In this case we can visually see that those are the three outcomes in E and F. Thus we have the:

$$\mathrm{P}(E|F) = \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)} = \frac{3/50}{14/50} = \frac{3}{14} \approx 0.21$$

Even though the visual example (with equally likely outcome spaces) is useful for gaining intuition, conditional probability applies regardless of whether the sample space has equally likely outcomes!

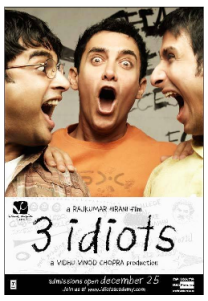
Conditional Probability Example

Let's use a real world example to better understand conditional probability: movie recommendation. Imagine a streaming service like Netflix wants to figure out the probability that a user will watch a movie E (for example, Life is Beautiful), based on knowing that they watched a different movie F (say Amélie). To start lets answer the simpler question, what is the probability that a user watches the movie Life is Beautiful, E? We can solve this problem using the definition of probability and a dataset of movie watching [1]:

$$\begin{aligned}\mathrm{P}(E) &= \lim_{n\to\infty} \frac{\text{count}(E)}{n} \approx \frac{\#\text{ people who watched movie } E}{\#\text{ people on Netflix}} \\ &= \frac{1,234,231}{50,923,123} \approx 0.02\end{aligned}$$

In fact we can do this for many movies E:

$\mathrm{P}(E) = 0.02$



$\mathrm{P}(E) = 0.01$

$\mathrm{P}(E) = 0.05$

$\mathrm{P}(E) = 0.09$

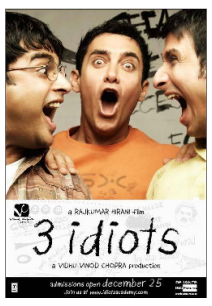
$\mathrm{P}(E) = 0.03$

Now for a more interesting question. What is the probability that a user will watch the movie Life is Beautiful (E), given they watched Amelie (F)? We can use the definition of conditional probability.

$$\begin{aligned}\mathrm{P}(E|F) &= \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)} && \text{Def of Cond Prob} \\ &\approx \frac{(\#\text{ who watched } E \text{ and } F)/(\#\text{ of people on Netflix})}{(\#\text{ who watched movie } F)/(\#\text{ people on Netflix})} && \text{Def of Prob} \\ &\approx \frac{\#\text{ of people who watched both } E \text{ and } F}{\#\text{ of people who watched movie } F} && \text{Simplifying}\end{aligned}$$

If we let F be the event that someone watches the movie Amélie, we can now calculate $\mathrm{P}(E|F)$, the *conditional* probability that someone watches movie E:

$\mathrm{P}(E|F) = 0.09$



$\mathrm{P}(E|F) = 0.03$

$\mathrm{P}(E|F) = 0.05$

$\mathrm{P}(E|F) = 0.02$

$\mathrm{P}(E|F) = \mathbf{1.00}$

Why do some probabilities go up, some probabilities go down, and some probabilities are unchanged after we observe that the person has watched Amelie (F)? If you know someone watched Amelie, they are more likely to watch Life is Beautiful, and less likely to watch Star Wars. We have new information on the person!

The Conditional Paradigm

When you condition on an event you enter the universe where that event has taken place. In that new universe all the laws of probability still hold. Thus, as long as you condition consistently on the same event, every one of the tools we have learned still apply. Let's look at a few of our old friends when we condition consistently on an event (in this case G):

Name of Rule	**Original Rule**	**Rule Conditioned on G**
Axiom of probability 1	$0 \leq \mathrm{P}(E) \leq 1$	$0 \leq \mathrm{P}(E\|G) \leq 1$
Axiom of probability 2	$\mathrm{P}(S) = 1$	$\mathrm{P}(S\|G) = 1$
Axiom of probability 3	$\mathrm{P}(E \text{ or } F) = \mathrm{P}(E) + \mathrm{P}(F)$ for mutually exclusive events	$\mathrm{P}(E \text{ or } F\|G) = \mathrm{P}(E\|G) + \mathrm{P}(F\|G)$ for mutually exclusive events
Identity 1	$\mathrm{P}(E^{\mathrm{C}}) = 1 - \mathrm{P}(E)$	$\mathrm{P}(E^{\mathrm{C}}\|G) = 1 - \mathrm{P}(E\|G)$

Conditioning on Multiple Events

The conditional paradigm also applies to the definition of conditional probability! Again if we consistently condition on some event G occurring, the rule still holds:

$$\mathrm{P}(E|F,G) = \frac{\mathrm{P}(E \text{ and } F|G)}{\mathrm{P}(F|G)}$$

The term $\mathrm{P}(E|F,G)$ is new notation for conditioning on multiple events. You should read that term as "The probability of E occurring, given that both F and G have occurred". This equation states that the definition for conditional probability of $E|F$ still applies in the universe where G has occurred. Do you think that $\mathrm{P}(E|F,G)$ should be equal to $\mathrm{P}(E|F)$? The answer is: sometimes yes and sometimes no.

Independence

So far we have talked about mutual exclusion as an important "property" that two or more events can have. In this chapter we will introduce you to a second property: independence. Independence is perhaps one of the most important properties to consider! Like for mutual exclusion, if you can establish that this property applies (either by logic, or by declaring it as an assumption) it will make analytic probability calculations much easier!

Definition: Independence

Two events are said to be independent if knowing the outcome of one event does not change your belief about whether or not the other event will occur. For example, you might say that two separate dice rolls are independent of one another: the outcome of the first dice gives you no information about the outcome of the second -- and vice versa.

$$\mathrm{P}(E|F) = \mathrm{P}(E)$$

Alternative Definition

Another definition of independence can be derived by using an equation called the chain rule, which we will learn about later, in the context where two events are independent. Consider two indepedent events A and B:

$$\begin{aligned} \mathrm{P}(A, B) &= \mathrm{P}(A) \cdot \mathrm{P}(B|A) && \text{Chain Rule} \\ &= \mathrm{P}(A) \cdot \mathrm{P}(B) && \text{Independence} \end{aligned}$$

Independence is Symmetric

This definition is symmetric. If E is independent of F, then F is independent of E. We can prove that $\mathrm{P}(F|E) = \mathrm{P}(F)$ implies $\mathrm{P}(E|F) = \mathrm{P}(E)$ starting with a law called Bayes' Theorem which we will cover shortly:

$$\begin{aligned} \mathrm{P}(E|F) &= \frac{\mathrm{P}(F|E) \cdot \mathrm{P}(E)}{\mathrm{P}(F)} && \text{Bayes Theorem} \\ &= \frac{\mathrm{P}(F) \cdot \mathrm{P}(E)}{\mathrm{P}(F)} && \mathrm{P}(F|E) = \mathrm{P}(F) \\ &= \mathrm{P}(E) && \text{Cancel} \end{aligned}$$

Generalized Independence

Events $E_1, E_2, \ldots, E_n$ are independent if for every subset with r elements (where $r \leq n$):

$$\mathrm{P}(E_{1'}, E_{2'}, \ldots, E_{r'}) = \prod_{i=1}^{r} \mathrm{P}(E_i')$$

As an example, consider the probability of getting 5 heads on 5 coin flips where we assume that each coin flip is independent of one another.

Let H_i be the event that the ith coin flip is a heads:

$$
\begin{aligned}
&\mathrm{P}(H_1, H_2, H_3, H_4, H_5) \\
&\quad = \mathrm{P}(H_1) \cdot \mathrm{P}(H_2) \cdots \mathrm{P}(H_5) && \text{Independence} \\
&\quad = \prod_{i=1}^{5} \mathrm{P}(H_i) && \text{Product notation} \\
&\quad = \prod_{i=1}^{5} \frac{1}{2} \\
&\quad = \frac{1}{2}^5 \\
&\quad = 0.03125
\end{aligned}
$$

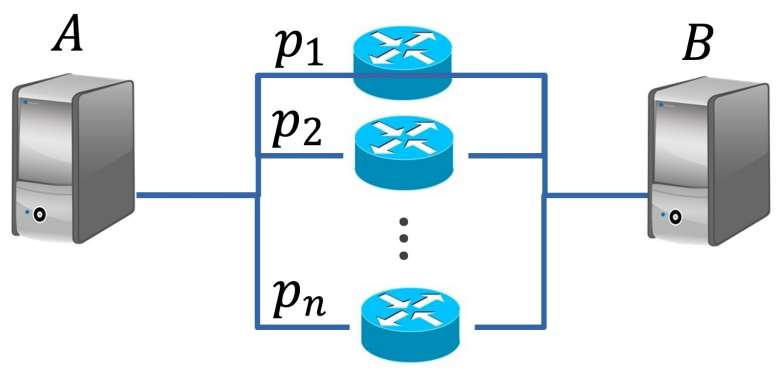
How to Establish Independence

How can you show that two or more events are independent? The default option is to show it mathematically. If you can show that $\mathrm{P}(E|F) = \mathrm{P}(E)$ then you have proven that the two events are independent. When working with probabilities that come from data, very few things will exactly match the mathematical definition of independence. That can happen for two reasons: first, events that are calculated from data or simulation are not perfectly precise and it can be impossible to know if a discrepancy between $\mathrm{P}(E)$ and $\mathrm{P}(E|F)$ is due to innacuracy in estimating probabilities, or dependence of events. Second, in our complex world many things actually influence each other, even if just a tiny amount. Despite that we often make the wrong, but useful, independence assumption. Since independence makes it so much easier for humans and machines to calculate composite probabilities, you may declare the events to be independent. It could mean your resulting calculation is slightly incorrect — but this "modelling assumption" might make it feasible to come up with a result.

Independence is a property which is often "assumed" if you think it is reasonable that one event is unlikely to influence your belief that the other will occur (or if the influence is negligible). Let's work through an example to better understand.

Example: Parallel Networks

Over networks, such as the internet, computers can send information. Often there are multiple paths (mediated by routers) between two computers and as long as one path is functional, information can be sent. Consider the following parallel network with n ***independent*** routers, each with probability p_i of functioning (where $1 \leq i \leq n$). Let E be the event that there is a functional path from A to B. What is $\mathrm{P}(E)$?



A simple network that connects two computers, A and B.

Let F_i be the event that router i fails. Note that the problem states that routers are indepenent, and as such we assume that the events F_i are all independent of one another.

$$
\begin{aligned}
\mathrm{P}(E) &= \mathrm{P}(\text{At least one router works}) \\
&= 1 - \mathrm{P}(\text{All routers fail}) \\
&= 1 - \mathrm{P}(F_1 \text{ and } F_2 \text{ and } \ldots \text{ and } F_n) \\
&= 1 - \prod_{i=1}^{n} \mathrm{P}(F_i) && \text{Independence of } F_i \\
&= 1 - \prod_{i=1}^{n} 1 - p_i
\end{aligned}
$$

Where p_i is the probability that router i is functional.

Independence and Compliments

Given independent events A and B, we can prove that A and B^C are independent. Formally we want to show that: $\mathrm{P}(AB^C) = \mathrm{P}(A)\,\mathrm{P}(B^C)$. This starts with a rule called the Law of Total Probability which we will cover shortly.

$$\begin{aligned}
\mathrm{P}(AB^C) &= \mathrm{P}(A) - \mathrm{P}(AB) && \text{LOTP} \\
&= \mathrm{P}(A) - \mathrm{P}(A)\,\mathrm{P}(B) && \text{Independence} \\
&= \mathrm{P}(A)[1 - \mathrm{P}(B)] && \text{Algebra} \\
&= \mathrm{P}(A)\,\mathrm{P}(B^C) && \text{Identity 1}
\end{aligned}$$

Conditional Independence

We saw earlier that the laws of probability still held if you consistently conditioned on an event. As such, the definition of independence also transfers to the universe of conditioned events. We use the terminology "conditional independence" to refer to events that are independent when consistently conditioned. For example if someone claims that events E_1, E_2, E_3 are conditionally independent given event F. This implies that

$$\mathrm{P}(E_1, E_2, E_3 | F) = P(E_1|F) \cdot P(E_2|F) \cdot P(E_3|F)$$

Which can be written more succinctly in product notation

$$\mathrm{P}(E_1, E_2, E_3 | F) = \prod_{i=1}^{3} P(E_i|F)$$

Warning: While the rules of probability stay the same when conditioning on an event, the independence *property* between events might change. Events that were dependent can become independent when conditioning on an event. Events that were independent can become dependent. For example, if events E_1, E_2, E_3 are conditionally independent given event F it is not necessarily true that

$$\mathrm{P}(E_1, E_2, E_3) = \prod_{i=1}^{3} P(E_i)$$

As we are no longer conditioning on F.

Probability of **and**

The probability of the ***and*** of two events, say E and F, written $\mathrm{P}(E \text{ and } F)$, is the probability of both events happening. You might see equivalent notations $\mathrm{P}(EF)$, $\mathrm{P}(E \cap F)$ and $\mathrm{P}(E, F)$ to mean the probability of and. How you calculate the probability of event E and event F happening depends on whether or not the events are "independent". In the same way that mutual exclusion makes it easy to calculate the probability of the ***or*** of events, independence is a property that makes it easy to calculate the probability of the ***and*** of events.

And with Independent Events

If events are **independent** then calculating the probability of ***and*** becomes simple multiplication:

Definition: Probability of ***and*** for independent events.
If two events: E, F are independent then the probability of E ***and*** F occurring is:

$$\mathrm{P}(E \text{ and } F) = \mathrm{P}(E) \cdot \mathrm{P}(F)$$

This property applies regardless of how the probabilities of E and F were calculated and whether or not the events are mutually exclusive.

The independence principle extends to more than two events. For n events $E_1, E_2, \ldots E_n$ that are ***mutually*** independent of one another -- the independence equation also holds for all subsets of the events.

$$\mathrm{P}(E_1 \text{ and } E_2 \text{ and } \ldots \text{ and } E_n) = \prod_{i=1}^{n} \mathrm{P}(E_i)$$

We can prove this equation by combining the definition of conditional probability and the definition of independence.

Proof: If E is independent of F then $\mathrm{P}(E \text{ and } F) = \mathrm{P}(E) \cdot \mathrm{P}(F)$

$$\begin{aligned}
\mathrm{P}(E|F) &= \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)} && \text{Definition of conditional probability} \\
\mathrm{P}(E) &= \frac{\mathrm{P}(E \text{ and } F)}{\mathrm{P}(F)} && \text{Definition of independence} \\
\mathrm{P}(E \text{ and } F) &= \mathrm{P}(E) \cdot \mathrm{P}(F) && \text{Rearranging terms}
\end{aligned}$$

See the chapter on independence to learn about when you can assume that two events are independent

And with Dependent Events

Events which are not independent are called ***dependent*** events. How can you calculate the probability of the **and** of dependent events? If your events are mutually exclusive you might be able to use a technique called DeMorgan's law, which we cover in a later chapter. For the probability of and in dependent events there is a direct formula called the chain rule which can be directly derived from the definition of conditional probability:

Definition: The chain rule.

The formula in the definition of conditional probability can be re-arranged to derive a general way of calculating the probability of the ***and*** of any two events:

$$\mathrm{P}(E \text{ and } F) = \mathrm{P}(E|F) \cdot \mathrm{P}(F)$$

Of course there is nothing special about E that says it should go first. Equivalently:

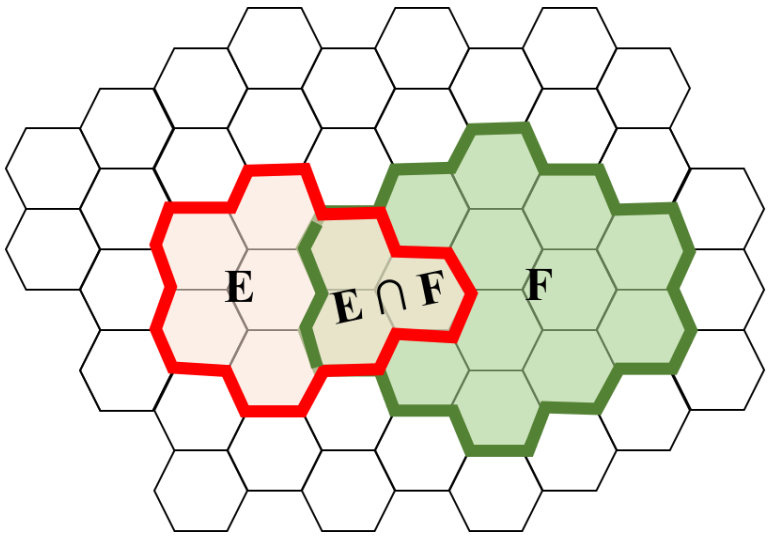
$$\mathrm{P}(E \text{ and } F) = \mathrm{P}(F \text{ and } E) = \mathrm{P}(F|E) \cdot \mathrm{P}(E)$$

We call this formula the "chain rule." Intuitively it states that the probability of observing events E ***and*** F is the probability of observing F, multiplied by the probability of observing E, given that you have observed F. It generalizes to more than two events:

$$\begin{aligned}\mathrm{P}(E_1 \text{ and } E_2 \text{ and } \ldots \text{ and } E_n) =& \mathrm{P}(E_1) \cdot \mathrm{P}(E_2|E_1) \cdot \mathrm{P}(E_3|E_1 \text{ and } E_2) \ldots \\ & \mathrm{P}(E_n|E_1 \ldots E_{n-1})\end{aligned}$$

Law of Total Probability

An astute person once observed that when looking at a picture, like the one we saw for conditional probability:



that event E can be thought of as having two parts, the part that is in F, (E and F), and the part that isn't, (E and F^{C}). This is true because F and F^{C} are (a) mutually exclusive sets of outcomes which (b) together cover the entire sample space. After further investigation this proved to be mathematically true, and there was much rejoicing:

$$\text{P}(E) = \text{P}(E \text{ and } F) + \text{P}(E \text{ and } F^{\text{C}})$$

This observation proved to be particularly useful when it was combined with the chain rule and gave rise to a tool so useful, it was given the big name, law of total probability.

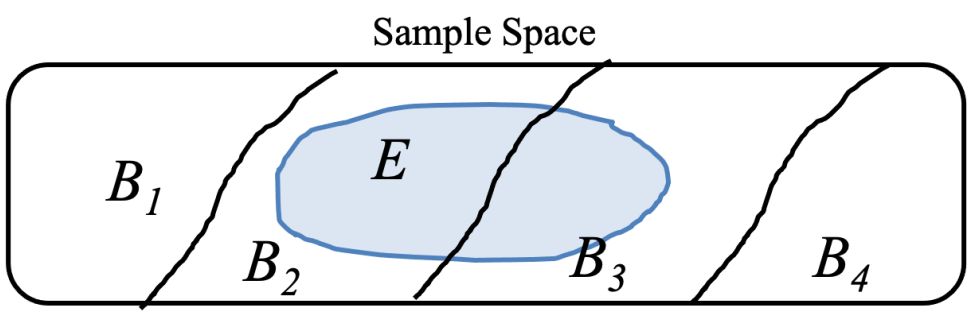
The Law of Total Probability
If we combine our above observation with the chain rule, we get a very useful formula:

$$\text{P}(E) = \text{P}(E|F)\,\text{P}(F) + \text{P}(E|F^{\text{C}})\,\text{P}(F^{\text{C}})$$

There is a more general version of the rule. If you can divide your sample space into any number of mutually exclusive events: $B_1, B_2, \ldots B_n$ such that every outcome in the sample space falls into one of those events, then:

$$\begin{aligned} \text{P}(E) &= \sum_{i=1}^{n} \text{P}(E \text{ and } B_i) && \text{Extension of our observation} \\ &= \sum_{i=1}^{n} \text{P}(E|B_i)\,\text{P}(B_i) && \text{Using chain rule on each term} \end{aligned}$$

We can build intuition for the general version of the law of total probability in a similar way. If we can divide a sample space into a set of several mutually exclusive sets (where the or of all the sets covers the entire sample space) then any event can be solved for by thinking of the likelihood of the event and each of the mutually exclusive sets.



In the image above, you could compute $\mathrm{P}(E)$ to be equal to $\mathrm{P}\left[(E \,\text{and}\, B_1) \text{ or } (E \,\text{and}\, B_2) \ldots\right]$. Of course this is worth mentioning because there are many real world cases where the sample space can be discretized into several mutual exclusive events. As an example, if you were thinking about the probability of the location of an object on earth, you could discretize the area over which you are tracking into a grid.

Bayes' Theorem

Bayes' Theorem is one of the most ubiquitous results in probability for computer scientists. In a nutshell, Bayes' theorem provides a way to convert a conditional probability from one direction, say $\mathrm{P}(E|F)$, to the other direction, $\mathrm{P}(F|E)$.

Bayes' theorem is a mathematical identity which we can derive ourselves. Start with the definition of conditional probability and then expand the and term using the chain rule:

$$\begin{aligned}\mathrm{P}(F|E) &= \frac{\mathrm{P}(F \text{ and } E)}{\mathrm{P}(E)} && \text{Def of conditional probability} \\ &= \frac{\mathrm{P}(E|F) \cdot \mathrm{P}(F)}{\mathrm{P}(E)} && \text{Substitute the chain rule for } \mathrm{P}(F \text{ and } E)\end{aligned}$$

This theorem makes no assumptions about E or F so it will apply for any two events. Bayes' theorem is exceptionally useful because it turns out to be the ubiquitous way to answer the question: "how can I update a belief about something, which is not directly observable, given evidence." This is for good reason. For many "noisy" measurements it is straightforward to estimate the probability of the noisy observation given the true state of the world. However, what you would really like to know is the conditional probability the other way around: what is the probability of the true state of the world given evidence. There are countless real world situations that fit this situation:

Example 1: Medical tests
What you want to know: Probability of a disease given a test result
What is easy to know: Probability of a test result given the true state of disease
Causality: We believe that diseases influences test results

Example 2: Student ability
What you want to know: Student knowledge of a subject given their answers
What is easy to know: Likelihood of answers given a student's knowledge of a subject
Causality: We believe that ability influences answers

Example 3: Cell phone location
What you want to know: Where is a cell phone, given noisy measure of distance to tower
What is easy to know: Error in noisy measure, given the true distance to tower
Causality: We believe that cell phone location influences distance measure

There is a pattern here: in each example we care about knowing some unobservable -- or hard to observe -- state of the world. This state of the world "causes" some easy-to-observe evidence. For example: having the flu (something we would like to know) ***causes*** a fever (something we can easily observe), not the other way around. We often call the unobservable state the "belief" and the observable state the "evidence". For that reason lets rename the events! Lets call the unobservable thing we want to know B for belief. Lets call the thing we have evidence of E for evidence. This makes it clear that Bayes' theorem allows us to calculate an updated belief given evidence: $\mathrm{P}(B|E)$

Definition: Bayes' Theorem
The most common form of Bayes' Theorem is **Bayes' Theorem Classic**:

$$\mathrm{P}(B|E) = \frac{\mathrm{P}(E|B) \cdot \mathrm{P}(B)}{\mathrm{P}(E)}$$

There are names for the different terms in the Bayes' Rule formula. The term $\mathrm{P}(B|E)$ is often called the "posterior": it is your updated belief of B after you take into account evidence E. The term $\mathrm{P}(B)$ is often called the "prior": it was your belief before seeing any evidence. The term $\mathrm{P}(E|B)$ is called the update and $\mathrm{P}(E)$ is often called the normalization constant.

There are several techniques for handling the case where the denominator is not known. One technique is to use the law of total probability to expand out the term, resulting in another formula, called **Bayes' Theorem with Law of Total Probability**:

$$\mathrm{P}(B|E) = \frac{\mathrm{P}(E|B) \cdot \mathrm{P}(B)}{\mathrm{P}(E|B) \cdot \mathrm{P}(B) + \mathrm{P}(E|B^{\mathrm{C}}) \cdot \mathrm{P}(B^{\mathrm{C}})}$$

Recall the law of total probability which is responsible for our new denominator:

$$\mathrm{P}(E) = \mathrm{P}(E|B) \cdot \mathrm{P}(B) + \mathrm{P}(E|B^{\mathrm{C}}) \cdot \mathrm{P}(B^{\mathrm{C}})$$

A common scenario for applying the Bayes' Rule formula is when you want to know the probability of something “unobservable” given an “observed” event. For example, you want to know the probability that a student understands a concept, given that you observed them solving a particular problem. It turns out it is much easier to first estimate the probability that a student can solve a problem given that they understand the concept and then to apply Bayes' Theorem. Intuitively, you can think about this as updating a belief given evidence.

Bayes' Theorem Applied

Sometimes the (correct) results from Bayes' Theorem can be counter intuitive. Here we work through a classic result: Bayes' applied to medical tests. We show a dynamic solution and present a visualization for understanding what is happening.

Example: Probability of a disease given a noisy test

In this problem we are going to calculate the probability that a patient has an illness given test-result for the illness. A positive test result means the test thinks the patient has the illness. You know the following information, which is typical for medical tests:

Natural % of population with illness: 13

Probability of a positive result given the patient has the illness 0.92

Probability of a positive result given the patient does not have the illness 0.10

The numbers in this example are from the Mammogram test for breast cancer. The seriousness of cancer underscores the potential for bayesian probability to be applied to important contexts. The natural occurrence of breast cancer is 8%. The mammogram test returns a positive result 95% of the time for patients who have breast cancer. The test resturns a positive result 7% of the time for people who do not have breast cancer. In this demo you can enter different input numbers and it will recalculate.

Answer

The probability that the patient has the illness given a positive test result is: 0.5789

Terms:
Let I be the event that the patient has the illness
Let E be the event that the test result is positive
$\mathrm{P}(I|E)$ = probability of the illness given a positive test. This is the number we want to calculate.
$\mathrm{P}(E|I)$ = probability of a positive result given illness = 0.92
$\mathrm{P}(E|I^{\mathrm{C}})$ = probability of a positive result given no illness = 0.10
$\mathrm{P}(I)$ = natural probability of the illness = 0.13

Bayes Theorem:
In this problem we know $\mathrm{P}(E|I)$ and $\mathrm{P}(E|I^{\mathrm{C}})$ but we want to know $\mathrm{P}(I|E)$. We can apply Bayes Theorem to turn our knowledge of one conditional into knowledge of the reverse.

$$\mathrm{P}(I|E) = \frac{\mathrm{P}(E|I)P(I)}{\mathrm{P}(E|I)\,\mathrm{P}(I) + \mathrm{P}(E|I^{\mathrm{C}})\,\mathrm{P}(I^{\mathrm{C}})} \qquad \text{Bayes' Theorem with Total Prob.}$$

Now all we need to do is plug values into this formula. The only value we don't explicitly have is $\mathrm{P}(I^{\mathrm{C}})$. But we can simply calculate it since $\mathrm{P}(I^{\mathrm{C}}) = 1 - \mathrm{P}(I)$. Thus:

$$P(I|E) = \frac{(0.92)(0.13)}{(0.92)(0.13) + (0.10)(1 - 0.13)} = 0.5789$$

Natural Frequency Intuition

One way to build intuition for Bayes Theorem is to think about "natural frequences". Let's take another approach to answer the probability question in the above example on belief of illness given a test. In this take, we are going to imagine we have a population of 1000 people. Let's think about how many of those have the illness and test positive and how many don't have the illness and test positive. This visualization is based off the numbers in the fields above. Feel free to change them!

There are many possibilities for how many people have the illness, but one very plausible number is 1000, the number of people in our population, multiplied by the probability of the disease.

$1000 \times \mathrm{P}(\text{Illness})$ people have the illness
$1000 \times (1 - \mathrm{P}(\text{Illness}))$ people do not have the illness.

We are going to color people who **have the illness** in blue and those **without the illness** in pink (those colors do not imply gender!).

A certain number of people **with the illness will test positive** (which we will draw in Dark Blue) and a certain number of people **without the illness will test positive** (which we will draw in Dark Pink):

$1000 \times \mathrm{P}(\text{Illness}) \times \mathrm{P}(\text{Positive}|\text{Illness})$ people have the illness and test positive
$1000 \times \mathrm{P}(\text{Illness}^{\mathrm{C}}) \times \mathrm{P}(\text{Positive}|\text{Illness}^{\mathrm{C}})$ people do not have the illness and test positive.

Here is the whole population of 1000 people:

The number of people who **test positive and have the illness** is 76.
The number of people who **test positive and don't have the illness** is 65.
The total number of people who test positive is 141.

Out of the subset of people who test positive, the fraction that have the illness is 76/141 = 0.5390 which is a close approximation of the answer. If instead of using 1000 imaginary people, we had used more, the approximation would have been even closer to the actual answer (which we calculated using Bayes Theorem).

Bayes with the General Law of Total Probability

A classic challenge when applying Bayes' theorem is to calculate the probability of the normalization constant $\mathrm{P}(E)$ in the denominator of Bayes' Theorem. One common strategy for calculating this probability is to use the law of total probability. Our expanded version of Bayes' Theorem uses the simple version of the total law of probability: $\mathrm{P}(E) = \mathrm{P}(E|F)\,\mathrm{P}(F) + \mathrm{P}(E|F^c)\,\mathrm{P}(F^c)$. Sometimes you will want the more expanded version of the law of total probability: $\mathrm{P}(E) = \sum_i \mathrm{P}(E|B_i)\,\mathrm{P}(B_i)$. Recall that this only works if the events B_i are mutually exclusive and cover the sample space.

For example say we are trying to track a phone which could be in any one of n discrete locations and we have prior beliefs $\mathrm{P}(B_1) \ldots \mathrm{P}(B_n)$ as to whether the phone is in location B_i. Now we gain some evidence (such as a particular signal strength from a particular cell tower) that we call E and we need to update all of our probabilities to be $\mathrm{P}(B_i|E)$. We should use Bayes' Theorem!

The probability of the observation, assuming that the the phone is in location B_i, $\mathrm{P}(E|B_i)$, is something that can be given to you by an expert. In this case the probability of getting a particular signal strength given a location B_i will be determined by the distance between the cell tower and location B_i .

Since we are assuming that the phone must be in *exactly* one of the locations, we can find the probability of any of the event B_i given E by first applying Bayes' Theorem and then applying the general version of the law of total probability:

$$\begin{aligned}
\mathrm{P}(B_i|E) &= \frac{\mathrm{P}(E|B_i) \cdot \mathrm{P}(B_i)}{\mathrm{P}(E)} && \text{Bayes Theorem. What to do about } \mathrm{P}(E)\text{?} \\
&= \frac{\mathrm{P}(E|B_i) \cdot \mathrm{P}(B_i)}{\sum_{j=1}^{n} \mathrm{P}(E|B_j) \cdot \mathrm{P}(B_j)} && \text{Use General Law of Total Probability for } \mathrm{P}(E)
\end{aligned}$$

Unknown Normalization Constant

There are times when we would like to use Bayes' Theorem to update a belief, but we don't know the probability of E, $\mathrm{P}(E)$. All hope is not lost. This term is called the "normalization constant" because it is the same regardless of whether or not the event B happens. The solution that we used above was the law of total probability: $\mathrm{P}(E) = \mathrm{P}(E|B)\,\mathrm{P}(B) + \mathrm{P}(E|B^{\mathrm{C}})\,\mathrm{P}(B^{\mathrm{C}})$. This allows us to calculate $\mathrm{P}(E)$.

Here is another strategy for dealing with an unknown $\mathrm{P}(E)$. We can make it cancel out by calculating the ratio of $\frac{\mathrm{P}(B|E)}{\mathrm{P}(B^{\mathrm{C}}|E)}$. This fraction tells you how many times more likely it is that B will happen given E than not B:

$$\begin{aligned}
\frac{\mathrm{P}(B|E)}{\mathrm{P}(B^{\mathrm{C}}|E)} &= \frac{\frac{\mathrm{P}(E|B)\,\mathrm{P}(B)}{\mathrm{P}(E)}}{\frac{\mathrm{P}(E|B^{\mathrm{C}})\,\mathrm{P}(B^{\mathrm{C}})}{\mathrm{P}(E)}} && \text{Apply Bayes' Theorem to both terms} \\
&= \frac{\mathrm{P}(E|B)\,\mathrm{P}(B)}{\mathrm{P}(E|B^{\mathrm{C}})\,\mathrm{P}(B^{\mathrm{C}})} && \text{The term } \mathrm{P}(E) \text{ cancels}
\end{aligned}$$

Log Probabilities

A log probability $\log \mathrm{P}(E)$ is simply the log function applied to a probability. For example if $\mathrm{P}(E) = 0.00001$ then $\log \mathrm{P}(E) = \log(0.00001) \approx -11.51$. Note that in this book, the default base is the natural base e. There are many reasons why log probabilities are an essential tool for digital probability: (a) computers can be rather limited when representing very small numbers and (b) logs have the wonderful ability to turn multiplication into addition, and computers are much faster at addition.

You may have noticed that the log in the above example produced a negative number. Recall that $\log b = c$, with the implied natural base e is the same as the statement $e^c = b$. It says that c is the exponent of e that produces b. If b is a number between 0 and 1, what power should you raise e to in order to produce b? If you raise e^0 it produces 1. To produce a number less than 1, you must raise e to a power less than 0. That is a long way of saying: if you take the log of a probability, the result will be a negative number.

$$\begin{aligned} 0 \leq \mathrm{P}(E) \leq 1 \qquad & \text{Axiom 1 of probability} \\ -\infty \leq \log \mathrm{P}(E) \leq 0 \qquad & \text{Rule for log probabilities} \end{aligned}$$

Products become Addition

The product of probabilities $\mathrm{P}(E)$ and $\mathrm{P}(F)$ becomes addition in logarithmic space:

$$\log(\mathrm{P}(E) \cdot \mathrm{P}(F)) = \log \mathrm{P}(E) + \log \mathrm{P}(F)$$

This is especially convenient because computers are much more efficient when adding than when multiplying. It can also make derivations easier to write. This is especially true when you need to multiply many probabilities together:

$$\log \prod_i \mathrm{P}(E_i) = \sum_i \log \mathrm{P}(E_i)$$

Representing Very Small Probabilities

Computers have the power to process many events and consider the probability of very unlikely situations. While computers are capable of doing all the computation, the floating point representation means that computers can not represent decimals to perfect precision. In fact, python is unable to represent any probability smaller than 2.225e-308. On the other hand the log of that same number is -307.652 is very easy for a computer to store.

Why would you care? Often in the digital world, computers are asked to reason about the probability of data, or a whole dataset. For example, perhaps your data is words and you want to reason about the probability that a given author would write these specific words. While this probability is very small (we are talking about an exact document) it might be larger than the probability that a different author would write a specific document with specific words. For these sort of small probabilities, if you use computers, you would need to use log probabilities.

Many Coin Flips

In this section we are going to consider the number of heads on n coin flips. This thought experiment is going to be a basis for much probability theory! It goes far beyond coin flips.

Say a coin comes up heads with probability p. Most coins are fair and as such come up heads with probability $p = 0.5$. There are many events for which coin flips are a great analogy that have different values of p so lets leave p as a variable. You can try simulating coins here. Note that `H` is short for Heads and `T` is short for Tails. We think of each coin as distinct:

Coin Flip Simulator

Number of flips n: 10 Probability of heads p: 0.60 New simulation

Simulator results:

```
H, H, H, H, T, H, H, H, H, T
```

Total number of heads: 8

Let's explore a few probability questions in this domain.

Warmups

What is the probability that all n flips are heads?

Lets say $n = 10$ this question is asking what is the probability of getting:

```
H, H, H, H, H, H, H, H, H, H
```

Each coin flip is independent so we can use the rule for probability of **and** with independent events. As such, the probability of n heads is p multiplied by itself n times: p^n. If $n = 10$ and $p = 0.6$ then the probability of n heads is around 0.006.

What is the probability that all n flips are tails?

Lets say $n = 10$ this question is asking what is the probability of getting:

```
T, T, T, T, T, T, T, T, T, T
```

Each coin flip is independent. The probability of tails on any coin flip is $1 - p$. Again, since the coin flips are independent, the probability of tails n times on n flips is $(1 - p)$ multiplied by itself n times: $(1 - p)^n$. If $n = 10$ and $p = 0.6$ then the probability of n tails is around 0.0001.

First k heads then $n - k$ tails

Lets say $n = 10$ and $k = 4$, this question is asking what is the probability of getting:

```
H, H, H, H, T, T, T, T, T, T
```

The coins are still independent! The first k heads occur with probability p^k the run of $n - k$ tails occurs with probability $(1 - p)^{n-k}$. The probability of k heads then $n - k$ tails is the product of those two

terms: $p^k \cdot (1-p)^{n-k}$

Exactly k heads

Next lets try to figure out the probability of exactly k heads in the n flips. Importantly we don't care where in the n flips that we get the heads, as long as there are k of them. Note that this question is different than the question of first k heads and then $n-k$ tails which requires that the k heads come first! That particular result does generate exactly k coin flips, but there are others.

There are many others! In fact any permutation of k heads and $n-k$ tails will satisfy this event. Lets ask the computer to list them all for exactly $k=4$ heads within $n=10$ coin flips. The output region is scrollable:

```
(H, H, H, H, T, T, T, T, T, T)
(H, H, H, T, H, T, T, T, T, T)
(H, H, H, T, T, H, T, T, T, T)
(H, H, H, T, T, T, H, T, T, T)
(H, H, H, T, T, T, T, H, T, T)
(H, H, H, T, T, T, T, T, H, T)
(H, H, H, T, T, T, T, T, T, H)
(H, H, T, H, H, T, T, T, T, T)
(H, H, T, H, T, H, T, T, T, T)
(H, H, T, H, T, T, H, T, T, T)
(H, H, T, H, T, T, T, H, T, T)
(H, H, T, H, T, T, T, T, H, T)
(H, H, T, H, T, T, T, T, T, H)
(H, H, T, T, H, H, T, T, T, T)
(H, H, T, T, H, T, H, T, T, T)
(H, H, T, T, H, T, T, H, T, T)
(H, H, T, T, H, T, T, T, H, T)
(H, H, T, T, H, T, T, T, T, H)
```

Exactly how many outcomes are there with $k=4$ heads in $n=10$ flips? 210. The answer can be calculated using permutations of indistinct objects:

$$N = \frac{n!}{k!(n-k)!} = \binom{n}{k}$$

The probability of exactly $k=4$ heads is the probability of the **or** of each of these outcomes. Because we consider each coin to be unique, each of these outcomes is "mutually exclusive" and as such if E_i is the outcome from the ith row,

$$\mathrm{P}(\text{exactly } k \text{ heads}) = \sum_{i=1}^{N} \mathrm{P}(E_i)$$

The next question is, what is the probability of each of these outcomes?

Here is a arbitrarily chosen outcome which satisfies the event of exactly $k=4$ heads in $n=10$ coin flips. In fact it is the one on row 128 in the list above:

```
T, H, T, T, H, T, T, H, H, T
```

What is the probability of getting the exact sequence of heads and tails in the example above? Each coin flip is still independent, so we multiply p for each heads and $1-p$ for each tails. Let E_{128} be the event of this exact outcome:

$$\mathrm{P}(E_{128}) = (1-p) \cdot p \cdot (1-p) \cdot (1-p) \cdot p \cdot (1-p) \cdot (1-p) \cdot p \cdot p \cdot (1-p)$$

If you rearrange these multiplication terms you get:

$$\begin{aligned} \mathrm{P}(E_{128}) &= p \cdot p \cdot p \cdot p \cdot (1-p) \cdot (1-p) \cdot (1-p) \cdot (1-p) \cdot (1-p) \cdot (1-p) \\ &= p^4 \cdot (1-p)^6 \end{aligned}$$

There is nothing too special about row 128. If you chose any row, you would get k independent heads and $n-k$ independent tails. For any row i, $\mathrm{P}(E_i) = p^k \cdot (1-p)^{n-k}$. Now we are ready to calculate the probability of exactly k heads:

$$\begin{aligned}
\mathrm{P}(\text{exactly } k \text{ heads}) &= \sum_{i=1}^{N} \mathrm{P}(E_i) && \text{Mutual Exclusion} \\
&= \sum_{i=1}^{N} p^k \cdot (1-p)^{n-k} && \text{Sub in } \mathrm{P}(E_i) \\
&= N \cdot p^k \cdot (1-p)^{n-k} && \text{Sum } N \text{ times} \\
&= \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} && \text{Perm of indistinct objects}
\end{aligned}$$

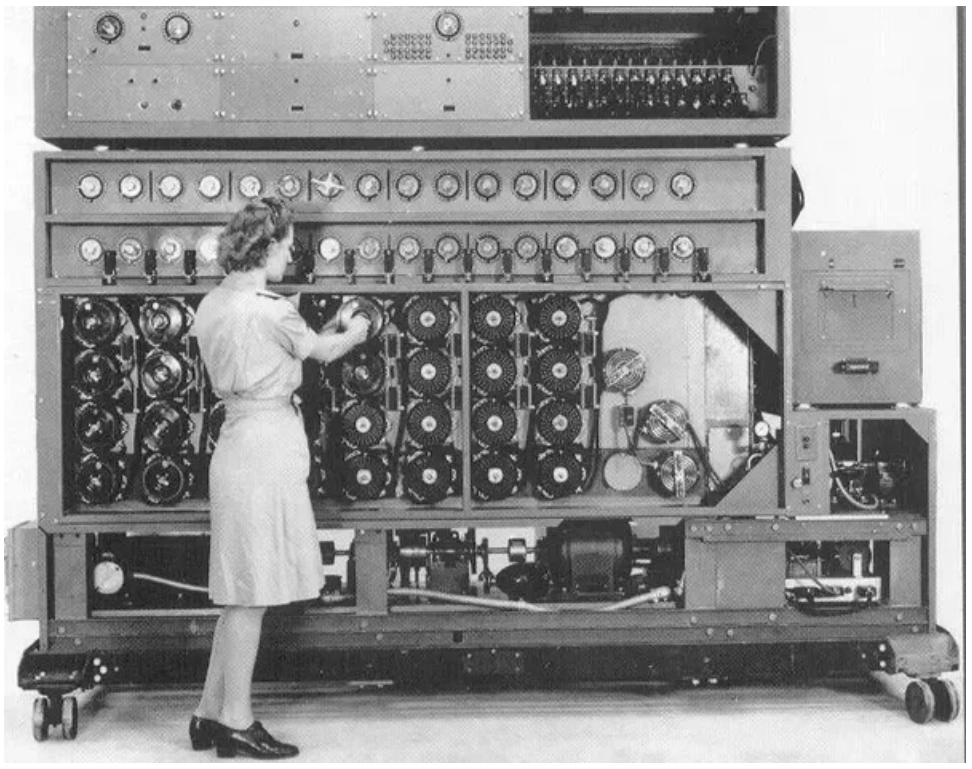
More than k heads

You can use the formula for exactly k heads to compute other probabilities. For example the probability of more than k heads is:

$$\begin{aligned}
\mathrm{P}(\text{more than } k \text{ heads}) &= \sum_{i=k+1}^{n} \mathrm{P}(\text{exactly } i \text{ heads}) && \text{Mutual Exclusion} \\
&= \sum_{i=k+1}^{n} \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} && \text{Substitution}
\end{aligned}$$

Enigma Machine

One of the very first computers was built to break the Nazi “enigma” codes in WW2. It was a hard problem because the “enigma” machine, used to make secret codes, had so many unique configurations. Every day the Nazis would choose a new configuration and if the Allies could figure out the daily configuration, they could read all enemy messages. One solution was to try all configurations until one produced legible German. This begs the question: How many configurations are there?

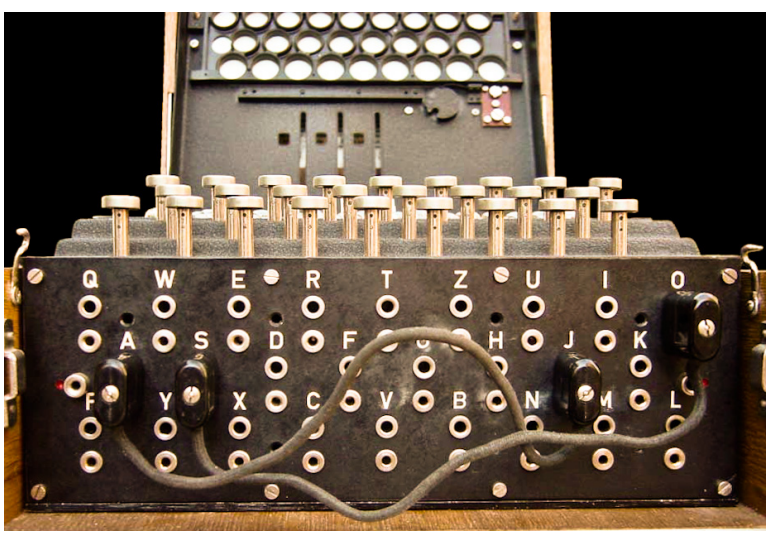


The WW2 machine built to search different enigma configurations.

The enigma machine has three rotors. Each rotor can be set to one of 26 different positions. How many unique configurations are there of the three rotors?

Using the steps rule of counting: $26 \cdot 26 \cdot 26 = 26^3 = 17,576$.

Whats more! The machine has a plug board which could swap the electrical signal for letters. On the plug board, wires can connect any pair of letters to produce a new configuration. A wire can’t connect a letter to itself. Wires are indistinct. A wire from ‘K’ to ’L’ is not considered distinct from a wire from ‘L’ to ’K’. We are going to work up to considering any number of wires.



The engima plugboard. For electrical reasons, each letter has two jacks and each plug has two prongs. Semantically this is equivalent to one plug location per letter.

One wire: How many ways are there to place exactly one wire that connects two letters?

Chosing 2 letters from 26 is a combination. Using the combination formula: $\binom{26}{2} = 325$.

Two wires: How many ways are there to place exactly two wires? Recall that wires are not considered distinct. Each letter can have at most one wire connected to it, thus you couldn't have a wire connect 'K' to 'L' and another one connect 'L' to 'X'

There are $\binom{26}{2}$ ways to place the first wire and $\binom{24}{2}$ ways to place the second wire. However, since the wires are indistinct, we have double counted every possibility. Because every possibility is counted twice we should divide by 2:

$$\text{Total} = \frac{\binom{26}{2} \cdot \binom{24}{2}}{2} = 44,850$$

Three wires: How many ways are there to place **exactly** three wires?

There are $\binom{26}{2}$ ways to place the first wire and $\binom{24}{2}$ ways to place the second wire. There are now $\binom{22}{2}$ ways to place the third. However, since the wires are indistinct, and our step counting implicitly treats them as distinct, we have overcounted each possibility. How many times is each pairing of three letters overcounted? It's the number of permutations of three distinct objects: 3!

$$\text{Total} = \frac{\binom{26}{2} \cdot \binom{24}{2} \cdot \binom{22}{2}}{3!} = 3,453,450$$

There is another way to arrive at the same answer. First we are going to choose the letters to be paired, then we are going to pair them off. There are $\binom{26}{6}$ ways to select the letters that are being wired up. We then need to pair off those letters. One way to think about pairing the letters off is to first permute them (6! ways) and then pair up the first two letters, the next two, the next two, and so on. For example, if our letters were {A,B,C,D,E,F} and our permutation was BADCEF, then this would correspond to wiring B to A and D to C and E to F. We are overcounting by a lot. First, we are overcounting by a factor of $3!$ since the ordering of the pairs doesn't matter. Second, we are overcounting by a factor of 2^3 since the ordering of the letters within each pair doesn't matter.

$$\text{Total} = \binom{26}{6} \frac{6!}{3! \cdot 2^3} = 3,453,450$$

Arbitrary wires: How many ways are there to place k wires, thus connecting $2 \cdot k$ letters? During WW2 the Germans always used a fixed number of wires. But one fear was that if they discovered the Enigma machine was cracked, they could simply use an arbitrary number of wires.

The set of ways to use exactly i wires is mutually exclusive from the set of ways to use exactly j wires if $i \neq j$ (since no way can use both exactly i and j wires). As such $\text{Total} = \sum_{k=0}^{13} \text{Total}_k$ Where Total_k is the number of ways to use exactly k wires. Continuing our logic for ways to used exact number of wires:

$$\text{Total}_k = \frac{\prod_{i=1}^{k} \binom{28-2i}{2}}{k!}$$

Bringing it all together:

$$\begin{aligned}
\text{Total} &= \sum_{k=0}^{13} \text{Total}_k \\
&= \sum_{k=0}^{13} \frac{\prod_{i=1}^{k} \binom{28-2i}{2}}{k!} \\
&= 532,985,208,200,576
\end{aligned}$$

The actual Enigma used in WW2 had exactly 10 wires connecting 20 letters allowing for 150,738,274,937,250 unique configuration. The enigma machine also chose the three rotors from a set of five adding another factor of $\binom{5}{3} = 60$.

When you combine the number of ways of setting the rotors, with the number of ways you could set the plug board you get the total number of configurations of an enigma machine. Thinking of this as two steps we can multiply the two numbers we earlier calculated: 17,576 · 150,738,274,937,250 · 60 $\approx 159 \cdot 10^{18}$ unique settings. So, Alan Turing and his team at Blechly Park went on to build a machine which could help test many configurations -- a predecessor to the first computers.

Serendipity

The word serendipity comes from the Persian fairy tale of the [Three Princes of Serendip]

Problem

What is the probability of a seredipitous encounter with a friend? Imagine you are live in an area with a large general population (eg Stanford with 17,000 students). A small subset of the population are friends. What are the chances that you run into at least one friend if you see a handful of people from the population? Assume that seeing each person from the population is equally likely.

Total Population

17000

Friends

150

People that you see

100

Calculate

Answer

The probability that you see at least one friend is:

Random Shuffles

Here is a suprising claim. If you shuffle a standard deck of cards seven times, with almost total certainty you can claim that the exact ordering of cards has never been seen! Wow! Let's explore. We can ask this question formally as: What is the probability that in the n shuffles seen since the start of time, yours is unique?

Orderings of 52 Cards

Our adventure starts with a simple observation: there are **very** many ways to order 52 cards. But exactly how many unique orderings of a standard deck are there?

There are 52! ways to order a standard deck of cards. Since each card is unique (each card has a unique suit, value combination) then we can apply the rule for Permutations of Distinct Objects:

$$\text{Num. Unique Orderings} = 52!$$

That is a humongous number. 52! equals
80658175170943878571660636856403766975289505440883277824000000000000.
That is over $8 \cdot 10^{67}$. Recall it is estimated that there are around 10^{82} atoms in the observable universe

Number of Shuffles Ever Seen

Of course we don't know what the value of n is — nobody has been counting how many times humans have shuffled cards. We can come up with a reasonable overestimate. Assume k = 7 billion people have been shuffling cards once a second since cards were invented. Playing cards may have been invented as far back as the Tang dynasty in the 9th century. To the best of my knowledge the oldest set of 52 cards is the Topkapı deck of cards in Istanbul around the 15th century ad. That is about s = 16,472,828,422 seconds ago. As such our overestimate is $n = s \cdot k \approx 10^{20}$.

Next let's calculate the probability that none of those n historical shuffles matches your particular ordering of 52 cards. There are two valid approaches: using equally likely outcomes, and using independence.

Equally Likely Outcomes

One way to the probability that your ordering of 52 cards is unique in history is to use Equally Likely Outcomes. Consider the sample space of all the possible ordering of all the cards ever dealt. Each outcome in this set will have n card decks each with their own ordering. As such the size of the sample space is $|S| = (52!)^n$. Note that all outcomes in the sample space are equally likely — we can convince ourselves of this by symmetry — no ordering is more likely than any other. Out of that sample space we want to count the number of outcomes where none of the orderings matches yours. There are $52! - 1$ ways to order 52 cards that are not yours. We can construct the event space by steps: for each of the n shuffles in history select any one of those $52! - 1$ orderings. Thus $|E| = (52! - 1)^n$.

Let U be the event that your particular ordering of 52 cards is unique

$$
\begin{aligned}
\mathrm{P}(U) &= \frac{|E|}{|S|} && \text{Equally Likely Outcomes} \\
&= \frac{(52!-1)^n}{(52!)^n} \\
&= \frac{(52!-1)^{10^{20}}}{(52!)^{10^{20}}} && n = 10^{20} \\
&= \left(\frac{52!-1}{52!}\right)^{10^{20}}
\end{aligned}
$$

In theory that is the correct answer, but those numbers are so big, its not clear how to evaluate it, even when using a computer. One good idea is to first compute the log probability:

$$
\begin{aligned}
\log \mathrm{P}(U) &= \log\left[\left(\frac{52!-1}{52!}\right)^{10^{20}}\right] \\
&= 10^{20} \cdot \log\left(\frac{52!-1}{52!}\right) \\
&= 10^{20} \cdot \Big[\log(52!-1) - \log(52!)\Big] \\
&= 10^{20} \cdot (-1.24 \times 10^{-68}) \\
&= -1.24 \times 10^{-48}
\end{aligned}
$$

Now if we undo the log (and use the fact that e^{-x} is very close to $1-x$ for small values of x):

$$
\begin{aligned}
\mathrm{P}(U) &= e^{-1.24 \times 10^{-48}} \\
&\approx 1 - 1.24 \times 10^{-48}
\end{aligned}
$$

So the probability that your particular ordering is unique is very close to 1, and the probability that someone else got the same ordering, $1 - P(U)$, is a number with 47 zeros after the decimal point. It is safe to say your ordering is unique.

In python, you can use a special library called decimal to compute very small probabilities. Here is an example of how to compute $\log \frac{52!-1}{52!}$:

```python
from decimal import *
import math

n = math.pow(10, 20)
card_perms = math.factorial(52)
denominator = card_perms
numerator = card_perms - 1

# decimal library because these are tiny numbers
getcontext().prec = 100 # increase precision
log_numer = Decimal(numerator).ln()
log_denom = Decimal(denominator).ln()
log_pr = log_numer - log_denom

# approximately -1.24E-68
print(log_pr)
```

We can also check our result using the binomial approximation.

For small values of x, the value $(1-x)^n$ is very close to $1 - nx$, and this gives us another way to compute $P(U)$:

$$
\begin{aligned}
\mathrm{P}(U) &= \frac{(52!-1)^n}{(52!)^n} \\
&= \left(1 - \frac{1}{52!}\right)^{10^{20}} && n = 10^{20} \\
&\approx 1 - \frac{10^{20}}{52!} \\
&\approx 1 - 1.24 \times 10^{-48}
\end{aligned}
$$

This agrees with the result we got using python's decimal library.

Independence

Another approach is to define events D_i that the ith card shuffle is different than yours. Because we assume each shuffle is independent, then $P(U) = \prod_i P(D_i)$. What is the probability of (D_i)? If you think of the sample space of D_i, it is 52! ways of ordering a deck cards. The event space is the 52! - 1 outcomes which are not your ordering.

$$
\begin{aligned}
\mathrm{P}(U) &= \prod_{i=1}^{n} \mathrm{P}(D_i) \\
\log \mathrm{P}(U) &= \sum_{i=1}^{n} \log \mathrm{P}(D_i) \\
&= n \cdot \log \mathrm{P}(D_i) \\
&= 10^{20} \cdot \log \frac{52! - 1}{52!} \\
&\approx 10^{20} \cdot -1.24 \times 10^{-68}
\end{aligned}
$$

Which is the same answer we got with the other approach for $\log \mathrm{P}(U)$

How Random is your Shuffle?

A final question we can look into. How do you get a truly random ordering of cards? Dave Bayer and Persi Diaconis in 1992 worked through this problem and published their results in the article Trailing the Dovetail Shuffle to its Lair. They showed that if you shuffle a deck of cards seven times using a riffle shuffle also known as the dovetail shuffle, you are almost garunteed a random ordering of cards. The methodology used paved the way for studying psuedo random numbers produced by computers.

Bacteria Evolution

A wonderful property of modern life is that we have anti-biotics to kill bacterial infections. However, we only have a fixed number of anti-biotic medicines, and bacteria are evolving to become resistent to our anti-biotics. In this example we are going to use probability to understand evolution of anti-biotic resistence in bacteria.

Imagine you have a population of 1 million infectious bacteria in your gut, 10% of which have a mutation that makes them slightly more resistant to anti-biotics. You take a course of anti-biotics. The probability that bacteria with the mutation survives is 20%. The probability that bacteria without the mutation survives is 1%.

What is the probability that a randomly chosen bacterium survives the anti-biotics?

Let E be the event that our bacterium survives. Let M be the event that a bacteria has the mutation. By the Law of Total Probability (LOTP):

$$\begin{aligned} \mathrm{P}(E) &= \mathrm{P}(E \text{ and } M) + \mathrm{P}(E \text{ and } M^{\mathrm{C}}) && \text{LOTP} \\ &= \mathrm{P}(E|M)\,\mathrm{P}(M) + \mathrm{P}(E|M^{\mathrm{C}})\,\mathrm{P}(M^{\mathrm{C}}) && \text{Chain Rule} \\ &= 0.20 \cdot 0.10 + 0.01 \cdot 0.90 && \text{Substituting} \\ &= 0.029 \end{aligned}$$

What is the probability that a surviving bacterium has the mutation?

Using the same events in the last section, this question is asking for $\mathrm{P}(M|E)$. We aren't giving the conditional probability in that direction, instead we know $P(E|M)$. Such situations call for Bayes' Theorem:

$$\begin{aligned} \mathrm{P}(M|E) &= \frac{\mathrm{P}(E|M)\,\mathrm{P}(M)}{\mathrm{P}(E)} && \text{Bayes} \\ &= \frac{0.20 \cdot 0.10}{\mathrm{P}(E)} && \text{Given} \\ &= \frac{0.20 \cdot 0.10}{0.029} && \text{Calculated} \\ &\approx 0.69 \end{aligned}$$

After the course of anti-biotics, 69% of bacteria have the mutation, up from 10% before. If this population is allowed to reproduce you will have a much more resistent set of bacteria!

Random Variables

A Random Variable (RV) is a variable that probabilistically takes on a value and they are one of the most important constructs in all of probability theory. You can think of an RV as being like a variable in a programming language, and in fact random variables are just as important to probability theory as variables are to programming. Random Variables take on values, have types and have domains over which they are applicable.

Random variables work with all of the foundational theory we have built up to this point. We can define events that occur if the random variable takes on values that satisfy a numerical test (eg does the variable equal 5, is the variable less than 8).

Lets look at a first example of a random variable. Say we flip three fair coins. We can define a random variable Y to be the total number of “heads” on the three coins. We can ask about the probability of Y taking on different values using the following notation:

Let Y be the number of heads on three coin flips

$\mathrm{P}(Y = 0) = 1/8$	(T, T, T)
$\mathrm{P}(Y = 1) = 3/8$	(H, T, T), (T, H, T), (T, T, H)
$\mathrm{P}(Y = 2) = 3/8$	(H, H, T), (H, T, H), (T, H, H)
$\mathrm{P}(Y = 3) = 1/8$	(H, H, H)
$\mathrm{P}(Y \geq 4) = 0$	

Even though we use the same notation for random variables and for events (both use capital letters) they are distinct concepts. An event is a scenario, a random variable is an object. The scenario where a random variable takes on a particular value (or range of values) is an event. When possible, I will try and use letters E,F,G for events and X,Y,Z for random variables.

Using random variables is a convenient notation technique that assists in decomposing problems. There are many different types of random variables (indicator, binary, choice, Bernoulli, etc). The two main families of random variable types are discrete and continuous. Discrete random variables can only take on integer values. Continuous random variables can take on decimal values. We are going to develop our intuitions using discrete random variable and then introduce continuous.

Properties of random variables

There are many properties of a random variable some of which we will dive into extensively. Here is a brief summary. Each random variable has:

Property	Notation Example	Description
Meaning		A semantic description of the random variable
Symbol	X	A letter used to denote the random variable
Support or Range	$\{0, 1, \ldots, 3\}$	the values the random variable can take on
Distribution Function (PMF or PDF)	$\mathrm{P}(X = x)$	A function which maps values the RV can take on to likelihood.

Property	Notation Example	Description
Expectation	$\mathrm{E}[X]$	A weighted average
Variance	$\mathrm{Var}(X)$	A measure of spread
Standard Deviation	$\mathrm{Std}(X)$	The square root of variance
Mode		The most likely value of the random variable

You should set a goal of deeply understanding what each of these properties mean. There are many more properties than the ones in the table above: properties like entropy, median, skew, kurtosis.

Random variables vs Events

Random variables and events are two different concepts. An event is an outcome, or a set of outcomes, to an experiment. A random variable is a more like an experiment -- it will take on an outcome eventually. Probabilities are over events, so if you want to talk about probability in the context of a random variable, you must construct an event. You can make events by using any of the Relational Operators: $<$, $\leq$, $>$, $\geq$, $=$, or $\neq$ (not equal to). This is analogous to coding where you can use relational operators to create boolean expressions from numbers.

Lets continue our example of the random variable Y which represents the number of heads on three coin flips. Here are some events using the variable Y:

Event	Meaning	Probability Statement
$Y = 1$	Y takes on the value 1 (there was one heads)	$\mathrm{P}(Y = 1)$
$Y < 2$	Y takes on 0 or 1 (note this Y can't be negative)	$\mathrm{P}(Y < 2)$
$X > Y$	X takes on a value greater than the value Y takes on.	$\mathrm{P}(X > Y)$
$Y = y$	Y takes on a value represented by non-random variable y	$\mathrm{P}(Y = y)$

You will see many examples like this last one, $\mathrm{P}(Y = y)$, in this text book as well as in scientific and math research papers. It allows us to talk about the likelihood of Y taking on a value, in general. For example, later in this book we will derive that for three coin flips where Y is the number of heads, the probability of getting exactly y heads is:

$$\mathrm{P}(Y = y) = \frac{0.75}{y!(3-y)!} \qquad \text{If } 0 \leq y \leq 3$$

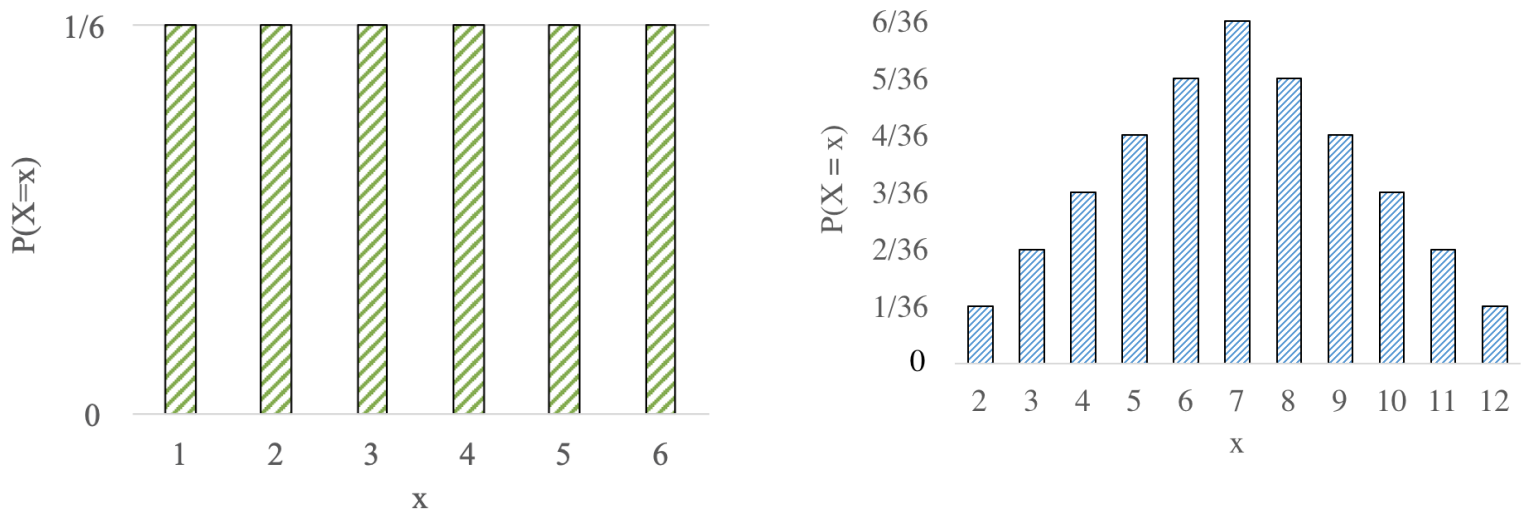
This statement above is a function which takes in a parameter y as input and returns the numeric probability $\mathrm{P}(Y = y)$ as output. This particular expression allows us to talk about the probability that the number of heads is 0, 1, 2 or 3 all in one expression. You can plug in any one of those values for y to get the corresponding probability. It is customary to use lower-case symbols for non-random values. The use of an equals sign in the "event" can be confusing. For example what does this expression say $\mathrm{P}(Y = 1) = 0.375$? It says that the probability that "Y takes on the value 1" is 0.375. For discrete random variables this function is called the "probability mass function" and it is the topic of our next chapter.

Probability Mass Functions

For a random variable, the most important thing to know is: how likely is each outcome? For a discrete random variable, this information is called the "**probability mass function**". The probability mass function (PMF) provides the "mass" (i.e. amount) of "probability" for each possible assignment of the random variable.

Formally, the probability mass function is a mapping between the values that the random variable could take on and the probability of the random variable taking on said value. In mathematics, we call these associations functions. There are many different ways of representing functions: you can write an equation, you can make a graph, you can even store many samples in a list. Let's start by looking at PMFs as graphs where the x-axis is the values that the random variable could take on and the y-axis is the probability of the random variable taking on said value.

In the following example, on the left we show a PMF, as a graph, for the random variable: X = the value of a six-sided die roll. On the right we show a contrasting example of a PMF for the random variable X = value of the sum of two dice rolls:



Left: the PMF of a single six-sided die roll. Right: the PMF of the sum of two dice rolls.

The sum of two dice example in the equally likely probability section. Again, the information that is provided in these graphs is the likelihood of a random variable taking on different values. In the graph on the right, the value "6" on the x-axis is associated with the probability $\frac{5}{36}$ on the y-axis. This x-axis refers to the event "the sum of two dice is 6" or $Y = 6$. The y-axis tells us that the probability of that event is $\frac{5}{36}$. In full: $\mathrm{P}(Y = 6) = \frac{5}{36}$. The value "2" is associated with "$\frac{1}{36}$" which tells us that, $\mathrm{P}(Y = 2) = \frac{1}{36}$, the probability that two dice sum to 2 is $\frac{1}{36}$. There is no value associated with "1" because the sum of two dice can not be 1. If you find this notation confusing, revisit the random variables section.

Here is the exact same information in equation form:

$$\mathrm{P}(X = x) = \frac{1}{6} \qquad \text{if } 1 \leq x \leq 6$$

$$\mathrm{P}(Y = y) = \begin{cases} \frac{(y-1)}{36} & \text{if } 1 \leq y \leq 7 \\ \frac{(13-y)}{36} & \text{if } 8 \leq y \leq 12 \end{cases}$$

As a final example, here is the PMF for Y, the sum of two dice, in Python code:

```python
def pmf_sum_two_dice(y):
    # Returns the probability that the sum of two dice is y
    if y < 2 or y > 12:
        return 0
    if y <= 7:
        return (y-1) / 36
    else:
        return (13-y) / 36
```

Notation

You may feel that $\mathrm{P}(Y = y)$ is redundant notation. In probability research papers and higher-level work, mathematicians often use the shorthand $\mathrm{P}(y)$ to mean $\mathrm{P}(Y = y)$. This shorthand assumes that the lowercase value (e.g. y) has a capital letter counterpart (e.g. Y) that represents a random variable even though it's not written explicitly. In this book, we will often use the full form of the event $\mathrm{P}(Y = y)$, but we will occasionally use the shorthand $\mathrm{P}(y)$.

Probabilities Must Sum to 1

For a variable (call it X) to be a proper random variable it must be the case that if you summed up the values of $\mathrm{P}(X = k)$ for all possible values k that X can take on, the result must be 1:

$$\sum_k \mathrm{P}(X = k) = 1$$

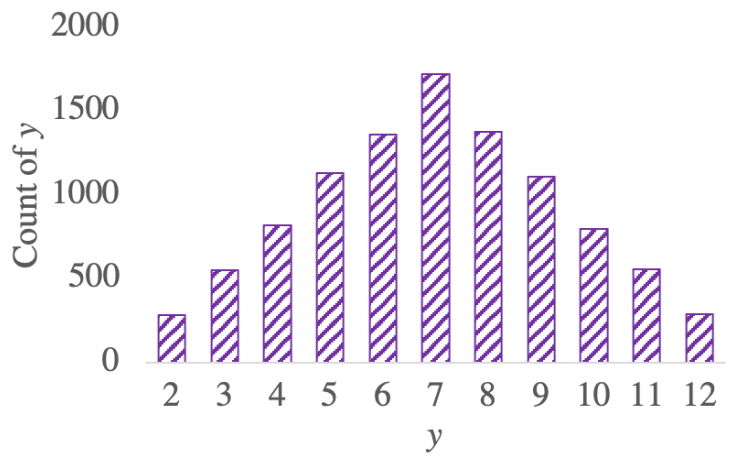
For further understanding, let's derive why this is the case. A random variable taking on a value is an event (for example $X = 2$). Each of those events is mutually exclusive because a random variable will take on exactly one value. Those mutually exclusive cases define an entire sample space. Why? Because X must take on *some* value.

Data to Histograms to Probability Mass Functions

One surprising way to store a likelihood function (recall that a PMF is the name of the likelihood function for *discrete* random variables) is simply a list of data. We simulated summing two die 10,000 times to make this example dataset:

```
[8, 4, 9, 7, 7, 7, 7, 5, 6, 8, 11, 5, 7, 7, 7, 6, 7, 8, 8, 9, 9, 4, 6, 7, 10, 12, 6,
7, 8, 9, 3, 7, 4, 9, 2, 8, 5, 8, 9, 6, 8, 7, 10, 7, 6, 7, 7, 5, 4, 6, 9, 5, 7, 4, 2,
11, 10, 11, 8, 4, 11, 9, 7, 10, 12, 4, 8, 5, 11, 5, 3, 9, 7, 5, 5, 5, 3, 8, 6, 11,
11, 2, 7, 7, 6, 5, 4, 6, 3, 8, 5, 8, 7, 6, 9, 4, 3, 7, 6, 6, 6, 5, 6, 10, 5, 9, 9,
8, 8, 7, 4, 8, 4, 9, 8, 5, 10, 10, 9, 7, 9, 7, 7, 10, 4, 7, 8, 4, 7, 8, 9, 11, 7, 9,
10, 10, 2, 7, 9, 4, 8, 8, 12, 9, 5, 11, 10, 7, 6, 4, 8, 9, 9, 6, 5, 6, 5, 6, 11, 7,
```

Note that this data, on its own, represents an approximation for the probability mass function. If you wanted to approximate $\mathrm{P}(Y = 3)$ you could simply count the number of times that "3" occurs in your data. This is an approximation based on the definition of probability. Here is the full histogram of the data, a count of times each value occurs:



A normalized histogram (where each value is divided by the length of your data list) is an approximation of the PMF. For a dataset of discrete numbers, a histogram shows the count of each value (in this case y). By the definition of probability, if you divide this count by the number of experiments run, you arrive at an approximation of the probability of the event $\mathrm{P}(Y = y)$. In our example, we have 10,000 elements in our dataset. The count of times that 3 occurs is 552. Note that:

$$\frac{\mathrm{count}(Y = 3)}{n} = \frac{552}{10000} = 0.0552$$
$$\mathrm{P}(Y = 3) = \frac{4}{36} = 0.0555$$

In this case, since we ran 10,000 trials, the histogram is a very good approximation of the PMF. We use the sum of dice as an example because it is easy to understand. Datasets in the real world often represent more exciting events.

Expectation

A random variable is fully represented by its probability mass function (PMF), which represents each of the values the random variable can take on, and the corresponding probabilities. A PMF can be a lot of information. Sometimes it is useful to summarize the random variable! The most common, and arguably the most useful, summary of a random variable is its "**Expectation**".

Definition: Expectation

The expectation of a random variable X, written $\mathrm{E}[X]$ is the average of all the values the random variable can take on, each weighted by the probability that the random variable will take on that value.

$$\mathrm{E}[X] = \sum_x x \cdot \mathrm{P}(X = x)$$

Expectation goes by many other names: Mean, Weighted Average, Center of Mass, 1st Moment. All of which are calculated using the same formula.

Recall that $\mathrm{P}(X = x)$, also written as $\mathrm{P}(x)$, is the probability mass function of the random variable X. Here is code that calculates the expectation of the sum of two dice, based off the probability mass function:

```
def expectation_sum_two_dice():
    exp_sum_two_dice = 0
    # sum of dice can take on the values 2 through 12
    for x in range(2, 13):
        pr_x = pmf_sum_two_dice(x) # pmf gives Pr(x)
        exp_sum_two_dice += x * pr_x
    return exp_sum_two_dice
```

If we worked it out manually we would get that if X is the sum of two dice, $\mathrm{E}[X] = 7$:

$$\mathrm{E}[X] = \sum_x x \cdot \mathrm{P}(X = x) = 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + \cdots + 12\frac{1}{36} = 7$$

7 is the "average" number you expect to get if you took the sum of two dice near infinite times. In this case it also happens to be the same as the mode, the most likely value of the sum of two dice, but this is not always the case!

Properties of expectation

Property: Linearity of Expectation

$$E[aX + b] = a\,\mathrm{E}[X] + b$$

Property: Expectation of the Sum of Random Variables

$$E[X + Y] = E[X] + E[Y]$$

Property: Law of Unconcious Statistician

$$E[g(X)] = \sum_x g(x)\,\mathrm{P}(X = x)$$

One can also calculate the expected value of a function g(X) of a random variable X when one knows the probability distribution of X but one does not explicitly know the distribution of g(X). This theorem has the humorous name of "the Law of the Unconscious Statistician" (LOTUS), because it is so useful that you should be able to employ it unconciously.

Property: Expectation of a Constant

$$E[a] = a$$

Sometimes in proofs, you will end up with the expectation of a constant (rather than a random variable). For example what does the $\mathrm{E}[5]$ mean? Since 5 is not a random variable, it does not change, and will always be 5, $\mathrm{E}[5] = 5$.

Variance

Definition: Variance of a Random Variable

The variance is a measure of the "spread" of a random variable around the mean. Variance for a random variable, X, with expected value $\mathrm{E}[X] = \mu$ is:

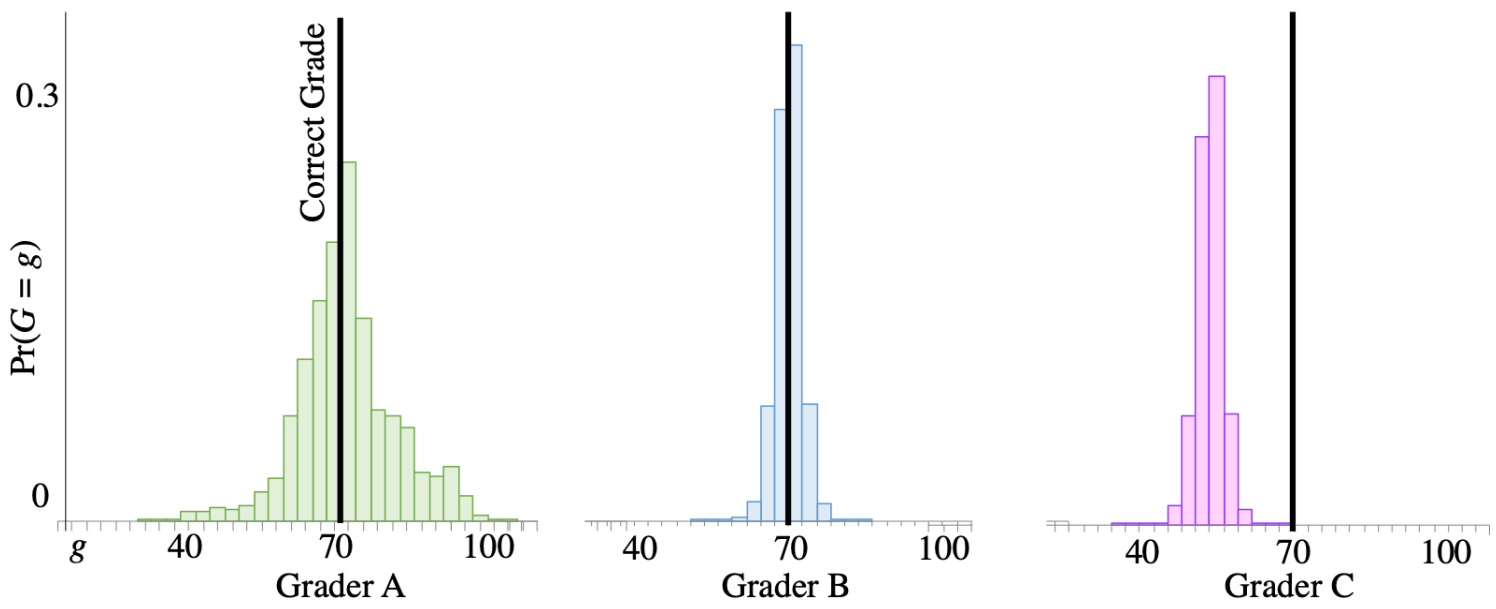
$$\mathrm{Var}(X) = \mathrm{E}[(X-\mu)^2]$$

Semantically, this is the average distance of a sample from the distribution to the mean. When computing the variance often we use a different (equivalent) form of the variance equation:

$$\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}[X]^2$$

In the last section we showed that Expectation was a useful summary of a random variable (it calculates the "weighted average" of the random variable). One of the next most important properties of random variables to understand is variance: the measure of spread.

To start, let's consider probability mass functions for three sets of graders. When each of them grades an assigment, meant to receive a 70/100, they each have a probability distribution of grades that they could give.



Distributions of three types of peer graders. Data is from a massive online course.

The distribution for graders in group C has a different *expectation*. The average grade that they give when grading an assignment worth 70 is a 55/100. That is clearly not great! But what is the difference between graders A and B? Both of them have the same expected value (which is equal to the correct grade). The graders in group A have a higher "spread". When grading an assignment worth 70, they have a reasonable chance of giving it a 100, or of giving it a 40. Graders in group B have much less spread. Most of the probability mass is close to 70. You want graders like those in group B: in expectation they give the correct grade, and they have low spread. As an aside: scores in group B came from a probabilistic algorithm over peer grades.

Theorists wanted a number to describe spread. They invented variance to be the average of the distance between values that the random variable could take on and the mean of the random variable. There are many reasonable choices for the distance function, probability theorists chose squared deviation from the mean:

$$\mathrm{Var}(X) = \mathrm{E}[(X-\mu)^2]$$

Proof: $\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}[X]^2$

It is much easier to compute variance using $\mathrm{E}[X^2] - \mathrm{E}[X]^2$. You certainly don't need to know why its an equivalent expression, but in case you were wondering, here is the proof.

$$
\begin{aligned}
\mathrm{Var}(X) &= \mathrm{E}[(X - \mu)^2] && \text{Note: } \mu = \mathrm{E}[X] \\
&= \sum_x (x - \mu)^2 \, \mathrm{P}(x) && \text{Definition of Expectation} \\
&= \sum_x (x^2 - 2\mu x + \mu^2) \, \mathrm{P}(x) && \text{Expanding the square} \\
&= \sum_x x^2 \, \mathrm{P}(x) - 2\mu \sum_x x \, \mathrm{P}(x) + \mu^2 \sum_x \mathrm{P}(x) && \text{Propagating the sum} \\
&= \sum_x x^2 \, \mathrm{P}(x) - 2\mu \, \mathrm{E}[X] + \mu^2 \sum_x \mathrm{P}(x) && \text{Substitute def of expectation} \\
&= \mathrm{E}[X^2] - 2\mu \, \mathrm{E}[X] + \mu^2 \sum_x \mathrm{P}(x) && \text{LOTUS } g(x) = x^2 \\
&= \mathrm{E}[X^2] - 2\mu \, \mathrm{E}[X] + \mu^2 && \text{Since } \sum_x \mathrm{P}(x) = 1 \\
&= \mathrm{E}[X^2] - 2 \, \mathrm{E}[X]^2 + \mathrm{E}[X]^2 && \text{Since } \mu = \mathrm{E}[X] \\
&= \mathrm{E}[X^2] - \mathrm{E}[X]^2 && \text{Cancelation}
\end{aligned}
$$

Standard Deviation

Variance is especially useful for comparing the "spread" of two distributions and it has the useful property that it is easy to calculate. In general a larger variance means that there is more deviation around the mean — more spread. However, if you look at the leading example, the units of variance are the square of points. This makes it hard to interpret the numerical value. What does it mean that the spread is 52 points^2? A more interpretable measure of spread is the square root of Variance, which we call the Standard Deviation $\mathrm{Std}(X) = \sqrt{\mathrm{Var}(X)}$. The standard deviation of our grader is 7.2 points. In this example folks find it easier to think of spread in points rather than points^2. As an aside, the standard deviation is the average distance of a sample (from the distribution) to the mean, using the euclidean distance function.

Bernoulli Distribution

Parametric Random Variables

There are many classic and commonly-seen random variable abstractions that show up in the world of probability. At this point in the class, you will learn about several of the most significant parametric discrete distributions. When solving problems, if you can recognize that a random variable fits one of these formats, then you can use its pre-derived probability mass function (PMF), expectation, variance, and other properties. Random variables of this sort are called **parametric** random variables. If you can argue that a random variable falls under one of the studied parametric types, you simply need to provide parameters. A good analogy is a `class` in programming. Creating a parametric random variable is very similar to calling a constructor with input parameters.

Bernoulli Random Variables

A Bernoulli random variable (also called a *boolean* or *indicator* random variable) is the simplest kind of parametric random variable. It can take on two values, 1 and 0. It takes on a 1 if an experiment with probability p resulted in success and a 0 otherwise. Some example uses include a coin flip, a random binary digit, whether a disk drive crashed, and whether someone likes a Netflix movie. Here p is the parameter, but different instances of Bernoulli random variables might have different values of p.

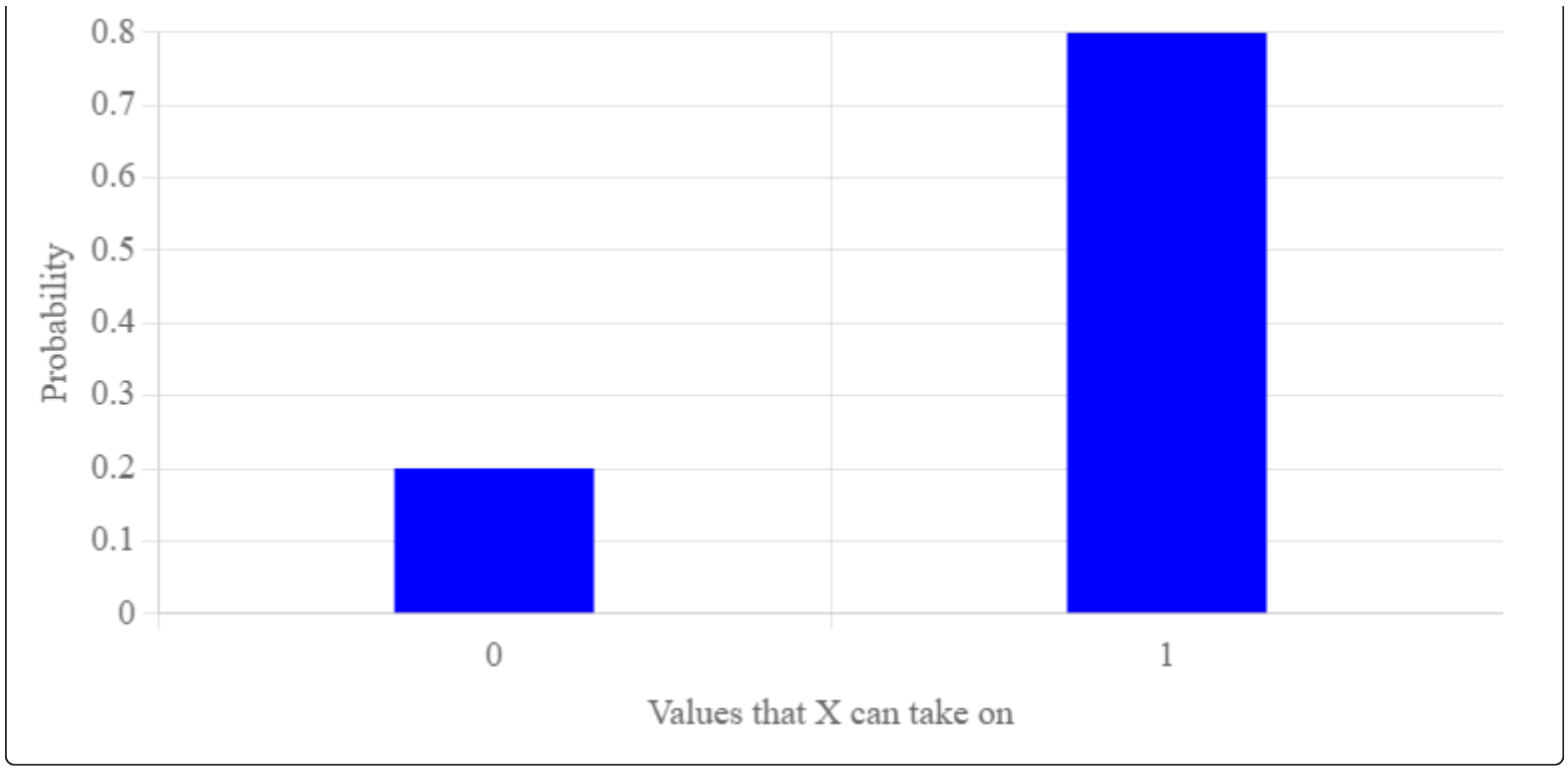
Here is a full description of the key properties of a Bernoulli random variable. If X is declared to be a Bernoulli random variable with parameter p, denoted $X \sim \text{Bern}(p)$:

Bernoulli Random Variable

Notation:	$X \sim \text{Bern}(p)$
Description:	A boolean variable that is 1 with probability p
Parameters:	p, the probability that $X = 1$.
Support:	x is either 0 or 1
PMF equation:	$\text{P}(X = x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 0 \end{cases}$
PMF (smooth):	$\text{P}(X = x) = p^x (1-p)^{1-x}$
Expectation:	$\text{E}[X] = p$
Variance:	$\text{Var}(X) = p(1-p)$

PMF graph:

Parameter p: 0.80



Because Bernoulli distributed random variables are parametric, as soon as you declare a random variable to be of type Bernoulli you automatically can know all of these pre-derived properties! Some of these properties are straightforward to prove for a Bernoulli. For example, you could have solved for expectation:

Proof: Expectation of a Bernoulli. If X is a Bernoulli with parameter p, $X \sim \text{Bern}(p)$:

$$\begin{aligned} \text{E}[X] &= \sum_x x \cdot \text{P}(X = x) && \text{Definition of expectation} \\ &= 1 \cdot p + 0 \cdot (1-p) && X \text{ can take on values 0 and 1} \\ &= p && \text{Remove the 0 term} \end{aligned}$$

Proof: Variance of a Bernoulli. If X is a Bernoulli with parameter p, $X \sim \text{Bern}(p)$:

To compute variance, first compute $E[X^2]$:

$$\begin{aligned} E[X^2] &= \sum_x x^2 \cdot \text{P}(X = x) && \text{LOTUS} \\ &= 0^2 \cdot (1-p) + 1^2 \cdot p \\ &= p \end{aligned}$$

$$\begin{aligned} \text{Var}(X) &= E[X^2] - E[X]^2 && \text{Def of variance} \\ &= p - p^2 && \text{Substitute } E[X^2] = p, E[X] = p \\ &= p(1-p) && \text{Factor out } p \end{aligned}$$

Indicator Random Variable

Definition: Indivator Random Variable

An indicator variable is a Bernoulli random variable which takes on the value 1 if an underlying event occurs, and 0 otherwise.

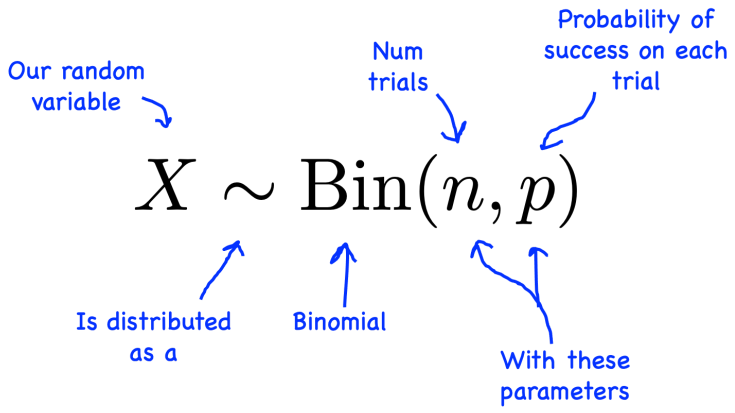
Indicator random variables are a convenient way to convert the "true/false" outcome of an event into a number. That number may be easier to incoporate into an equation. See the binomial expectation derivation for an example.

A random variable I is an indicator variable for an event A if $I = 1$ when A occurs and $I = 0$ if A does not occur. Indicator random variables are Bernoulli random variables, with $p = \text{P}(A)$. I is a common choice of name for an indicator random variable.

Here are some properties of indicator random variables: $\mathrm{P}(I = 1) = \mathrm{P}(A)$ and $\mathrm{E}[I] = \mathrm{P}(A)$.

Binomial Distribution

In this section, we will discuss the binomial distribution. To start, imagine the following example. Consider n independent trials of an experiment where each trial is a "success" with probability p. Let X be the number of successes in n trials. This situation is truly common in the natural world, and as such, there has been a lot of research into such phenomena. Random variables like X are called binomial random variables. If you can identify that a process fits this description, you can inherit many already proved properties such as the PMF formula, expectation, and variance!



Here are a few examples of binomial random variables:

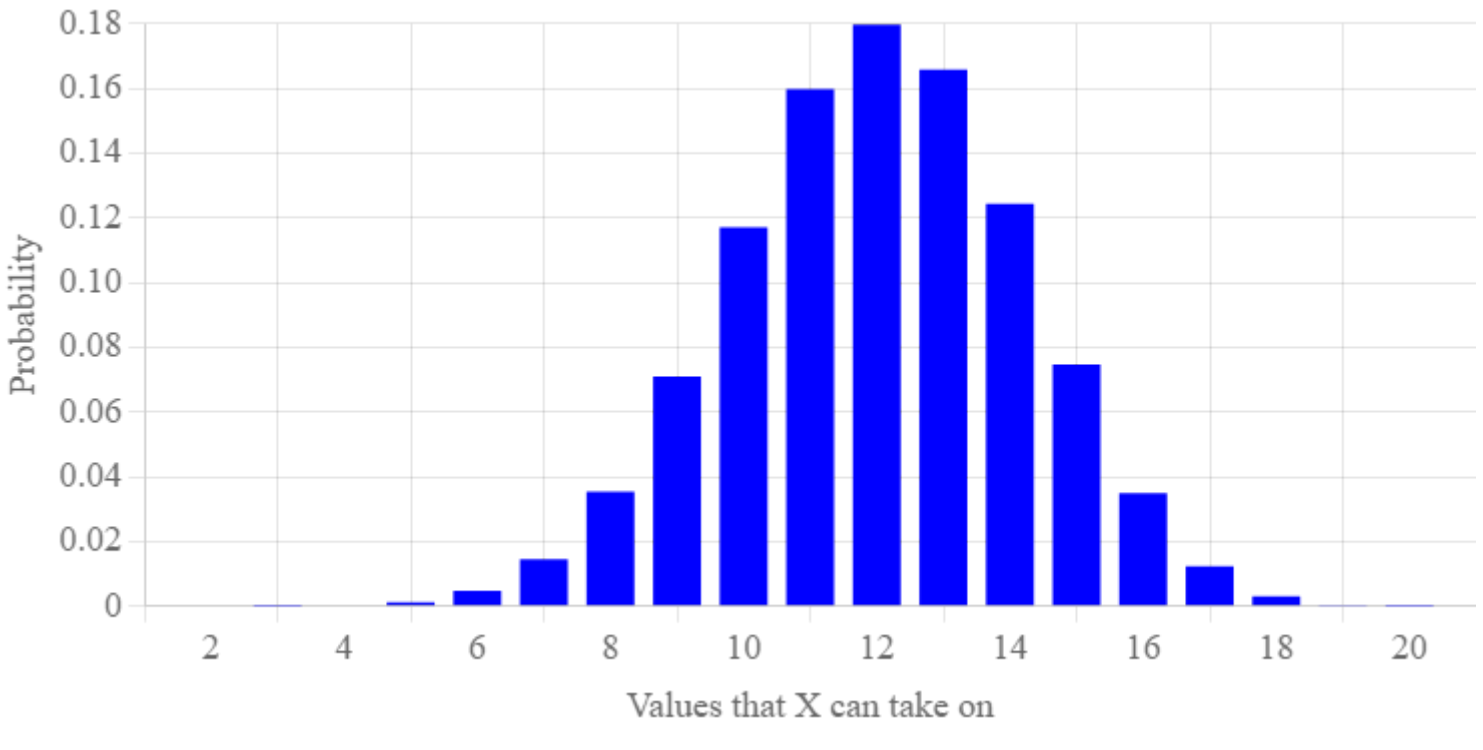
- # of heads in n coin flips
- # of 1's in randomly generated length n bit string
- # of disk drives crashed in 1000 computer cluster, assuming disks crash independently

Binomial Random Variable

Notation:	$X \sim \text{Bin}(n, p)$
Description:	Number of "successes" in n identical, independent experiments each with probability of success p.
Parameters:	$n \in \{0, 1, \ldots\}$, the number of experiments. $p \in [0, 1]$, the probability that a single experiment gives a "success".
Support:	$x \in \{0, 1, \ldots, n\}$
PMF equation:	$\text{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$
Expectation:	$\text{E}[X] = n \cdot p$
Variance:	$\text{Var}(X) = n \cdot p \cdot (1-p)$

PMF graph:

Parameter n: 20 Parameter p: 0.60



One way to think of the binomial is as the sum of n Bernoulli variables. Say that $Y_i \sim \text{Bern}(p)$ is an indicator Bernoulli random variable which is 1 if experiment i is a success. Then if X is the total number of successes in n experiments, $X \sim \text{Bin}(n, p)$:

$$X = \sum_{i=1}^{n} Y_i$$

Recall that the outcome of Y_i will be 1 or 0, so one way to think of X is as the sum of those 1s and 0s.

Binomial PMF

The most important property to know about a binomial is its PMF function:

Probability Mass Function for a Binomial

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Probability that our variable takes on the value k

Recall, we derived this formula in Part 1. There is a complete example on the probability of k heads in n coin flips, where each flip is heads with probability 0.5: Many Coin Flips. To briefly review, if you think of each experiment as being distinct, then there are $\binom{n}{k}$ ways of permuting k successes from n experiments. For any of the mutually exclusive permutations, the probability of that permutation is $p^k \cdot (1-p)^{n-k}$.

The name binomial comes from the term $\binom{n}{k}$ which is formally called the binomial coefficient.

Expectation of Binomial

There is an easy way to calculate the expectation of a binomial and a hard way. The easy way is to leverage the fact that a binomial is the sum of Bernoulli indicator random variables. $X = \sum_{i=1}^{n} Y_i$ where Y_1 is an indicator of whether the ith experiment was a success: $Y_i \sim \text{Bern}(p)$. Since the expectation of the sum of random variables is the sum of expectations, we can add the expectation, $\text{E}[Y_i] = p$, of each of the Bernoulli's:

$$\begin{aligned}
\text{E}[X] &= \text{E}\left[\sum_{i=1}^{n} Y_i\right] && \text{Since } X = \sum_{i=1}^{n} Y_i \\
&= \sum_{i=1}^{n} \text{E}[Y_i] && \text{Expectation of sum} \\
&= \sum_{i=1}^{n} p && \text{Expectation of Bernoulli} \\
&= n \cdot p && \text{Sum } n \text{ times}
\end{aligned}$$

The hard way is to use the definition of expectation:

$$\begin{aligned}
\text{E}[X] &= \sum_{i=0}^{n} i \cdot \text{P}(X = i) && \text{Def of expectation} \\
&= \sum_{i=0}^{n} i \cdot \binom{n}{i} p^i (1-p)^{n-i} && \text{Sub in PMF} \\
&\cdots && \text{Many steps later} \\
&= n \cdot p
\end{aligned}$$

Binomial Distribution in Python

As you might expect, you can use binomial distributions in code. The standardized library for binomials is `scipy.stats.binom`.

One of the most helpful methods that this package provides is a way to calculate the PMF. For example, say $X \sim \text{Bin}(n = 5, p = 0.6)$ and you want to find $\text{P}(X = 2)$ you could use the following code:

```
from scipy import stats

# define variables for x, n, and p
n = 5
p = 0.6
x = 2

# use scipy to compute the pmf
p_x = stats.binom.pmf(x, n, p)

# use the probability for future work
print(f'P(X = {x}) = {p_x}')
```

Console:

```
P(X = 2) = 0.2304
```

Another particularly helpful function is the ability to generate a random sample from a binomial. For example, say $X \sim \text{Bin}(n = 10, p = 0.3)$ represents the number of requests to a website. We can draw 100 samples from this distribution using the following code:

```
from scipy import stats

# define variables for x, n, and p
n = 10
p = 0.3
x = 2

# use scipy to compute the pmf
samples = stats.binom.rvs(n, p, size=100)

# use the probability for future work
print(samples)
```

Console:

```
[4 5 3 1 4 5 3 1 4 6 5 6 1 2 1 1 2 3 2 5 2 2 2 4 4 2 2 3 6 3 1 1 4 2 6 2 4
 2 3 3 4 2 4 2 4 5 0 1 4 3 4 3 3 1 3 1 1 2 2 2 2 3 5 3 3 3 2 1 3 2 1 2 3 3
 4 5 1 3 7 1 4 1 3 3 4 4 1 2 4 4 0 2 4 3 2 3 3 1 1 4]
```

You might be wondering what a random sample is! A random sample is a randomly chosen assignment for our random variable. Above we have 100 such assignments. The probability that value x is chosen is given by the PMF: $\text{P}(X = x)$. You will notice that even though 8 is a possible assignment to the binomial above, in 100 samples we never saw the value 8. Why? Because $P(X = 8) \approx 0.0014$. You would need to draw 1,000 samples before you would expect to see an 8.

There are also functions for getting the mean, the variance, and more. You can read the scipy.stats.binom documentation, especially the list of methods.

Poisson Distribution

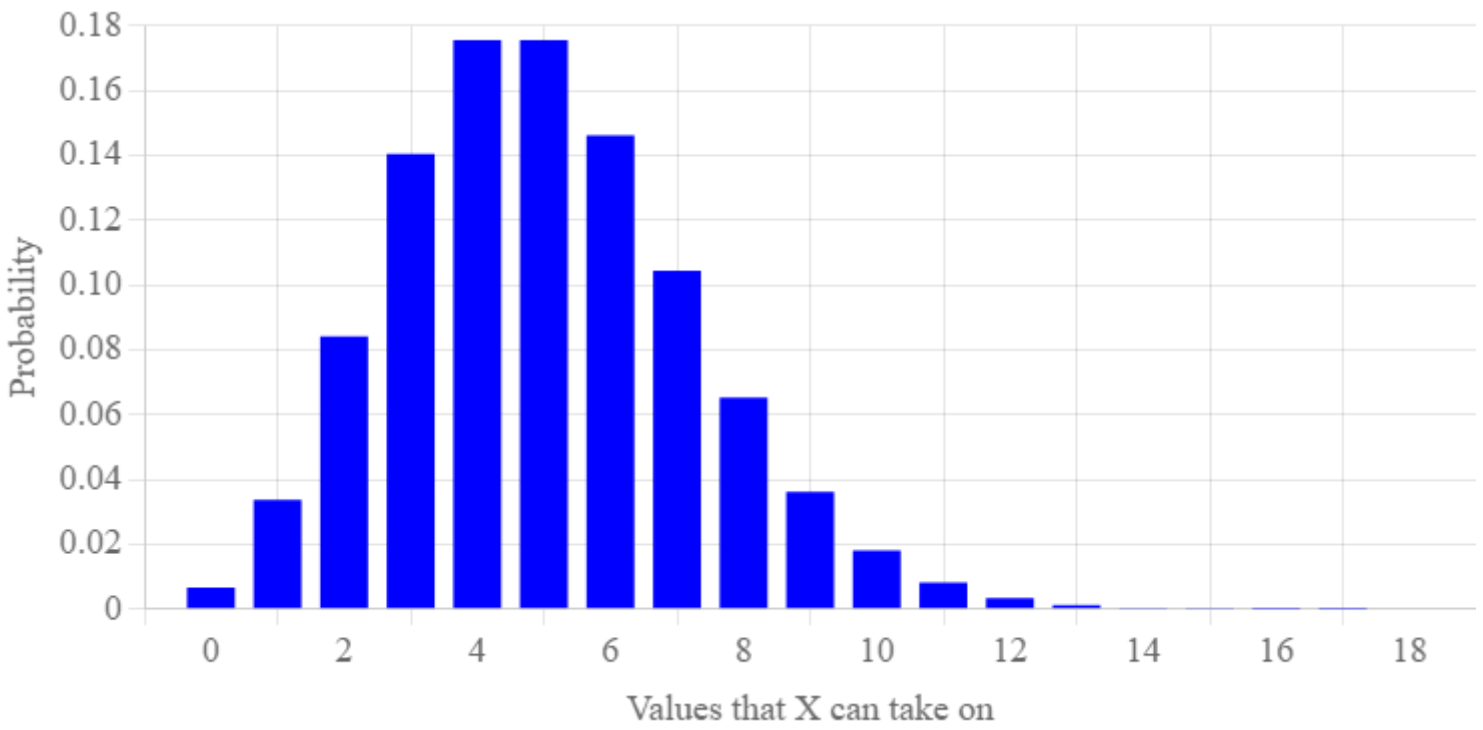
A Poisson random variable gives the probability of a given number of events in a fixed interval of time (or space). It makes the Poisson assumption that events occur with a known constant mean rate and independently of the time since the last event.

Poisson Random Variable

Notation:	$X \sim \text{Poi}(\lambda)$
Description:	Number of events in a fixed time frame if (a) the events occur with a constant mean rate and (b) they occur independently of time since last event.
Parameters:	$\lambda \in \{0, 1, \ldots\}$, the constant average rate.
Support:	$x \in \{0, 1, \ldots\}$
PMF equation:	$\text{P}(X = x) = \dfrac{\lambda^x e^{-\lambda}}{x!}$
Expectation:	$\text{E}[X] = \lambda$
Variance:	$\text{Var}(X) = \lambda$

PMF graph:

Parameter λ: 5

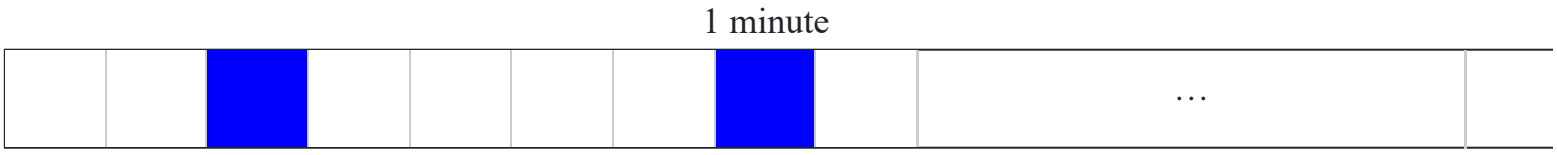


Poisson Intuition

In this section we show the intuition behind the Poisson derivation. It is both a great way to deeply understand the Poisson, as well as good practice with Binomial distributions.

Let's work on the problem of predicting the chance of a given number of events occurring in a fixed time interval — the next minute. For example, imagine you are working on a ride sharing application and you care about the probability of how many requests you get from a particular area. From historical data, you know that the average requests per minute is $\lambda = 5$. What is the probability of getting 1, 2, 3, etc requests in a minute?

💡: We could approximate a solution to this problem by using a binomial distribution! Lets say we split our minute into 60 seconds, and make each second an indicator Bernoulli variable — you either get a request or you don't. If you get a request in a second, the indicator is 1. Otherwise it is 0. Here is a visualization of our 60 binary-indicators. In this example imagine we have requests at 2.75 and 7.12 seconds. the corresponding indicator variables are **blue** filled in boxes:



The total number of requests received over the minute can be approximated as the sum of the sixty indicator variables, which conveniently matches the description of a binomial — a sum of Bernoullis. Specifically define X to be the number of requests in a minute. X is a binomial with $n = 60$ trials. What is the probability, p, of a success on a single trial? To make the expectation of X equal the observed historical average $\lambda = 5$ we should choose p so that $\lambda = \mathrm{E}[X]$.

$$\begin{aligned} \lambda &= \mathrm{E}[X] && \text{Expectation matches historical average} \\ \lambda &= n \cdot p && \text{Expectation of a Binomial is } n \cdot p \\ p &= \frac{\lambda}{n} && \text{Solving for } p \end{aligned}$$

In this case since $\lambda = 5$ and $n = 60$, we should choose $p = 5/60$ and state that $X \sim \mathrm{Bin}(n = 60, p = 5/60)$. Now that we have a form for X we can answer probability questions about the number of requests by using the Binomial PMF:

$$\mathrm{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

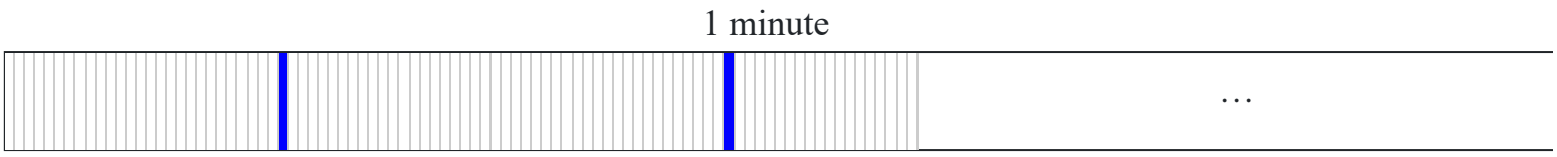
So for example:

$$\mathrm{P}(X = 1) = \binom{60}{1} (5/60)^1 (55/60)^{60-1} \approx 0.0295$$

$$\mathrm{P}(X = 2) = \binom{60}{2} (5/60)^2 (55/60)^{60-2} \approx 0.0790$$

$$\mathrm{P}(X = 3) = \binom{60}{3} (5/60)^3 (55/60)^{60-3} \approx 0.1389$$

Great! But don't forget that this was an approximation. We didn't account for the fact that there can be more than one event in a single second. One way to assuage this issue is to divide our minute into more fine-grained intervals (the choice to split it into 60 seconds was rather arbitrary). Instead lets divide our minute into 600 deciseconds, again with requests at 2.75 and 7.12 seconds:



Now $n = 600$, $p = 5/600$ and $X \sim \mathrm{Bin}(n = 600, p = 6/600)$. We can repeat our example calculations using this better approximation:

$$\mathrm{P}(X = 1) = \binom{600}{1} (5/600)^1 (595/60)^{600-1} \approx 0.0333$$

$$\mathrm{P}(X = 2) = \binom{600}{2} (5/600)^2 (595/600)^{600-2} \approx 0.0837$$

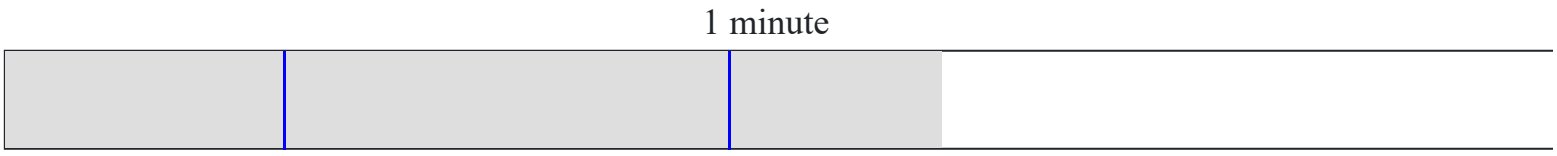
$$\mathrm{P}(X = 3) = \binom{600}{3} (5/600)^3 (595/600)^{600-3} \approx 0.1402$$

Choose any value of n, the number of buckets to divide our minute into: 0

The larger n is, the more accurate the approximation. So what happens when n is infinity? It becomes a Poisson!

Poisson, a Binomial in the limit

Or if we really cared about making sure that we don't get two events in the same bucket, we can divide our minute into infinitely small buckets:



Proof: Derivation of the Poisson

What does the PMF of X look like now that we have infinite divisions of our minute? We can write the equation and think about it as n goes to infinity. Recall that p still equals λ/n:

$$\mathrm{P}(X = x) = \lim_{n\to\infty} \binom{n}{x} (\lambda/n)^x (1-\lambda/n)^{n-x}$$

While it may look intimidating, this expression simplifies nicely. This proof uses a few special limit rules that we haven't introduced in this book:

$$\begin{aligned}
\mathrm{P}(X = x) &= \lim_{n\to\infty} \binom{n}{x} (\lambda/n)^x (1-\lambda/n)^{n-x} && \text{Start: binomial in the limit} \\
&= \lim_{n\to\infty} \binom{n}{x} \cdot \frac{\lambda^x}{n^x} \cdot \frac{(1-\lambda/n)^n}{(1-\lambda/n)^x} && \text{Expanding the power terms} \\
&= \lim_{n\to\infty} \frac{n!}{(n-x)!x!} \cdot \frac{\lambda^x}{n^x} \cdot \frac{(1-\lambda/n)^n}{(1-\lambda/n)^x} && \text{Expanding the binomial term} \\
&= \lim_{n\to\infty} \frac{n!}{(n-x)!x!} \cdot \frac{\lambda^x}{n^x} \cdot \frac{e^{-\lambda}}{(1-\lambda/n)^x} && \text{Rule } \lim_{n\to\infty}(1-\lambda/n)^n = e^{-\lambda} \\
&= \lim_{n\to\infty} \frac{n!}{(n-x)!x!} \cdot \frac{\lambda^x}{n^x} \cdot \frac{e^{-\lambda}}{1} && \text{Rule } \lim_{n\to\infty} \lambda/n = 0 \\
&= \lim_{n\to\infty} \frac{n!}{(n-x)!} \cdot \frac{1}{x!} \cdot \frac{\lambda^x}{n^x} \cdot \frac{e^{-\lambda}}{1} && \text{Splitting first term} \\
&= \lim_{n\to\infty} \frac{n^x}{1} \cdot \frac{1}{x!} \cdot \frac{\lambda^x}{n^x} \cdot \frac{e^{-\lambda}}{1} && \lim_{n\to\infty} \frac{n!}{(n-x)!} = n^x \\
&= \lim_{n\to\infty} \frac{\lambda^x}{x!} \cdot \frac{e^{-\lambda}}{1} && \text{Cancel } n^x \\
&= \frac{\lambda^x \cdot e^{-\lambda}}{x!} && \text{Simplify}
\end{aligned}$$

That is a beautiful expression! Now we can calculate the real probability of number of requests in a minute, if the historical average is $\lambda = 5$:

$$\mathrm{P}(X = 1) = \frac{5^1 \cdot e^{-5}}{1!} = 0.03369$$

$$\mathrm{P}(X = 2) = \frac{5^2 \cdot e^{-5}}{2!} = 0.08422$$

$$\mathrm{P}(X = 3) = \frac{5^3 \cdot e^{-5}}{3!} = 0.14037$$

This is both more accurate and much easier to compute!

Changing time frames

Say you are given a rate over one unit of time, but you want to know the rate in another unit of time. For example, you may be given the rate of hits to a website per minute, but you want to know the probability over a 20 minute period. You would just need to multiply this rate by 20 in order to go from the "per 1 minute of time" rate to obtain the "per 20 minutes of time" rate.

Continuous Distribution

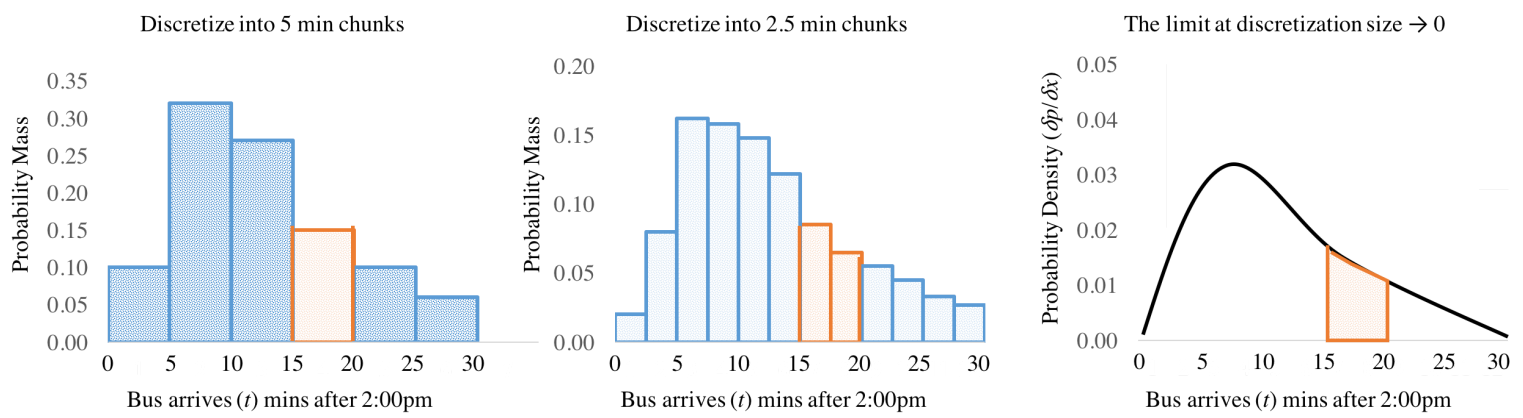
So far, all random variables we have seen have been *discrete*. In all the cases we have seen in CS109 this meant that our RVs could only take on integer values. Now it's time for *continuous* random variables which can take on values in the real number domain ($\mathbb{R}$). Continuous random variables can be used to represent measurements with arbitrary precision (eg height, weight, time).

From Discrete to Continuous

To make our transition from thinking about discrete random variables, to thinking about continuous random variables, lets start with a thought experiment: Imagine you are running to catch the bus. You know that you will arrive at 2:15pm but you don't know exactly when the bus will arrive, and want to think of the arrival time in minutes past 2pm as a random variable T so that you can calculate the probability that you will have to wait more than five minutes $P(15 < T < 20)$.

We immediately face a problem. For discrete distributions we would describe the probability that a random variable takes on exact values. This doesn't make sense for continuous values, like the time the bus arrives. As an example, what is the probability that the bus arrives at exactly 2:17pm and 12.12333911102389234 seconds? Similarly, if I were to ask you: what is the probability of a child being born with weight *exactly* equal to 3.523112342234 kilos, you might recognize that question as ridiculous. No child will have precisely that weight. Real values can have infinite precision and as such it is a bit mind boggling to think about the probability that a random variable takes on a specific value.

Instead, let's start by discretizing time, our continuous variable, by breaking it into 5 minute chunks. We can now think about something like, the probability that the bus arrives between 2:00p and 2:05 as an event with some probability (see figure below on the left). Five minute chunks seem a bit coarse. You could imagine that instead, we could have discretized time into 2.5 minute chunks (figure in the center). In this case the probability that the bus shows up between 15 mins and 20 mins after 2pm is the sum of two chunks, shown in orange. Why stop there? In the limit we could keep breaking time down into smaller and smaller pieces. Eventually we will be left with a derivative of probability at each moment of time, where the probability that $P(15 < T < 20)$ is the integral of that derivative between 15 and 20 (figure on the right).



Probability Density Functions

In the world of discrete random variables, the most important property of a random variable was its probability mass function (PMF) that would tell you the probability of the random variable taking on any value. When we move to the world of continuous random variables, we are going to need to rethink this basic concept. In the continuous world, every random variable instead has a Probability *Density* Function (PDF) which defines the relative likelihood that a random variable takes on a particular value. We traditionally use the symbol f for the probability density function and write it in one of two ways:

$$f(X = x) \quad \text{or} \quad f(x)$$

Where the notation on the right hand side is shorthand and the lowercase x implies that we are talking about the relative likelihood of a continuous random variable which is the upper case X. Like in the bus example, the PDF is the derivative of probability at all points of the random variable. This means that the PDF has the important property that you can integrate over it to find the probability that the random variable takes on values within a range (a, b).

Definition: Continuous Random Variable

X is a Continuous Random Variable if there is a Probability Density Function (PDF) $f(x)$ that takes in real valued numbers x such that:

$$\mathrm{P}(a \leq X \leq b) = \int_a^b f(x)\,dx$$

The following properties must also hold. These preserve the axiom that $P(a \leq X \leq b)$ is a probability:

$$0 \leq \mathrm{P}(a \leq X \leq b) \leq 1$$
$$\mathrm{P}(-\infty < X < \infty) = 1$$

A common misconception is to think of $f(x)$ as a probability. It is instead what we call a probability density. It represents probability/unit of X. Generally this is only meaningful when we either take an integral over the PDF **or** we compare probability densities. As we mentioned when motivating probability densities, the probability that a continuous random variable takes on a specific value (to infinite precision) is 0.

$$\mathrm{P}(X = a) = \int_a^a f(x)\,dx = 0$$

That is pretty different than in the discrete world where we often talked about the probability of a random variable taking on a particular value.

Cumulative Distribution Function

Having a probability density is great, but it means we are going to have to solve an integral every single time we want to calculate a probability. To avoid this unfortunate fate, we are going to use a standard called a cumulative distribution function (CDF). The CDF is a function which takes in a number and returns the probability that a random variable takes on a value less than that number. It has the pleasant property that, if we have a CDF for a random variable, we don't need to integrate to answer probability questions!

For a continuous random variable X the Cumulative Distribution Function, written $F(x)$ is:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)\,dx$$

Why is the CDF the probability that a random variable takes on a value **less than** the input value as opposed to greater than? It is a matter of convention. But it is a useful convention. Most probability questions can be solved simply by knowing the CDF (and taking advantage of the fact that the integral over the range $-\infty$ to ∞ is 1. Here are a few examples of how you can answer probability questions by just using a CDF:

Probability Query	Solution	Explanation
$\mathrm{P}(X < a)$	$F(a)$	That is the definition of the CDF
$\mathrm{P}(X \leq a)$	$F(a)$	Trick question. $\mathrm{P}(X = a) = 0$
$\mathrm{P}(X > a)$	$1 - F(a)$	$\mathrm{P}(X < a) + \mathrm{P}(X > a) = 1$
$\mathrm{P}(a < X < b)$	$F(b) - F(a)$	$F(a) + \mathrm{P}(a < X < b) = F(b)$

The continuous distribution also exists for discrete random variables, but there is less utility to a CDF in the discrete world as none of our discrete random variables had "closed form" (eg without any summation) functions for the CDF:

$$F_X(a) = \sum_{i=1}^{a} P(X = i)$$

Solving for Constants

Let X be a continuous random variable with PDF:

$$f(x) = \begin{cases} C(4x - 2x^2) & \text{when } 0 < x < 2 \\ 0 & \text{otherwise} \end{cases}$$

In this function, C is a constant. What value is C? Since we know that the PDF must sum to 1:

$$\int_0^2 C(4x - 2x^2)\,dx = 1$$
$$C\left(2x^2 - \frac{2x^3}{3}\right)\Big|_0^2 = 1$$
$$C\left(\left(8 - \frac{16}{3}\right) - 0\right) = 1$$
$$C = 3/8$$

Now that we know C, what is $\mathrm{P}(X > 1)$?

$$\begin{aligned} \mathrm{P}(X > 1) &= \int_1^\infty f(x)\,dx \\ &= \int_1^2 \frac{3}{8}(4x - 2x^2)\,dx \\ &= \frac{3}{8}\left(2x^2 - \frac{2x^3}{3}\right)\Big|_1^2 \\ &= \frac{3}{8}\left[\left(8 - \frac{16}{3}\right) - \left(2 - \frac{2}{3}\right)\right] = \frac{1}{2} \end{aligned}$$

Expectation and Variance of Continuous Variables

For continuous RV X:

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx$$
$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) dx$$
$$E[X^n] = \int_{-\infty}^{\infty} x^n f(x) dx$$

For both continuous and discrete RVs:

$$E[aX + b] = aE[X] + b$$
$$\mathrm{Var}(X) = E[(X - \mu)^2] = E[X^2] - (E[X])^2$$
$$\mathrm{Var}(aX + b) = a^2 \mathrm{Var}(X)$$

Uniform Distribution

The most basic of all the continuous random variables is the uniform random variable, which is equally likely to take on any value in its range (α, β). X is a *uniform random variable* ($X \sim \text{Uni}(\alpha, \beta)$) if it has PDF:

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{when } \alpha \leq x \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

Notice how the density $1/(\beta - \alpha)$ is exactly the same regardless of the value for x. That makes the density uniform. So why is the PDF $1/(\beta - \alpha)$ and not 1? That is the constant that makes it such that the integral over all possible inputs evaluates to 1.

Uniform Random Variable

Notation: $X \sim \text{Uni}(\alpha, \beta)$

Description: A continuous random variable that takes on values, with equal likelihood, between α and β

Parameters: $\alpha \in \mathbb{R}$, the minimum value of the variable.
$\beta \in \mathbb{R}$, $\beta > \alpha$, the maximum value of the variable.

Support: $x \in [\alpha, \beta]$

PDF equation: $f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{else} \end{cases}$

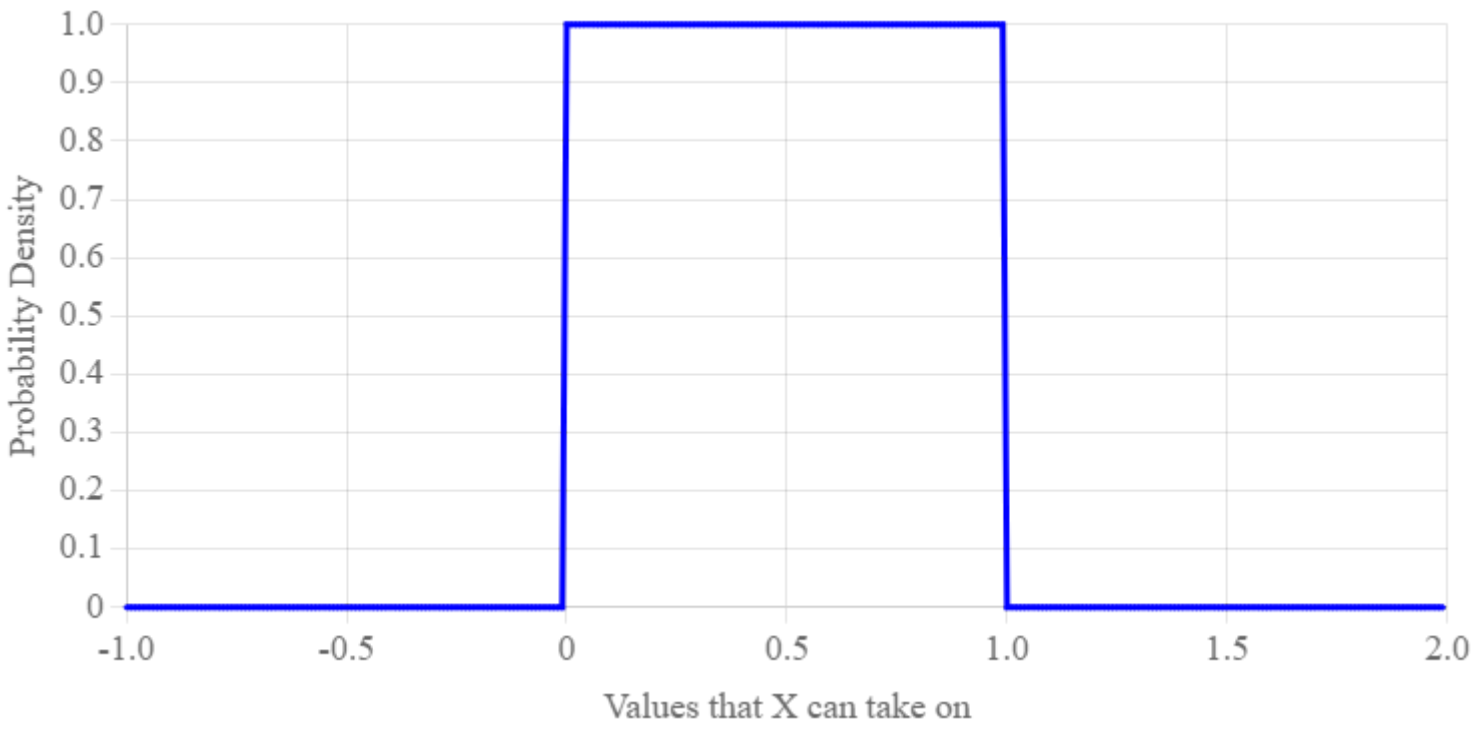
CDF equation: $F(x) = \begin{cases} \frac{x - \alpha}{\beta - \alpha} & \text{for } x \in [\alpha, \beta] \\ 0 & \text{for } x < \alpha \\ 1 & \text{for } x > \beta \end{cases}$

Expectation: $\text{E}[X] = \frac{1}{2}(\alpha + \beta)$

Variance: $\text{Var}(X) = \frac{1}{12}(\beta - \alpha)^2$

PDF graph:

Parameter α: 0 Parameter β: 1



Example: You are running to the bus stop. You don't know exactly when the bus arrives. You believe all times between 2 and 2:30 are equally likely. You show up at 2:15pm. What is P(wait < 5 minutes)?

Let T be the time, in minutes after 2pm that the bus arrives. Because we think that all times are equally likely in this range, $T \sim \text{Uni}(\alpha = 0, \beta = 30)$. The probability that you wait 5 minutes is equal to the probability that the bus shows up between 2:15 and 2:20. In other words $\text{P}(15 < T < 20)$:

$$
\begin{aligned}
\mathrm{P}(\text{Wait under 5 mins}) &= \mathrm{P}(15 < T < 20) \\
&= \int_{15}^{20} f_T(x) \partial x \\
&= \int_{15}^{20} \frac{1}{\beta - \alpha} \partial x \\
&= \frac{1}{30} \partial x \\
&= \frac{x}{30} \Big|_{15}^{20} \\
&= \frac{20}{30} - \frac{15}{30} = \frac{5}{30}
\end{aligned}
$$

We can come up with a closed form for the probability that a uniform random variable X is in the range a to b, assuming that $\alpha \leq a \leq b \leq \beta$:

$$
\begin{aligned}
\mathrm{P}(a \leq X \leq b) &= \int_a^b f(x)\, dx \\
&= \int_a^b \frac{1}{\beta - \alpha}\, dx \\
&= \frac{b - a}{\beta - \alpha}
\end{aligned}
$$

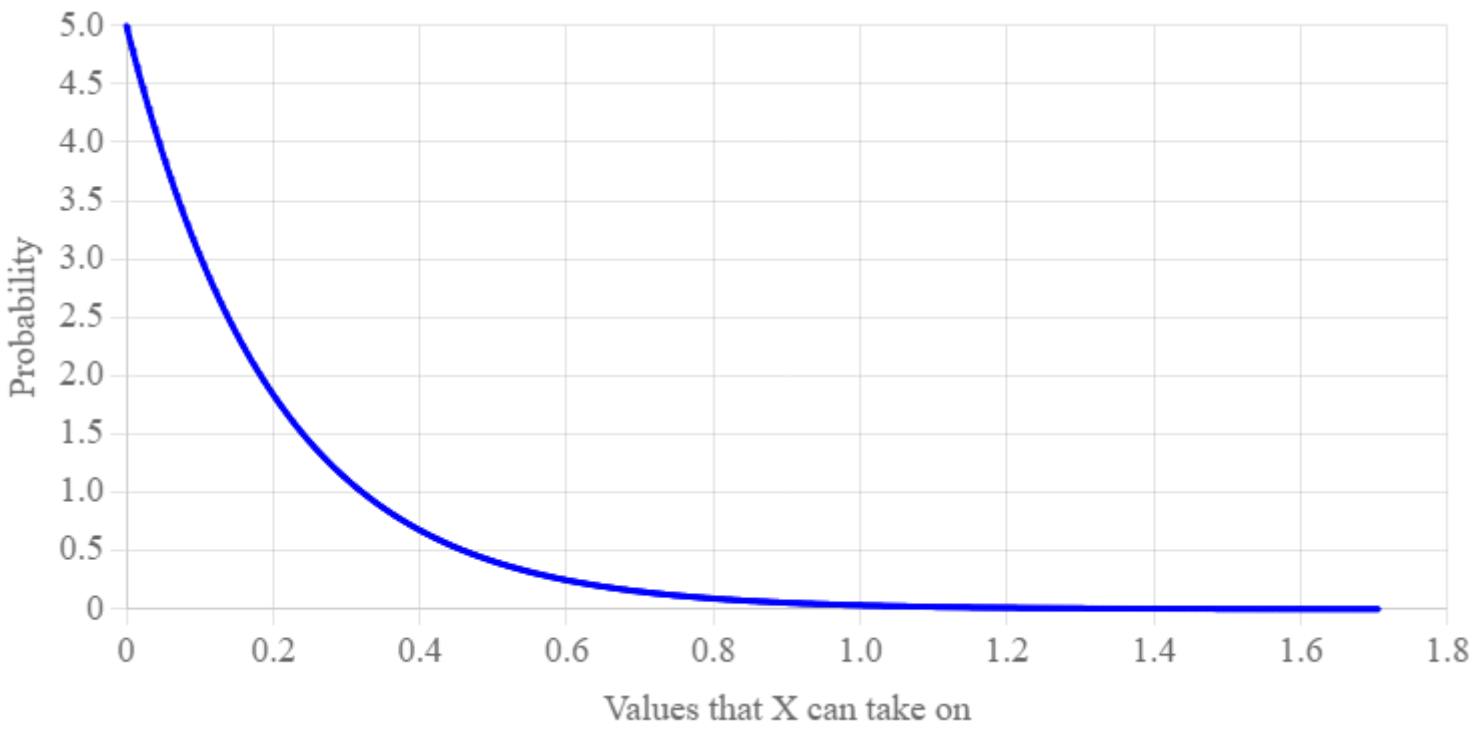
Exponential Distribution

An exponential distribution measures the **amount** of time until a next event occurs. It assumes that the events occur via a poisson process. Note that this is different from the Poisson Random Variable which measures number of events in a fixed amount of time.

Exponential Random Variable

Notation:	$X \sim \text{Exp}(\lambda)$
Description:	Time until next events if (a) the events occur with a constant mean rate and (b) they occur independently of time since last event.
Parameters:	$\lambda \in \{0, 1, \ldots\}$, the constant average rate.
Support:	$x \in \mathbb{R}^+$
PDF equation:	$f(x) = \lambda e^{-\lambda x}$
CDF equation:	$F(x) = 1 - e^{-\lambda x}$
Expectation:	$\text{E}[X] = 1/\lambda$
Variance:	$\text{Var}(X) = 1/\lambda^2$
PDF graph:	

Parameter λ: 5



An exponential distribution is a great example of a continuous distribution where the cumulative distribution function (CDF) is much easier to work with as it allows you to answer probability questions without using integrals.

Example: Based on historical data from the USGS, earthquakes of magnitude 8.0+ happen in a certain location at a rate of 0.002 per year. Earthquakes are known to occur via a poisson process. What is the probability of a major earthquake in the next 4 years?

Let Y be the years until the next major earthquake. Because Y measures time *until* the next event it fits the description of an exponential random variable: $Y \sim \text{Exp}(\lambda = 0.002)$. The question is asking, what is $\text{P}(Y < 4)$?

$$
\begin{aligned}
\text{P}(Y < 4) &= F_Y(4) && \text{The CDF measures } \text{P}(Y < y) \\
&= 1 - e^{-\lambda \cdot y} && \text{The CDF of an Exp} \\
&= 1 - e^{-0.002 \cdot 4} && \text{The CDF of an Exp} \\
&\approx 0.008
\end{aligned}
$$

Note that it is possible to answer this question using the PDF, but it will require solving an integral.

Exponential is Memoryless

One way to gain intuition for what is meant by the "poisson process" is through the proof that the exponential distribution is "memoryless". That means that the occurrence (or lack of occurrence) of events in the past does not change our belief as to how long until the next occurrence. This can be stated formally. If $X \sim \text{Exp}(\lambda)$ then for an interval of time until the start s, and a proceeding, query, interval of time t:

$$\text{P}(X > s + t | X > s) = \text{P}(X > t)$$

Which is something we can prove:

$$\begin{aligned}
\text{P}(X > s + t | X > s) &= \frac{\text{P}(X > s + t \text{ and } X > s)}{\text{P}(X > s)} && \text{Def of conditional prob.} \\
&= \frac{\text{P}(X > s + t)}{\text{P}(X > s)} && \text{Because } X > s + t \text{ implies } X > s \\
&= \frac{1 - F_X(s + t)}{1 - F_X(s)} && \text{Def of CDF} \\
&= \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} && \text{By CDF of Exp} \\
&= e^{-\lambda t} && \text{Simplify} \\
&= 1 - F_X(t) && \text{By CDF of Exp} \\
&= \text{P}(X > t) && \text{Def of CDF}
\end{aligned}$$

Normal Distribution

The single most important random variable type is the Normal (aka Gaussian) random variable, parametrized by a mean (μ) and variance (σ^2), or sometimes equivalently written as mean and variance (σ^2). If X is a normal variable we write $X \sim N(\mu, \sigma^2)$. The normal is important for many reasons: it is generated from the summation of independent random variables and as a result it occurs often in nature. Many things in the world are not distributed normally but data scientists and computer scientists model them as Normal distributions anyways. Why? Because it is the most entropic (conservative) modelling decision that we can make for a random variable while still matching a particular expectation (average value) and variance (spread).

The Probability Density Function (PDF) for a Normal $X \sim N(\mu, \sigma^2)$ is:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Notice the x in the exponent of the PDF function. When x is equal to the mean (μ) then e is raised to the power of 0 and the PDF is maximized.

By definition a Normal has $\mathrm{E}[X] = \mu$ and $\mathrm{Var}(X) = \sigma^2$.

There is no closed form for the integral of the Normal PDF, and as such there is no closed form CDF. However we can use a transformation of any normal to a normal with a precomputed CDF. The result of this mathematical gymnastics is that the CDF for a Normal $X \sim N(\mu, \sigma^2)$ is:

$$F_X(x) = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

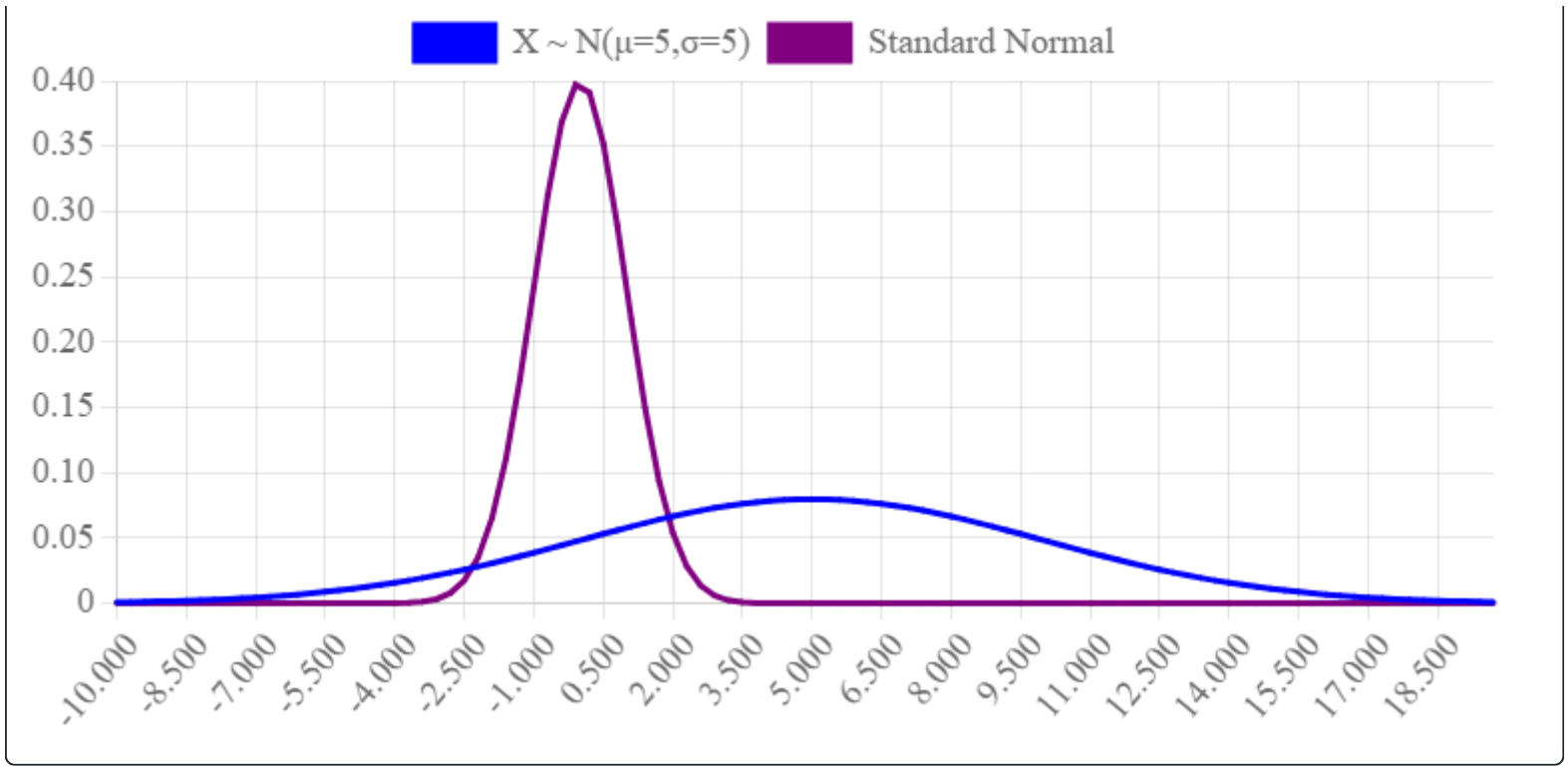
Where Φ is a precomputed function that represents that CDF of the Standard Normal.

Normal (aka Gaussian) Random Variable

Notation:	$X \sim \mathrm{N}(\mu, \sigma^2)$
Description:	A common, naturally occurring distribution.
Parameters:	$\mu \in \mathbb{R}$, the mean. $\sigma^2 \in \mathbb{R}$, the variance.
Support:	$x \in \mathbb{R}$
PDF equation:	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
CDF equation:	$F(x) = \phi(\frac{x - \mu}{\sigma})$ Where ϕ is the CDF of the standard normal
Expectation:	$\mathrm{E}[X] = \mu$
Variance:	$\mathrm{Var}(X) = \sigma^2$

PDF graph:

Parameter μ: 5 Parameter σ: 5



Linear Transform

If X is a Normal such that $X \sim N(\mu, \sigma^2)$ and Y is a linear transform of X such that $Y = aX + b$ then Y is also a Normal where:

$$Y \sim N(a\mu + b, a^2\sigma^2)$$

Projection to Standard Normal

For any Normal X we can find a linear transform from X to the standard normal $Z \sim N(0, 1)$. Note that Z is the typical notation choice for the standard normal. For any normal, if you subtract the mean (μ) of the normal and divide by the standard deviation (σ) the result is always the standard normal. We can prove this mathematically. Let $W = \frac{X-\mu}{\sigma}$:

$$\begin{aligned}
W &= \frac{X - \mu}{\sigma} && \text{Transform X: Subtract by } \mu \text{ and diving by } \sigma \\
&= \frac{1}{\sigma}X - \frac{\mu}{\sigma} && \text{Use algebra to rewrite the equation} \\
&= aX + b && \text{Linear transform where } a = \frac{1}{\mu}, b = -\frac{\mu}{\sigma} \\
&\sim N(a\mu + b, a^2\sigma^2) && \text{The linear transform of a Normal is another Normal} \\
&\sim N(\frac{\mu}{\sigma} - \frac{\mu}{\sigma}, \frac{\sigma^2}{\sigma^2}) && \text{Substituting values in for } a \text{ and } b \\
&\sim N(0, 1) && \text{The standard normal}
\end{aligned}$$

Using this transform we can express $F_X(x)$, the CDF of X, in terms of the known CDF of Z, $F_Z(x)$. Since the CDF of Z is so common it gets its own Greek symbol: $\Phi(x)$

$$\begin{aligned}
F_X(x) &= P(X \leq x) \\
&= P\left(\frac{X - \mu}{\sigma} \leq \frac{x - \mu}{\sigma}\right) \\
&= P\left(Z \leq \frac{x - \mu}{\sigma}\right) \\
&= \Phi\left(\frac{x - \mu}{\sigma}\right)
\end{aligned}$$

The values of $\Phi(x)$ can be looked up in a table. Every modern programming language also has the ability to calculate the CDF of a normal random variable!

Example: Let $X \sim \mathcal{N}(3, 16)$, what is $P(X > 0)$?

$$\begin{aligned}
P(X > 0) &= P\left(\frac{X - 3}{4} > \frac{0 - 3}{4}\right) = P\left(Z > -\frac{3}{4}\right) = 1 - P\left(Z \leq -\frac{3}{4}\right) \\
&= 1 - \Phi(-\frac{3}{4}) = 1 - (1 - \Phi(\frac{3}{4})) = \Phi(\frac{3}{4}) = 0.7734
\end{aligned}$$

What is $P(2 < X < 5)$?

$$\begin{aligned} P(2 < X < 5) &= P\left(\frac{2-3}{4} < \frac{X-3}{4} < \frac{5-3}{4}\right) = P\left(-\frac{1}{4} < Z < \frac{2}{4}\right) \\ &= \Phi(\frac{2}{4}) - \Phi(-\frac{1}{4}) = \Phi(\frac{1}{2}) - (1 - \Phi(\frac{1}{4})) = 0.2902 \end{aligned}$$

Example: You send voltage of 2 or -2 on a wire to denote 1 or 0. Let X = voltage sent and let R = voltage received. $R = X + Y$, where $Y \sim \mathcal{N}(0, 1)$ is noise. When decoding, if $R \geq 0.5$ we interpret the voltage as 1, else 0. What is $P(\text{error after decoding} | \text{original bit} = 1)$?

$$\begin{aligned} P(X + Y < 0.5) &= P(2 + Y < 0.5) \\ &= P(Y < -1.5) \\ &= \Phi(-1.5) \\ &\approx 0.0668 \end{aligned}$$

Example: The 67% rule of a normal within one standard deviation. What is the probability that a normal variable $X \sim \text{N}(\mu, \sigma)$ has a value within one standard deviation of its mean?

$$\begin{aligned} \text{P}(\text{Within one } \sigma \text{ of } \mu) &= \text{P}(\mu - \sigma < X < \mu + \sigma) \\ &= \text{P}(X < \mu + \sigma) - \text{P}(X < \mu - \sigma) && \text{Prob of a range} \\ &= \Phi\Big(\frac{(\mu + \sigma) - \mu}{\sigma}\Big) - \Phi\Big(\frac{(\mu - \sigma) - \mu}{\sigma}\Big) && \text{CDF of Normal} \\ &= \Phi\Big(\frac{\sigma}{\sigma}\Big) - \Phi\Big(\frac{-\sigma}{\sigma}\Big) && \text{Cancel } \mu\text{s} \\ &= \Phi(1) - \Phi(-1) && \text{Cancel } \sigma\text{s} \\ &\approx 0.8413 - 0.1587 \approx 0.683 && \text{Plug into } \Phi \end{aligned}$$

We made no assumption about the value of μ or the value of σ so this will apply to every single normal random variable. Since it uses the Normal CDF this doesn't apply to other types of random variables.

CDF Calculator

To calculate the Cumulative Density Function (CDF) for a normal (aka Gaussian) random variable at a value x, also writen as $F(x)$, you can transform your distribution to the "standard normal" and look up the corresponding value in the standard normal CDF. However, most programming libraries will provide a normal cdf funciton:

Norm CDF Calculator

x 0.0

mu 0

std 1

`norm.cdf(x, mu, std)`

In python you can calculate these values using the `scipy` library

```python
from scipy import stats

# get the input values
mean = 1.0
std_dev = 0.5
query = 0.1 # aka x

# calc the CDF in two lines
X = stats.norm(mean, std_dev)
p = X.cdf(query)

# calc the CDF in one line
p = stats.norm.cdf(query, mean, std_dev)
```

It is important to note that in the python library, the second parameter for the Normal distribution is standard deviation **not** variance, as it is typically defined in math notation. Recall that standard deviation is the square root of variance.

Binomial Approximation

There are times when it is exceptionally hard to numerically calculate probabilities for a binomial distribution, especially when n is large. For example, say $X \sim \text{Bin}(n = 10000, p = 0.5)$ and you want to calculate $\text{P}(X > 5500)$. The correct formula is:

$$\begin{aligned}\text{P}(X > 55) &= \sum_{i=5500}^{10000} \text{P}(X = x) \\ &= \sum_{i=5500}^{10000} \binom{10000}{i} p^i (1-p)^{10000-i}\end{aligned}$$

That is a difficult value to calculate. Luckily there is an easier way. For deep reasons which we will cover in our section on "uncertainty theory" it turns out that a binomial distribution can be very well approximated by both Normal distributions and Poisson distributions if n is large enough.

Use the Poisson approximation when n is large (>20) and p is small (<0.05). A slight dependence between results of each experiment is ok

Use the Normal approximation when n is large (>20), and p is mid-ranged. Specifically it's considered an accurate approximation when the variance is greater then 10, in other words: $np(1-p) > 10$. There are situations where either a Poisson or a Normal can be used to approximate a Binomial. In that situation go with the Normal!

Poisson Approximation

When defining the Poisson we proved that a Binomial in the limit as $n \to \infty$ and $p = \lambda/n$ is a Poisson. That same logic can be used to show that a Poisson is a great approximation for a Binomial when the Binomial has extreme values of n and p. A Poisson random variable approximates Binomial where n is large, p is small, and $\lambda = np$ is “moderate”. Interestingly, to calculate the things we care about (PMF, expectation, variance) we no longer need to know n and p. We only need to provide λ which we call the rate. When approximating a Poisson with a Binomial, always choose $\lambda = n \cdot p$.

There are different interpretations of "moderate". The accepted ranges are $n > 20$ and $p < 0.05$ or $n > 100$ and $p < 0.1$.

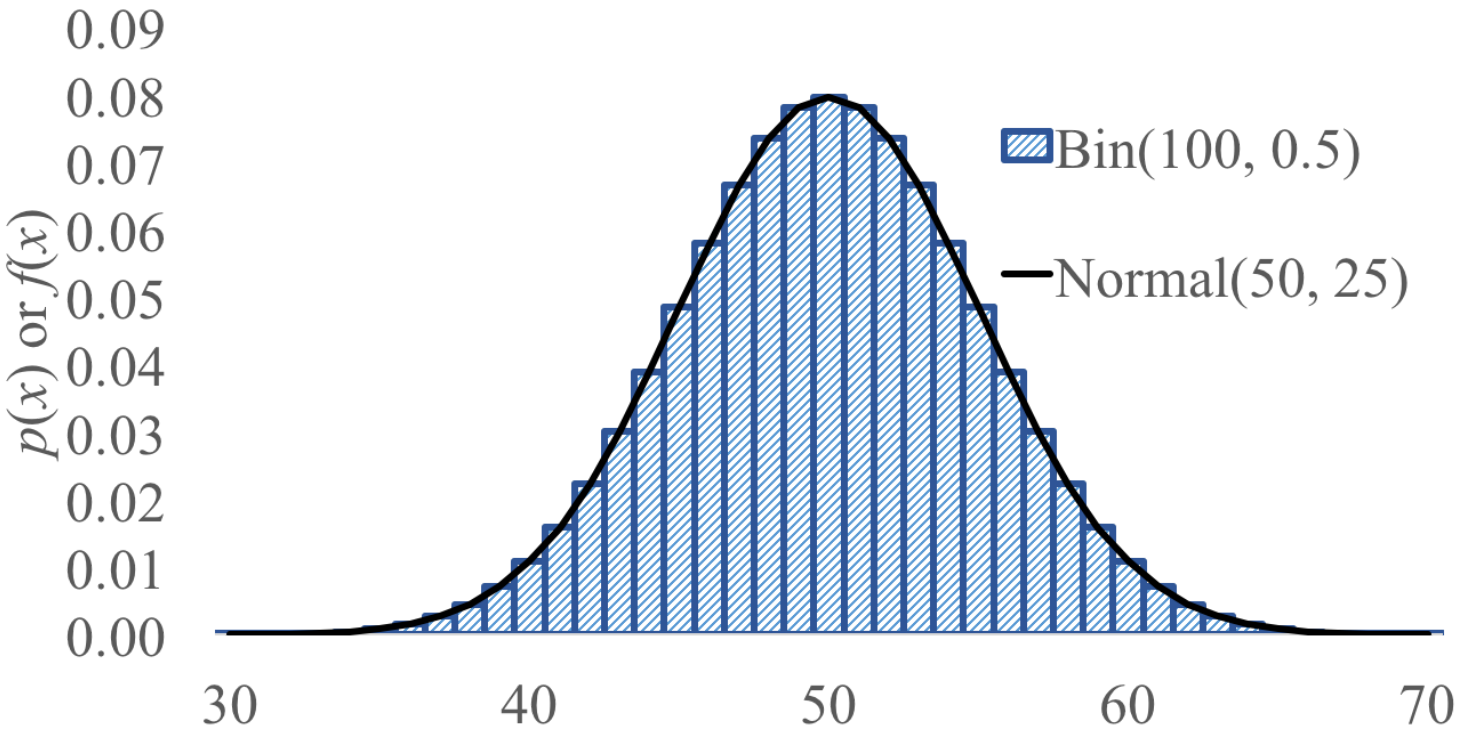
Let's say you want to send a bit string of length $n = 10^4$ where each bit is independently corrupted with $p = 10^{-6}$. What is the probability that the message will arrive uncorrupted? You can solve this using a Poisson with $\lambda = np = 10^4 10^{-6} = 0.01$. Semantically, $\lambda = 0.01$ means that we expect 0.01 corrupt bits per string, assuming bits are continuous. Let $X \sim Poi(0.01)$ be the number of corrupted bits. Using the PMF for Poisson:

$$\begin{aligned}P(X = 0) &= \frac{\lambda^i}{i!} e^{-\lambda} \\ &= \frac{0.01^0}{0!} e^{-0.01} \\ &\sim 0.9900498\end{aligned}$$

We could have also modelled X as a binomial such that $X \sim Bin(10^4, 10^{-6})$. That would have been impossible to calculate on a computer but would have resulted in the same number (up to the millionth decimal).

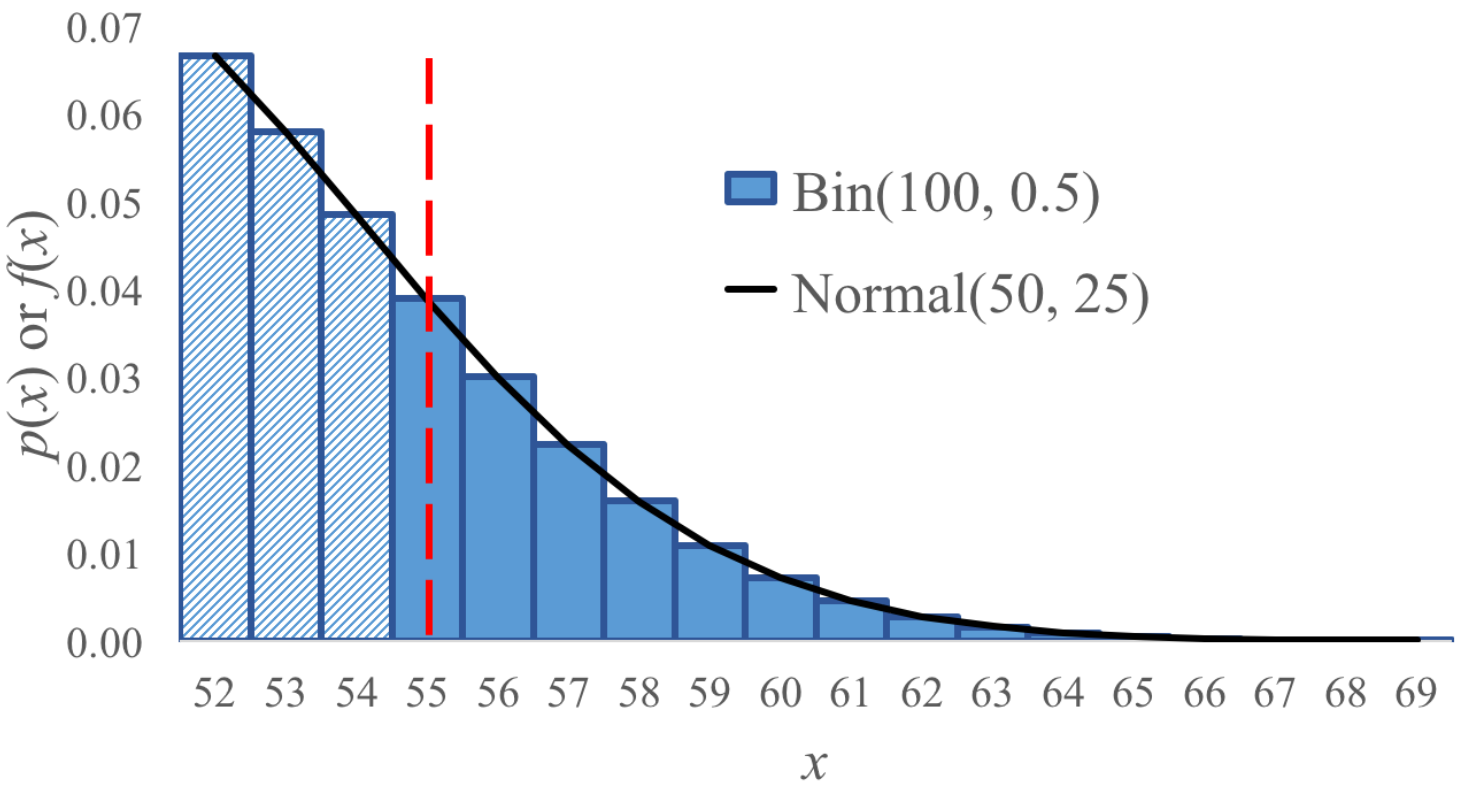
Normal Approximation

For a Binomial where n is large and p is mid-ranged, a Normal can be used to approximate the Binomial. Let's take a side by side view of a normal and a binomial:



Lets say our binomial is a random variable $X \sim \text{Bin}(100, 0.5)$ and we want to calculate $P(X \geq 55)$. We could cheat by using the closest fit normal (in this case $Y \sim N(50, 25)$). How did we choose that particular Normal? Simply select one with a mean and variance that matches the Binomial expectation and variance. The binomial expectation is $np = 100 \cdot 0.5 = 50$. The Binomial variance is $np(1-p) = 100 \cdot 0.5 \cdot 0.5 = 25$.

You can use a Normal distribution to approximate a Binomial $X \sim \text{Bin}(n, p)$. To do so define a normal $Y \sim (E[X], Var(X))$. Using the Binomial formulas for expectation and variance, $Y \sim (np, np(1-p))$. This approximation holds for large n and moderate p. That gets you very close. However since a Normal is continuous and Binomial is discrete we have to use a continuity correction to discretize the Normal.



$$P(X = k) \sim P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

You should get comfortable deciding what continuity correction to use. Here are a few examples of discrete probability questions and the continuity correction:

Discrete (Binomial) probability question	Equivalent continuous probability question
$P(X = 6)$	$P(5.5 < X < 6.5)$
$P(X \geq 6)$	$P(X > 5.5)$
$P(X > 6)$	$P(X > 6.5)$
$P(X < 6)$	$P(X < 5.5)$
$P(X \leq 6)$	$P(X < 6.5)$

Example: 100 visitors to your website are given a new design. Let X = # of people who were given the new design and spend more time on your website. Your CEO will endorse the new design if $X \geq 65$. What is $P(\text{CEO endorses change}|\text{it has no effect})$?

$E[X] = np = 50$. $\text{Var}(X) = np(1-p) = 25$. $\sigma = \sqrt{\text{Var}(X)} = 5$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(\mu = 50, \sigma^2 = 25)$.

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y - 50}{5} > \frac{64.5 - 50}{5}\right) = 1 - \Phi(2.9) = 0.0019$$

Example: Stanford accepts 2480 students and each student has a 68% chance of attending. Let X = # students who will attend. $X \sim \text{Bin}(2480, 0.68)$. What is $P(X > 1745)$?

$E[X] = np = 1686.4$. $\text{Var}(X) = np(1-p) = 539.7$. $\sigma = \sqrt{\text{Var}(X)} = 23.23$. We can thus use a Normal approximation: $Y \sim \mathcal{N}(\mu = 1686.4, \sigma^2 = 539.7)$.

$$\begin{aligned} P(X > 1745) &\approx P(Y > 1745.5) \\ &\approx P\left(\frac{Y - 1686.4}{23.23} > \frac{1745.5 - 1686.4}{23.23}\right) \\ &\approx 1 - \Phi(2.54) = 0.0055 \end{aligned}$$

100 Binomial Problems

Just for fun (and to give you a lot of practice) I wrote a generative probabilistic program which could sample binomial distribution problems. Here are 100 binomial questions:

Questions

Question 1: Laura is running a server cluster with 50 computers. The probability of a crash on a given server is 0.5. What is the standard deviation of crashes?

Answer 1:
Let X be the number of crashes. $X \sim \text{Bin}(n = 50, p = 0.5)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{50 \cdot 0.5 \cdot (1 - 0.5)} \\ &= 3.54\end{aligned}$$

Question 2: You are showing an online-ad to 30 people. The probability of an ad ignore on each ad shown is 2/3. What is the expected number of ad clicks?

Answer 2:
Let X be the number of ad clicks. $X \sim \text{Bin}(n = 30, p = 1/3)$

$$\begin{aligned}\text{E}[X] &= np \\ &= 30 \cdot 1/3 \\ &= 10\end{aligned}$$

Question 3: A machine learning algorithm makes binary predictions. The machine learning algorithm makes 50 guesses where the probability of a incorrect prediction on a given guess is 19/25. What is the probability that the number of correct predictions is greater than 0?

Answer 3:
Let X be the number of correct predictions. $X \sim \text{Bin}(n = 50, p = 6/25)$

$$\begin{aligned}\text{P}(X > 0) &= 1 - \text{P}(0 <= X <= 0) \\ &= 1 - \binom{n}{0} p^0 (1-p)^{n-0}\end{aligned}$$

Question 4: Wind blows independently across 50 locations. The probability of no wind at a given location is 0.5. What is the expected number of locations that have wind?

Answer 4:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 50, p = 0.5)$

$$\begin{aligned}\text{E}[X] &= np \\ &= 50 \cdot 0.5 \\ &= 25.0\end{aligned}$$

Question 5: Wind blows independently across 30 locations. What is the standard deviation of locations that have wind? the probability of wind at each location is 0.6.

Answer 5:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 30, p = 0.6)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{30 \cdot 0.6 \cdot (1-0.6)} \\ &= 2.68\end{aligned}$$

Question 6: You are trying to mine bitcoins. There are 50 independent attempts where the probability of a mining a bitcoin on a given attempt is 0.6. What is the expectation of bitcoins mined?

Answer 6:
Let X be the number of bitcoins mined. $X \sim \text{Bin}(n = 50, p = 0.6)$

$$\begin{aligned}\text{E}[X] &= np \\ &= 50 \cdot 0.6 \\ &= 30.0\end{aligned}$$

Question 7: You are testing a new medicine on 40 patients. What is P(X is exactly 38)? The number of cured patients can be represented by a random variable X. X ~ Bin(40, 3/10).

Answer 7:
Let X be the number of cured patients. $X \sim \text{Bin}(n = 40, p = 3/10)$

$$\begin{aligned}\text{P}(X = 38) &= \binom{n}{38} p^{38}(1-p)^{n-38} \\ &= \binom{40}{38} 3/10^{38}(1-3/10)^{40-38} \\ &< 0.00001\end{aligned}$$

Question 8: You are manufacturing chips and are testing for defects. There are 50 independent tests and 0.5 is the probability of a defect on each test. What is the standard deviation of defects?

Answer 8:
Let X be the number of defects. $X \sim \text{Bin}(n = 50, p = 0.5)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{50 \cdot 0.5 \cdot (1-0.5)} \\ &= 3.54\end{aligned}$$

Question 9: Laura is flipping a coin 12 times. The probability of a tail on a given coin-flip is 5/12. What is the probability that the number of tails is greater than or equal to 2?

Answer 9:
Let X be the number of tails. $X \sim \text{Bin}(n = 12, p = 5/12)$

$$\begin{aligned}\text{P}(X >= 2) &= 1 - \text{P}(0 <= X <= 1) \\ &= 1 - \sum_{i=0}^{1} \binom{n}{i} p^{i}(1-p)^{n-i}\end{aligned}$$

Question 10: You are asking a survey question where responses are "like" or "dislike". There are 30 responses. You can assume each response is independent where the probability of a dislike on a given response is 1/6. What is the probability that the number of likes is greater than 28?

Answer 10:
Let X be the number of likes. $X \sim \text{Bin}(n = 30, p = 5/6)$

$$\begin{aligned} \text{P}(X > 28) &= P(29 <= X <= 30) \\ &= \sum_{i=29}^{30} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 11: A ball hits a series of 50 pins where it can bounce either right or left. The probability of a left on a given pin hit is 0.4. What is the standard deviation of rights?

Answer 11:
Let X be the number of rights. $X \sim \text{Bin}(n = 50, p = 3/5)$

$$\begin{aligned} \text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{50 \cdot 3/5 \cdot (1 - 3/5)} \\ &= 3.46 \end{aligned}$$

Question 12: You are sending a stream of 30 bits to space. The probability of a no corruption on a given bit is 1/3. What is the probability that the number of corruptions is 10?

Answer 12:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 30, p = 2/3)$

$$\begin{aligned} \text{P}(X = 10) &= \binom{n}{10} p^{10} (1-p)^{n-10} \\ &= \binom{30}{10} 2/3^{10} (1 - 2/3)^{30-10} \\ &= 0.00015 \end{aligned}$$

Question 13: Wind blows independently across locations. The probability of wind at a given location is 0.9. The number of independent locations is 20. What is the probability that the number of locations that have wind is not less than 19?

Answer 13:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 20, p = 0.9)$

$$\begin{aligned} \text{P}(X >= 19) &= P(19 <= X <= 20) \\ &= \sum_{i=19}^{20} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 14: You are sending a stream of bits to space. There are 30 independent bits where 5/6 is the probability of a no corruption on each bit. What is the probability that the number of corruptions is 21?

Answer 14:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 30, p = 1/6)$

$$\begin{aligned} \text{P}(X = 21) &= \binom{n}{21} p^{21} (1-p)^{n-21} \\ &= \binom{30}{21} 1/6^{21} (1 - 1/6)^{30-21} \\ &< 0.00001 \end{aligned}$$

Question 15: Cody generates random bit strings. There are 20 independent bits. Each bit has a 1/4 probability of resulting in a 1. What is the probability that the number of 1s is 11?

Answer 15:
Let X be the number of 1s. $X \sim \text{Bin}(n = 20, p = 1/4)$

$$\begin{aligned}
\text{P}(X = 11) &= \binom{n}{11} p^{11}(1-p)^{n-11} \\
&= \binom{20}{11} 1/4^{11}(1-1/4)^{20-11} \\
&= 0.00301
\end{aligned}$$

Question 16: In a restaurant some customers ask for a water with their meal. A random sample of 40 customers is selected where the probability of a water requested by a given customer is 9/20. What is the probability that the number of waters requested is 16?

Answer 16:
Let X be the number of waters requested. $X \sim \text{Bin}(n = 40, p = 9/20)$

$$\begin{aligned}
\text{P}(X = 16) &= \binom{n}{16} p^{16}(1-p)^{n-16} \\
&= \binom{40}{16} 9/20^{16}(1-9/20)^{40-16} \\
&= 0.10433
\end{aligned}$$

Question 17: A student is guessing randomly on an exam with 12 questions. What is the expected number of correct answers? the probability of a correct answer on a given question is 5/12.

Answer 17:
Let X be the number of correct answers. $X \sim \text{Bin}(n = 12, p = 5/12)$

$$\begin{aligned}
\text{E}[X] &= np \\
&= 12 \cdot 5/12 \\
&= 5
\end{aligned}$$

Question 18: Laura is trying to mine bitcoins. The number of bitcoins mined can be represented by a random variable X. X ~ Bin(n = 100, p = 1/2). What is P(X is equal to 53)?

Answer 18:
Let X be the number of bitcoins mined. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned}
\text{P}(X = 53) &= \binom{n}{53} p^{53}(1-p)^{n-53} \\
&= \binom{100}{53} 1/2^{53}(1-1/2)^{100-53} \\
&= 0.06659
\end{aligned}$$

Question 19: You are showing an online-ad to customers. The add is shown to 100 people. The probability of an ad ignore on a given ad shown is 1/2. What is the standard deviation of ad clicks?

Answer 19:
Let X be the number of ad clicks. $X \sim \text{Bin}(n = 100, p = 0.5)$

$$\begin{aligned}
\text{Std}(X) &= \sqrt{np(1-p)} \\
&= \sqrt{100 \cdot 0.5 \cdot (1-0.5)} \\
&= 5.00
\end{aligned}$$

Question 20: You are running a server cluster with 40 computers. 5/8 is the probability of a computer continuing to work on each server. What is the expected number of crashes?

Answer 20:
Let X be the number of crashes. $X \sim \text{Bin}(n = 40, p = 3/8)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 40 \cdot 3/8 \\ &= 15 \end{aligned}$$

Question 21: You are hashing 100 strings into a hashtable. The probability of a hash to the first bucket on a given string hash is 3/20. What is the probability that the number of hashes to the first bucket is greater than or equal to 97?

Answer 21:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 100, p = 3/20)$

$$\begin{aligned} \text{P}(X >= 97) &= P(97 <= X <= 100) \\ &= \sum_{i=97}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 22: You are running in an election with 50 voters. 6/25 is the probability of a vote for you on each vote. What is the probability that the number of votes for you is less than 2?

Answer 22:
Let X be the number of votes for you. $X \sim \text{Bin}(n = 50, p = 6/25)$

$$\begin{aligned} \text{P}(X < 2) &= P(0 <= X <= 1) \\ &= \sum_{i=0}^{1} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 23: Irina is sending a stream of 40 bits to space. The probability of a corruption on each bit is 3/4. What is the probability that the number of corruptions is 22?

Answer 23:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned} \text{P}(X = 22) &= \binom{n}{22} p^{22} (1-p)^{n-22} \\ &= \binom{40}{22} 3/4^{22} (1-3/4)^{40-22} \\ &= 0.00294 \end{aligned}$$

Question 24: You are hashing 100 strings into a hashtable. The probability of a hash to the first bucket on a given string hash is 9/50. What is the probability that the number of hashes to the first bucket is greater than 97?

Answer 24:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 100, p = 9/50)$

$$\begin{aligned} \text{P}(X > 97) &= P(98 <= X <= 100) \\ &= \sum_{i=98}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 25: You generate random bit strings. There are 100 independent bits. The probability of a 1 at a given bit is 3/25. What is the probability that the number of 1s is less than 97?

Answer 25:
Let X be the number of 1s. $X \sim \text{Bin}(n = 100, p = 3/25)$

$$\begin{aligned}\text{P}(X < 97) &= 1 - \text{P}(97 <= X <= 100) \\ &= 1 - \sum_{i=97}^{100} \binom{n}{i} p^i (1-p)^{n-i}\end{aligned}$$

Question 26: You are manufacturing toys and are testing for defects. What is the probability that the number of defects is greater than 1? the probability of a non-defect on a given test is 16/25 and you test 50 objects.

Answer 26:
Let X be the number of defects. $X \sim \text{Bin}(n = 50, p = 9/25)$

$$\begin{aligned}\text{P}(X > 1) &= 1 - \text{P}(0 <= X <= 1) \\ &= 1 - \sum_{i=0}^{1} \binom{n}{i} p^i (1-p)^{n-i}\end{aligned}$$

Question 27: Laura is sending a stream of 40 bits to space. The number of corruptions can be represented by a random variable X. X is a Binomial with n = 40 and p = 3/4. What is P(X = 25)?

Answer 27:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned}\text{P}(X = 25) &= \binom{n}{25} p^{25} (1-p)^{n-25} \\ &= \binom{40}{25} 3/4^{25} (1 - 3/4)^{40-25} \\ &= 0.02819\end{aligned}$$

Question 28: 100 trials are run. What is the probability that the number of successes is 78? 1/2 is the probability of a success on each trial.

Answer 28:
Let X be the number of successes. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned}\text{P}(X = 78) &= \binom{n}{78} p^{78} (1-p)^{n-78} \\ &= \binom{100}{78} 1/2^{78} (1 - 1/2)^{100-78} \\ &< 0.00001\end{aligned}$$

Question 29: You are flipping a coin. You flip the coin 20 times. The probability of a tail on a given coin-flip is 1/10. What is the standard deviation of heads?

Answer 29:
Let X be the number of heads. $X \sim \text{Bin}(n = 20, p = 0.9)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{20 \cdot 0.9 \cdot (1 - 0.9)} \\ &= 1.34\end{aligned}$$

Question 30: Irina is showing an online-ad to 12 people. 5/12 is the probability of an ad click on each ad shown. What is the probability that the number of ad clicks is less than or equal to 11?

Answer 30:
Let X be the number of ad clicks. $X \sim \text{Bin}(n = 12, p = 5/12)$

$$\begin{aligned}\text{P}(X <= 11) &= 1 - \text{P}(12 <= X <= 12) \\ &= 1 - \binom{n}{12} p^1 2 (1-p)^{n-12}\end{aligned}$$

Question 31: You are flipping a coin 50 times. 19/25 is the probability of a head on each coin-flip. What is the standard deviation of tails?

Answer 31:
Let X be the number of tails. $X \sim \text{Bin}(n = 50, p = 6/25)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{50 \cdot 6/25 \cdot (1 - 6/25)} \\ &= 3.02\end{aligned}$$

Question 32: You are running in an election with 100 voters. The probability of a vote for you on each vote is 1/4. What is the probability that the number of votes for you is less than or equal to 97?

Answer 32:
Let X be the number of votes for you. $X \sim \text{Bin}(n = 100, p = 1/4)$

$$\begin{aligned}\text{P}(X <= 97) &= 1 - \text{P}(98 <= X <= 100) \\ &= 1 - \sum_{i=98}^{100} \binom{n}{i} p^i (1-p)^{n-i}\end{aligned}$$

Question 33: You are running a server cluster with 40 computers. What is the probability that the number of crashes is less than or equal to 39? 3/4 is the probability of a computer continuing to work on each server.

Answer 33:
Let X be the number of crashes. $X \sim \text{Bin}(n = 40, p = 1/4)$

$$\begin{aligned}\text{P}(X <= 39) &= 1 - \text{P}(40 <= X <= 40) \\ &= 1 - \binom{n}{40} p^4 0 (1-p)^{n-40}\end{aligned}$$

Question 34: Waddie is sending a stream of bits to space. Waddie sends 100 bits. The probability of a corruption on each bit is 1/2. What is the standard deviation of corruptions?

Answer 34:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{100 \cdot 1/2 \cdot (1 - 1/2)} \\ &= 5.00\end{aligned}$$

Question 35: A student is guessing randomly on an exam with 100 questions. Each question has a 0.5 probability of resulting in a incorrect answer. What is the probability that the number of correct answers is greater than 97?

Answer 35:
Let X be the number of correct answers. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned} \text{P}(X > 97) &= P(98 <= X <= 100) \\ &= \sum_{i=98}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 36: You are testing a new medicine on patients. 0.5 is the probability of a cured patient on each trial. There are 10 independent trials. What is the expected number of cured patients?

Answer 36:
Let X be the number of cured patients. $X \sim \text{Bin}(n = 10, p = 0.5)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 10 \cdot 0.5 \\ &= 5.0 \end{aligned}$$

Question 37: A ball hits a series of pins where it can either go right or left. The number of independent pin hits is 100. The probability of a right on each pin hit is 0.5. What is the standard deviation of rights?

Answer 37:
Let X be the number of rights. $X \sim \text{Bin}(n = 100, p = 0.5)$

$$\begin{aligned} \text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{100 \cdot 0.5 \cdot (1-0.5)} \\ &= 5.00 \end{aligned}$$

Question 38: You are flipping a coin 40 times. The probability of a head on a given coin-flip is 1/2. What is the probability that the number of heads is 38?

Answer 38:
Let X be the number of heads. $X \sim \text{Bin}(n = 40, p = 1/2)$

$$\begin{aligned} \text{P}(X = 38) &= \binom{n}{38} p^{38} (1-p)^{n-38} \\ &= \binom{40}{38} 1/2^{38} (1-1/2)^{40-38} \\ &< 0.00001 \end{aligned}$$

Question 39: 100 trials are run and the probability of a success on a given trial is 1/2. What is the standard deviation of successes?

Answer 39:
Let X be the number of successes. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned} \text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{100 \cdot 1/2 \cdot (1-1/2)} \\ &= 5.00 \end{aligned}$$

Question 40: You are trying to mine bitcoins. There are 40 independent attempts. The probability of a mining a bitcoin on each attempt is 3/10. What is the probability that the number of bitcoins mined is 19?

Answer 40:
Let X be the number of bitcoins mined. $X \sim \text{Bin}(n = 40, p = 3/10)$

$$\begin{aligned}\mathrm{P}(X = 19) &= \binom{n}{19} p^{19}(1-p)^{n-19} \\ &= \binom{40}{19} 3/10^{19}(1-3/10)^{40-19} \\ &= 0.00852\end{aligned}$$

Question 41: 20 trials are run. 0.5 is the probability of a failure on each trial. What is the probability that the number of successes is 6?

Answer 41:
Let X be the number of successes. $X \sim \mathrm{Bin}(n = 20, p = 0.5)$

$$\begin{aligned}\mathrm{P}(X = 6) &= \binom{n}{6} p^{6}(1-p)^{n-6} \\ &= \binom{20}{6} 0.5^{6}(1-0.5)^{20-6} \\ &= 0.03696\end{aligned}$$

Question 42: You are flipping a coin. What is the probability that the number of tails is 0? there are 30 independent coin-flips where the probability of a head on a given coin-flip is 5/6.

Answer 42:
Let X be the number of tails. $X \sim \mathrm{Bin}(n = 30, p = 1/6)$

$$\begin{aligned}\mathrm{P}(X = 0) &= \binom{n}{0} p^{0}(1-p)^{n-0} \\ &= \binom{30}{0} 1/6^{0}(1-1/6)^{30-0} \\ &= 0.00421\end{aligned}$$

Question 43: In a restaurant some customers ask for a water with their meal. A random sample of 20 customers is selected and each customer has a 1/4 probability of resulting in a water not requested. What is the probability that the number of waters requested is 14?

Answer 43:
Let X be the number of waters requested. $X \sim \mathrm{Bin}(n = 20, p = 3/4)$

$$\begin{aligned}\mathrm{P}(X = 14) &= \binom{n}{14} p^{14}(1-p)^{n-14} \\ &= \binom{20}{14} 3/4^{14}(1-3/4)^{20-14} \\ &= 0.16861\end{aligned}$$

Question 44: A student is guessing randomly on an exam. 3/8 is the probability of a incorrect answer on each question. The number of independent questions is 40. What is the probability that the number of correct answers is less than or equal to 37?

Answer 44:
Let X be the number of correct answers. $X \sim \mathrm{Bin}(n = 40, p = 5/8)$

$$\begin{aligned}\mathrm{P}(X <= 37) &= 1 - \mathrm{P}(38 <= X <= 40) \\ &= 1 - \sum_{i=38}^{40} \binom{n}{i} p^{i}(1-p)^{n-i}\end{aligned}$$

Question 45: You are running in an election with 30 voters. 3/5 is the probability of a vote for you on each vote. What is the standard deviation of votes for you?

Answer 45:
Let X be the number of votes for you. $X \sim \text{Bin}(n = 30, p = 3/5)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{30 \cdot 3/5 \cdot (1 - 3/5)} \\ &= 2.68\end{aligned}$$

Question 46: Charlotte is flipping a coin 100 times. The probability of a tail on each coin-flip is 0.5. What is the probability that the number of tails is greater than 2?

Answer 46:
Let X be the number of tails. $X \sim \text{Bin}(n = 100, p = 0.5)$

$$\begin{aligned}\text{P}(X > 2) &= 1 - \text{P}(0 <= X <= 2) \\ &= 1 - \sum_{i=0}^{2} \binom{n}{i} p^i (1-p)^{n-i}\end{aligned}$$

Question 47: You are trying to mine bitcoins. You try 50 times. 3/5 is the probability of a not mining a bitcoin on each attempt. What is the probability that the number of bitcoins mined is 14?

Answer 47:
Let X be the number of bitcoins mined. $X \sim \text{Bin}(n = 50, p = 2/5)$

$$\begin{aligned}\text{P}(X = 14) &= \binom{n}{14} p^{14} (1-p)^{n-14} \\ &= \binom{50}{14} 2/5^{14} (1 - 2/5)^{50-14} \\ &= 0.02597\end{aligned}$$

Question 48: You are testing a new medicine on 100 patients. The probability of a cured patient on a given trial is 3/25. What is the probability that the number of cured patients is not less than 97?

Answer 48:
Let X be the number of cured patients. $X \sim \text{Bin}(n = 100, p = 3/25)$

$$\begin{aligned}\text{P}(X >= 97) &= P(97 <= X <= 100) \\ &= \sum_{i=97}^{100} \binom{n}{i} p^i (1-p)^{n-i}\end{aligned}$$

Question 49: Wind blows independently across 40 locations. What is the probability that the number of locations that have wind is 40? 11/20 is the probability of no wind at each location.

Answer 49:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 40, p = 9/20)$

$$\begin{aligned}\text{P}(X = 40) &= \binom{n}{40} p^{40} (1-p)^{n-40} \\ &= \binom{40}{40} 9/20^{40} (1 - 9/20)^{40-40} \\ &< 0.00001\end{aligned}$$

Question 50: You are showing an online-ad to 30 people. 1/6 is the probability of an ad click on each ad shown. What is the probability that the number of ad clicks is less than or equal to 28?

Answer 50:
Let X be the number of ad clicks. $X \sim \text{Bin}(n = 30, p = 1/6)$

$$\begin{aligned} \text{P}(X <= 28) &= 1 - \text{P}(29 <= X <= 30) \\ &= 1 - \sum_{i=29}^{30} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 51: You are flipping a coin. You flip the coin 40 times and 7/8 is the probability of a head on each coin-flip. What is the standard deviation of tails?

Answer 51:
Let X be the number of tails. $X \sim \text{Bin}(n = 40, p = 1/8)$

$$\begin{aligned} \text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{40 \cdot 1/8 \cdot (1 - 1/8)} \\ &= 2.09 \end{aligned}$$

Question 52: Cody is sending a stream of bits to space. 2/5 is the probability of a no corruption on each bit and there are 20 independent bits. What is the expectation of corruptions?

Answer 52:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 20, p = 3/5)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 20 \cdot 3/5 \\ &= 12 \end{aligned}$$

Question 53: You are running in an election. There are 12 independent votes and 5/6 is the probability of a vote for you on each vote. What is the probability that the number of votes for you is greater than or equal to 9?

Answer 53:
Let X be the number of votes for you. $X \sim \text{Bin}(n = 12, p = 5/6)$

$$\begin{aligned} \text{P}(X >= 9) &= P(9 <= X <= 12) \\ &= \sum_{i=9}^{12} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 54: You are flipping a coin. The number of tails can be represented by a random variable X. X is a Bin(n = 30, p = 5/6). What is the probability that X = 1?

Answer 54:
Let X be the number of tails. $X \sim \text{Bin}(n = 30, p = 5/6)$

$$\begin{aligned} \text{P}(X = 1) &= \binom{n}{1} p^1 (1-p)^{n-1} \\ &= \binom{30}{1} 5/6^1 (1 - 5/6)^{30-1} \\ &< 0.00001 \end{aligned}$$

Question 55: In a restaurant some customers ask for a water with their meal. A random sample of 100 customers is selected where 0.3 is the probability of a water requested by each customer. What is the expected number of waters requested?

Answer 55:
Let X be the number of waters requested. $X \sim \text{Bin}(n = 100, p = 0.3)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 100 \cdot 0.3 \\ &= 30.0 \end{aligned}$$

Question 56: You are hashing strings into a hashtable. 30 strings are hashed. The probability of a hash to the first bucket on each string hash is 1/6. What is the expected number of hashes to the first bucket?

Answer 56:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 30, p = 1/6)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 30 \cdot 1/6 \\ &= 5 \end{aligned}$$

Question 57: You are flipping a coin 100 times. What is the probability that the number of tails is greater than or equal to 98? 19/20 is the probability of a head on each coin-flip.

Answer 57:
Let X be the number of tails. $X \sim \text{Bin}(n = 100, p = 1/20)$

$$\begin{aligned} \text{P}(X >= 98) &= P(98 <= X <= 100) \\ &= \sum_{i=98}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 58: Irina is running a server cluster. What is the probability that the number of crashes is less than 99? the server has 100 computers which crash independently and the probability of a computer continuing to work on a given server is 22/25.

Answer 58:
Let X be the number of crashes. $X \sim \text{Bin}(n = 100, p = 3/25)$

$$\begin{aligned} \text{P}(X < 99) &= 1 - \text{P}(99 <= X <= 100) \\ &= 1 - \sum_{i=99}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 59: You are manufacturing chairs and are testing for defects. You test 100 objects. 1/2 is the probability of a non-defect on each test. What is the probability that the number of defects is not greater than 97?

Answer 59:
Let X be the number of defects. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned} \text{P}(X <= 97) &= 1 - \text{P}(98 <= X <= 100) \\ &= 1 - \sum_{i=98}^{100} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 60: In a restaurant some customers ask for a water with their meal. There are 50 customers. You can assume each customer is independent. 0.2 is the probability of a water requested by each customer. What is the expected number of waters requested?

Answer 60:
Let X be the number of waters requested. $X \sim \text{Bin}(n = 50, p = 0.2)$

$$\begin{aligned}\mathrm{E}[X] &= np \\ &= 50 \cdot 0.2 \\ &= 10.0\end{aligned}$$

Question 61: You are showing an online-ad to 40 people. 1/4 is the probability of an ad ignore on each ad shown. What is the probability that the number of ad clicks is 9?

Answer 61:
Let X be the number of ad clicks. $X \sim \mathrm{Bin}(n = 40, p = 3/4)$

$$\begin{aligned}\mathrm{P}(X = 9) &= \binom{n}{9} p^9 (1-p)^{n-9} \\ &= \binom{40}{9} 3/4^9 (1 - 3/4)^{40-9} \\ &< 0.00001\end{aligned}$$

Question 62: 100 trials are run. Each trial has a 22/25 probability of resulting in a failure. What is the standard deviation of successes?

Answer 62:
Let X be the number of successes. $X \sim \mathrm{Bin}(n = 100, p = 3/25)$

$$\begin{aligned}\mathrm{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{100 \cdot 3/25 \cdot (1 - 3/25)} \\ &= 3.25\end{aligned}$$

Question 63: A machine learning algorithm makes binary predictions. There are 12 independent guesses where the probability of a incorrect prediction on a given guess is 1/6. What is the expected number of correct predictions?

Answer 63:
Let X be the number of correct predictions. $X \sim \mathrm{Bin}(n = 12, p = 5/6)$

$$\begin{aligned}\mathrm{E}[X] &= np \\ &= 12 \cdot 5/6 \\ &= 10\end{aligned}$$

Question 64: Waddie is showing an online-ad to customers. 1/2 is the probability of an ad click on each ad shown. The add is shown to 100 people. What is the average number of ad clicks?

Answer 64:
Let X be the number of ad clicks. $X \sim \mathrm{Bin}(n = 100, p = 1/2)$

$$\begin{aligned}\mathrm{E}[X] &= np \\ &= 100 \cdot 1/2 \\ &= 50\end{aligned}$$

Question 65: Charlotte is testing a new medicine on 50 patients. The probability of a cured patient on a given trial is 1/5. What is the probability that the number of cured patients is 12?

Answer 65:
Let X be the number of cured patients. $X \sim \mathrm{Bin}(n = 50, p = 1/5)$

$$\begin{aligned}
\mathrm{P}(X = 12) &= \binom{n}{12} p^{12}(1-p)^{n-12} \\
&= \binom{50}{12} 1/5^{12}(1-1/5)^{50-12} \\
&= 0.10328
\end{aligned}$$

Question 66: You are running in an election. The number of votes for you can be represented by a random variable X. X is a Bin(n = 50, p = 0.4). What is P(X is exactly 8)?

Answer 66:
Let X be the number of votes for you. $X \sim \mathrm{Bin}(n = 50, p = 0.4)$

$$\begin{aligned}
\mathrm{P}(X = 8) &= \binom{n}{8} p^{8}(1-p)^{n-8} \\
&= \binom{50}{8} 0.4^{8}(1-0.4)^{50-8} \\
&= 0.00017
\end{aligned}$$

Question 67: Irina is flipping a coin 100 times. The probability of a head on a given coin-flip is 1/2. What is the probability that the number of tails is less than or equal to 99?

Answer 67:
Let X be the number of tails. $X \sim \mathrm{Bin}(n = 100, p = 0.5)$

$$\begin{aligned}
\mathrm{P}(X <= 99) &= 1 - \mathrm{P}(100 <= X <= 100) \\
&= 1 - \binom{n}{100} p^{1}00(1-p)^{n-100}
\end{aligned}$$

Question 68: You are manufacturing airplanes and are testing for defects. You test 30 objects and the probability of a defect on a given test is 5/6. What is the probability that the number of defects is 14?

Answer 68:
Let X be the number of defects. $X \sim \mathrm{Bin}(n = 30, p = 5/6)$

$$\begin{aligned}
\mathrm{P}(X = 14) &= \binom{n}{14} p^{14}(1-p)^{n-14} \\
&= \binom{30}{14} 5/6^{14}(1-5/6)^{30-14} \\
&< 0.00001
\end{aligned}$$

Question 69: You are flipping a coin 20 times. The number of heads can be represented by a random variable X. X is a Binomial with 20 trials. Each trial is a success, independently, with probability 1/4. What is the standard deviation of X?

Answer 69:
Let X be the number of heads. $X \sim \mathrm{Bin}(n = 20, p = 1/4)$

$$\begin{aligned}
\mathrm{Std}(X) &= \sqrt{np(1-p)} \\
&= \sqrt{20 \cdot 1/4 \cdot (1-1/4)} \\
&= 1.94
\end{aligned}$$

Question 70: You are giving a survey question where responses are "like" or "dislike" to 100 people. What is the probability that X is equal to 4? The number of likes can be represented by a random variable X. X is a Bin(100, 0.5).

Answer 70:
Let X be the number of likes. $X \sim \text{Bin}(n = 100, p = 0.5)$

$$\begin{aligned}
\text{P}(X = 4) &= \binom{n}{4} p^4 (1-p)^{n-4} \\
&= \binom{100}{4} 0.5^4 (1-0.5)^{100-4} \\
&< 0.00001
\end{aligned}$$

Question 71: You are flipping a coin. There are 20 independent coin-flips where the probability of a tail on a given coin-flip is 0.9. What is the standard deviation of tails?

Answer 71:
Let X be the number of tails. $X \sim \text{Bin}(n = 20, p = 0.9)$

$$\begin{aligned}
\text{Std}(X) &= \sqrt{np(1-p)} \\
&= \sqrt{20 \cdot 0.9 \cdot (1-0.9)} \\
&= 1.34
\end{aligned}$$

Question 72: You are flipping a coin. There are 50 independent coin-flips. The probability of a tail on a given coin-flip is 4/5. What is the expectation of heads?

Answer 72:
Let X be the number of heads. $X \sim \text{Bin}(n = 50, p = 1/5)$

$$\begin{aligned}
\text{E}[X] &= np \\
&= 50 \cdot 1/5 \\
&= 10
\end{aligned}$$

Question 73: You are giving a survey question where responses are "like" or "dislike" to 100 people. What is the standard deviation of likes? the probability of a dislike on each response is 41/50.

Answer 73:
Let X be the number of likes. $X \sim \text{Bin}(n = 100, p = 9/50)$

$$\begin{aligned}
\text{Std}(X) &= \sqrt{np(1-p)} \\
&= \sqrt{100 \cdot 9/50 \cdot (1-9/50)} \\
&= 3.84
\end{aligned}$$

Question 74: In a restaurant some customers ask for a water with their meal. 0.6 is the probability of a water requested by each customer and there are 30 independent customers. What is the expected number of waters requested?

Answer 74:
Let X be the number of waters requested. $X \sim \text{Bin}(n = 30, p = 0.6)$

$$\begin{aligned}
\text{E}[X] &= np \\
&= 30 \cdot 0.6 \\
&= 18.0
\end{aligned}$$

Question 75: There are 40 independent trials and 0.5 is the probability of a failure on each trial. What is the expectation of successes?

Answer 75:
Let X be the number of successes. $X \sim \text{Bin}(n = 40, p = 1/2)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 40 \cdot 1/2 \\ &= 20 \end{aligned}$$

Question 76: Imran is showing an online-ad to 30 people. 5/6 is the probability of an ad click on each ad shown. What is the standard deviation of ad clicks?

Answer 76:
Let X be the number of ad clicks. $X \sim \text{Bin}(n = 30, p = 5/6)$

$$\begin{aligned} \text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{30 \cdot 5/6 \cdot (1 - 5/6)} \\ &= 2.04 \end{aligned}$$

Question 77: You are running a server cluster. What is the probability that the number of crashes is 1? the server has 30 computers which crash independently and each server has a 1/3 probability of resulting in a crash.

Answer 77:
Let X be the number of crashes. $X \sim \text{Bin}(n = 30, p = 1/3)$

$$\begin{aligned} \text{P}(X = 1) &= \binom{n}{1} p^1 (1-p)^{n-1} \\ &= \binom{30}{1} 1/3^1 (1 - 1/3)^{30-1} \\ &= 0.00008 \end{aligned}$$

Question 78: Cody is running a server cluster with 40 computers. What is P(X <= 39)? The number of crashes can be represented by a random variable X. X is a Bin(n = 40, p = 3/4).

Answer 78:
Let X be the number of crashes. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned} \text{P}(X <= 39) &= 1 - \text{P}(40 <= X <= 40) \\ &= 1 - \binom{n}{40} p^4 0 (1-p)^{n-40} \end{aligned}$$

Question 79: You are hashing strings into a hashtable. 5/6 is the probability of a hash to the first bucket on each string hash. There are 30 independent string hashes. What is the probability that the number of hashes to the first bucket is greater than or equal to 29?

Answer 79:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 30, p = 5/6)$

$$\begin{aligned} \text{P}(X >= 29) &= P(29 <= X <= 30) \\ &= \sum_{i=29}^{30} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned}$$

Question 80: Irina is flipping a coin. Irina flips the coin 30 times and the probability of a head on each coin-flip is 0.4. What is the probability that the number of tails is 19?

Answer 80:
Let X be the number of tails. $X \sim \text{Bin}(n = 30, p = 0.6)$

$$\begin{aligned} \text{P}(X = 19) &= \binom{n}{19} p^{19} (1-p)^{n-19} \\ &= \binom{30}{19} 0.6^{19} (1-0.6)^{30-19} \\ &= 0.13962 \end{aligned}$$

Question 81: You are asking a survey question where responses are "like" or "dislike". The probability of a like on a given response is 1/2. You give the survey to 100 people. What is the probability that the number of likes is not less than 2?

Answer 81:
Let X be the number of likes. $X \sim \text{Bin}(n = 100, p = 1/2)$

$$\begin{aligned} \text{P}(X >= 2) &= 1 - \text{P}(0 <= X <= 1) \\ &= 1 - \sum_{i=0}^{1} \binom{n}{i} p^{i} (1-p)^{n-i} \end{aligned}$$

Question 82: Wind blows independently across locations. The number of independent locations is 100. The probability of wind at a given location is 3/20. What is the probability that the number of locations that have wind is 93?

Answer 82:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 100, p = 3/20)$

$$\begin{aligned} \text{P}(X = 93) &= \binom{n}{93} p^{93} (1-p)^{n-93} \\ &= \binom{100}{93} 3/20^{93} (1-3/20)^{100-93} \\ &< 0.00001 \end{aligned}$$

Question 83: You are flipping a coin. 0.9 is the probability of a tail on each coin-flip. You flip the coin 50 times. What is the expected number of heads?

Answer 83:
Let X be the number of heads. $X \sim \text{Bin}(n = 50, p = 0.1)$

$$\begin{aligned} \text{E}[X] &= np \\ &= 50 \cdot 0.1 \\ &= 5.0 \end{aligned}$$

Question 84: A machine learning algorithm makes binary predictions. What is the probability that the number of correct predictions is less than or equal to 0? the probability of a incorrect prediction on a given guess is 1/4. The number of independent guesses is 40.

Answer 84:
Let X be the number of correct predictions. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned} \text{P}(X <= 0) &= P(0 <= X <= 0) \\ &= \binom{n}{0} p^{0} (1-p)^{n-0} \end{aligned}$$

Question 85: Wind blows independently across 20 locations. 1/2 is the probability of wind at each location. What is the standard deviation of locations that have wind?

Answer 85:
Let X be the number of locations that have wind. $X \sim \text{Bin}(n = 20, p = 1/2)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{20 \cdot 1/2 \cdot (1 - 1/2)} \\ &= 2.24\end{aligned}$$

Question 86: 7/10 is the probability of a failure on each trial and the number of independent trials is 100. What is the probability that the number of successes is 7?

Answer 86:
Let X be the number of successes. $X \sim \text{Bin}(n = 100, p = 0.3)$

$$\begin{aligned}\text{P}(X = 7) &= \binom{n}{7} p^7 (1-p)^{n-7} \\ &= \binom{100}{7} 0.3^7 (1 - 0.3)^{100-7} \\ &< 0.00001\end{aligned}$$

Question 87: You generate random bit strings. What is the expectation of 1s? there are 100 independent bits and 0.1 is the probability of a 1 at each bit.

Answer 87:
Let X be the number of 1s. $X \sim \text{Bin}(n = 100, p = 0.1)$

$$\begin{aligned}\text{E}[X] &= np \\ &= 100 \cdot 0.1 \\ &= 10.0\end{aligned}$$

Question 88: You are testing a new medicine on patients. 3/5 is the probability of a cured patient on each trial. There are 30 independent trials. What is the probability that the number of cured patients is greater than or equal to 1?

Answer 88:
Let X be the number of cured patients. $X \sim \text{Bin}(n = 30, p = 3/5)$

$$\begin{aligned}\text{P}(X >= 1) &= 1 - \text{P}(0 <= X <= 0) \\ &= 1 - \binom{n}{0} p^0 (1-p)^{n-0}\end{aligned}$$

Question 89: A student is guessing randomly on an exam. 0.9 is the probability of a correct answer on each question and the test has 20 questions. What is the standard deviation of correct answers?

Answer 89:
Let X be the number of correct answers. $X \sim \text{Bin}(n = 20, p = 0.9)$

$$\begin{aligned}\text{Std}(X) &= \sqrt{np(1-p)} \\ &= \sqrt{20 \cdot 0.9 \cdot (1 - 0.9)} \\ &= 1.34\end{aligned}$$

Question 90: A student is guessing randomly on an exam with 40 questions. What is the probability that the number of correct answers is 32? 0.5 is the probability of a correct answer on each question.

Answer 90:
Let X be the number of correct answers. $X \sim \text{Bin}(n = 40, p = 0.5)$

$$\begin{aligned}
\text{P}(X = 32) &= \binom{n}{32} p^{32} (1-p)^{n-32} \\
&= \binom{40}{32} 0.5^{32} (1-0.5)^{40-32} \\
&= 0.00007
\end{aligned}$$

Question 91: In a restaurant some customers ask for a water with their meal. A random sample of 40 customers is selected where the probability of a water not requested by a given customer is 1/4. What is the standard deviation of waters requested?

Answer 91:
Let X be the number of waters requested. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned}
\text{Std}(X) &= \sqrt{np(1-p)} \\
&= \sqrt{40 \cdot 3/4 \cdot (1 - 3/4)} \\
&= 2.74
\end{aligned}$$

Question 92: A machine learning algorithm makes binary predictions. The number of correct predictions can be represented by a random variable X. X is a Bin(n = 30, p = 2/5). What is P(X < 27)?

Answer 92:
Let X be the number of correct predictions. $X \sim \text{Bin}(n = 30, p = 2/5)$

$$\begin{aligned}
\text{P}(X < 27) &= 1 - \text{P}(27 <= X <= 30) \\
&= 1 - \sum_{i=27}^{30} \binom{n}{i} p^{i} (1-p)^{n-i}
\end{aligned}$$

Question 93: Irina is flipping a coin. The probability of a tail on each coin-flip is 3/4. The number of independent coin-flips is 40. What is the probability that the number of tails is greater than 0?

Answer 93:
Let X be the number of tails. $X \sim \text{Bin}(n = 40, p = 3/4)$

$$\begin{aligned}
\text{P}(X > 0) &= 1 - \text{P}(0 <= X <= 0) \\
&= 1 - \binom{n}{0} p^{0} (1-p)^{n-0}
\end{aligned}$$

Question 94: Waddie is sending a stream of 50 bits to space. The probability of a no corruption on a given bit is 1/2. What is the expectation of corruptions?

Answer 94:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 50, p = 0.5)$

$$\begin{aligned}
\text{E}[X] &= np \\
&= 50 \cdot 0.5 \\
&= 25.0
\end{aligned}$$

Question 95: You are hashing strings into a hashtable. There are 30 independent string hashes where the probability of a hash to the first bucket on each string hash is 5/6. What is the probability that the number of hashes to the first bucket is 24?

Answer 95:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 30, p = 5/6)$

$$\begin{aligned} \text{P}(X = 24) &= \binom{n}{24} p^{24} (1-p)^{n-24} \\ &= \binom{30}{24} 5/6^{24} (1-5/6)^{30-24} \\ &= 0.16009 \end{aligned}$$

Question 96: Charlotte is hashing strings into a hashtable. 100 strings are hashed and the probability of a hash to the first bucket on a given string hash is 1/5. What is the probability that the number of hashes to the first bucket is greater than or equal to 1?

Answer 96:
Let X be the number of hashes to the first bucket. $X \sim \text{Bin}(n = 100, p = 1/5)$

$$\begin{aligned} \text{P}(X >= 1) &= 1 - \text{P}(0 <= X <= 0) \\ &= 1 - \binom{n}{0} p^{0} (1-p)^{n-0} \end{aligned}$$

Question 97: You are flipping a coin. Each coin-flip has a 3/10 probability of resulting in a head and there are 100 coin-flips. You can assume each coin-flip is independent. What is the probability that the number of heads is 0?

Answer 97:
Let X be the number of heads. $X \sim \text{Bin}(n = 100, p = 3/10)$

$$\begin{aligned} \text{P}(X = 0) &= \binom{n}{0} p^{0} (1-p)^{n-0} \\ &= \binom{100}{0} 3/10^{0} (1-3/10)^{100-0} \\ &< 0.00001 \end{aligned}$$

Question 98: Chris is sending a stream of 50 bits to space. 16/25 is the probability of a no corruption on each bit. What is the probability that the number of corruptions is greater than or equal to 47?

Answer 98:
Let X be the number of corruptions. $X \sim \text{Bin}(n = 50, p = 9/25)$

$$\begin{aligned} \text{P}(X >= 47) &= P(47 <= X <= 50) \\ &= \sum_{i=47}^{50} \binom{n}{i} p^{i} (1-p)^{n-i} \end{aligned}$$

Question 99: You are flipping a coin 30 times. What is the probability that the number of tails is less than 29? the probability of a tail on a given coin-flip is 2/3.

Answer 99:
Let X be the number of tails. $X \sim \text{Bin}(n = 30, p = 2/3)$

$$\begin{aligned} \text{P}(X < 29) &= 1 - \text{P}(29 <= X <= 30) \\ &= 1 - \sum_{i=29}^{30} \binom{n}{i} p^{i} (1-p)^{n-i} \end{aligned}$$

Question 100: You are manufacturing chips and are testing for defects. There are 40 independent tests. The probability of a non-defect on a given test is 5/8. What is the probability that the number of defects is 10?

Answer 100:

Let X be the number of defects. $X \sim \text{Bin}(n = 40, p = 3/8)$

$$\begin{aligned} \text{P}(X = 10) &= \binom{n}{10} p^{10} (1-p)^{n-10} \\ &= \binom{40}{10} 3/8^{10} (1-3/8)^{40-10} \\ &= 0.03507 \end{aligned}$$

Winning the Series

The Golden State Warriors are the basketball team for the Bay Area. The Warriors are going to play the Bucks (another professional basketball team) in a best of 7 series during the next NBA finals. They will win the series if you win at least 4 games. What is the probability that the warriors win the series? Each game is independent. Each game, the warriors have a 0.55 probability of winning.

This problem is equivalent to: Flip a biased coin 7 times (with a $p = 0.55$ probability of getting a heads). What is the probability of at least 4 heads?

Note: without loss of generality you could imagine that the two teams always play all 7 games, regardless of the outcome. Technically they stop playing after one team has achieved 4 wins, because the outcomes of the games no longer impact who wins. However, you could imagine that they continue.

What is the probability that the warriors win the series? Leave your answer to 3 decimal places

A critical step is to define a random variable and to recognize it is a Binomial. Let X be the number of games won. Since each game is independent, $X \sim \text{Bin}(n = 7, p = 0.55)$. The question is asking: $\text{P}(X \geq 4)$?

To answer this question, first recognize that:

$$\begin{aligned}\text{P}(X \geq 4) =& \text{P}(X = 4) + \text{P}(X = 5) \\ &+ \text{P}(X = 6) + \text{P}(X = 7)\end{aligned}$$

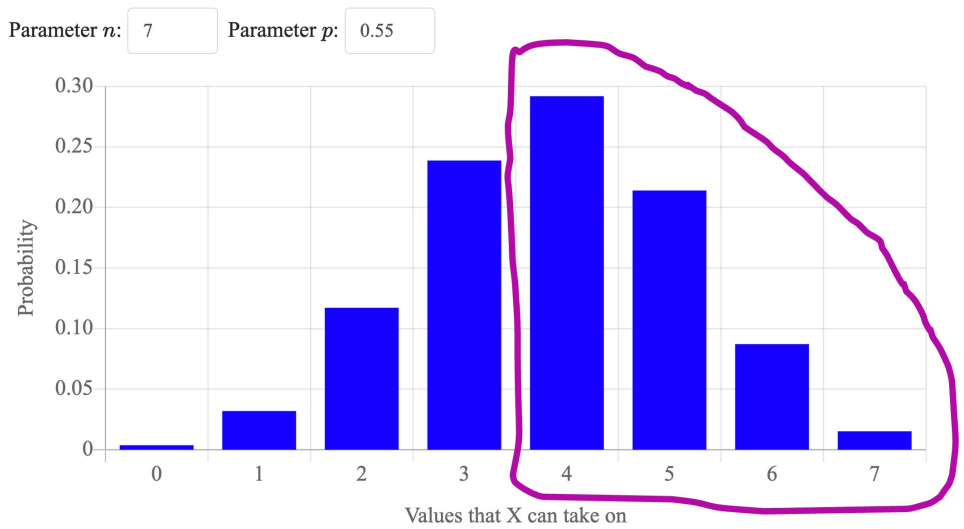
This is because the question is asking the probability of the "or" of each of the events on the right hand side of the equals sign. Since each of these events; $X = 4$, $X = 5$, etc are mutually exclusive, the probability of the "or" is simply the sum of the probabilities.

$$\begin{aligned}\text{P}(X \geq 4) &= \text{P}(X = 4) + \text{P}(X = 5) \\ &+ \text{P}(X = 6) + \text{P}(X = 7) \\ &= \sum_{i=4}^{7} P(X = i)\end{aligned}$$

Each of these probabilities are PMF questions:

$$
\begin{aligned}
\mathrm{P}(X \geq 4) &= \sum_{i=4}^{7} P(X = i) \\
&= \sum_{i=4}^{7} \binom{n}{i} p^i (1-p)^{n-i} \\
&= \sum_{i=4}^{7} \binom{7}{i} 0.55^i \cdot 0.45^{7-i}
\end{aligned}
$$

Here is that equation graphically. It represents the sum of these columns in the PMF:



At this point we have an equation that we can compute in order to find the answer. But how should we compute it? We could do it by hand! Or using a calculator. Or, we can use python, and specifically the scipy package:

```python
from scipy import stats

pr = 0
# calculate the sum
for i in range(4, 8):
   # this for loop gives i in [4,5,6,7]
   pr_i = stats.binom.pmf(i, n = 7, p = 0.55)
   pr += pr_i

print(f'P(X >= 4) = {pr}')
```

Which produces the correct answer:

```
P(X >= 4) = 0.6082877968750001
```

Buggy Solution

A good reason to study this problem is because of this common misconception for how to compute $P(X \geq 4)$. It is worth understanding why it is wrong.

Incorrectly trying to recreate the binomial

Similar to how we defined a binomial distribution PMF equation (see: Many Coin Flips) we can construct outcomes of the seven game series where the warriors win.

We are going to choose 4 slots where the Warriors win, and we don't care about the rest. They could either be wins or losses. Out of 7 games, select 4 where the Warriors win. There are $\binom{7}{4}$ ways to do so. The probability of each particular selection of four games to win is p^4 because we need them to win those four games, and we don't care about the rest. As such the probability is:

$$
P(X \geq 4) = \binom{7}{4} p^4
$$

This idea seems good, but it doesn't work. First of all, we can recognize that there is a problem by considering the outcome if we set $p = 1.0$. In this case $P(X \geq 4) = \binom{7}{4}p^4 = \binom{7}{4}1^4 = 35$. Clearly 35 is an invalid probability (which is much greater than 1). As such this can't be the right answer.

But what is wrong with this approach? Lets enumerate the 35 different outcomes that they are considering. Let B = we don't know who wins. Let W = the warriors win. Here each outcome is the assignment to each of the 7 games in the series:

```
(B, B, B, W, W, W, W)
(B, B, W, B, W, W, W)
(B, B, W, W, B, W, W)
(B, B, W, W, W, B, W)
(B, B, W, W, W, W, B)
(B, W, B, B, W, W, W)
(B, W, B, W, B, W, W)
(B, W, B, W, W, B, W)
(B, W, B, W, W, W, B)
(B, W, W, B, B, W, W)
(B, W, W, B, W, B, W)
(B, W, W, B, W, W, B)
(B, W, W, W, B, B, W)
(B, W, W, W, B, W, B)
(B, W, W, W, W, B, B)
(W, B, B, B, W, W, W)
(W, B, B, W, B, W, W)
(W, B, B, W, W, B, W)
(W, B, B, W, W, W, B)
(W, B, W, B, B, W, W)
(W, B, W, B, W, B, W)
(W, B, W, B, W, W, B)
(W, B, W, W, B, B, W)
(W, B, W, W, B, W, B)
(W, B, W, W, W, B, B)
(W, W, B, B, B, W, W)
(W, W, B, B, W, B, W)
(W, W, B, B, W, W, B)
(W, W, B, W, B, B, W)
(W, W, B, W, B, W, B)
(W, W, B, W, W, B, B)
(W, W, W, B, B, B, W)
(W, W, W, B, B, W, B)
(W, W, W, B, W, B, B)
(W, W, W, W, B, B, B)
```

It is in fact the case that the probability of any of these 35 outcomes is p^4. For example: (W, W, W, W, B, B, B). The warriors need to win the first 4 independent games: p^4. Then there are three events where either team could win. The probability of "B", that either team could win, is 1. That makes sense. Either the warriors win or the other team wins. As such our probability of any given outcome, aka row in the set of outcomes above, is: $p^4 \cdot 1^3 = p^4$

The bug here is that these outcomes are **not** mutually exclusive, yet the answer treats them as such. In the many coin flips example, we constructed outcomes in the format (T, T, T, H, H, T, H). These outcomes are in fact mutually exclusive. Its not possible for two distinct lists of outcomes to simultaneously exist. On the other hand in the version where "B" stands for either team could win, the outcomes do have overlap. For example consider the two rows from the examples above:

```
(B, W, W, W, B, B, W)
(B, W, W, W, B, W, B)
```

These could both occur (and hence are not mutually exclusive). For example if the warriors win all 7 games! Or the warriors win all games except for games 1 and 5. Both events are satisfied. Because the events are not mutually exclusive, if we want the probability of the "or" of each of these events we can

not just sum the probabilities of each of the events (and that is exactly what $P(X \geq 4) = \binom{7}{4}p^4$ implies. Instead you would need to use inclusion exclusion for the or of 35 events (yikes!). Alternatively see the answer we propose above.

Jury Selection

In the Supreme Court case: Berghuis v. Smith, the Supreme Court (of the US) discussed the question: "If a group is underrepresented in a jury pool, how do you tell?"

Justice Breyer [Stanford Alum] opened the questioning by invoking the binomial theorem. He hypothesized a scenario involving “an urn with a thousand balls, and sixty are red, and nine hundred forty are green, and then you select them at random… twelve at a time.” According to Justice Breyer and the binomial theorem, if the red balls were black jurors then “you would expect… something like a third to a half of juries would have at least one black person” on them.

Note: What is missing in this conversation is the power of diverse backgrounds when making difficult decisions.

Simulate

Simulation:

Explanation:

Technically, since jurors are selected without replacement, you should represent the number of under-representative jurors as being a Hyper Geometric Random Variable (a random variable we don't look at explicitely in CS109) st

$$X \sim \text{HypGeo}(n = 12, N = 1000, m = 60)$$

$$\begin{aligned} P(X \geq 1) &= 1 - P(X = 0) \\ &= 1 - \frac{\binom{60}{0}\binom{940}{12}}{\binom{1000}{12}} \\ &\approx 0.5261 \end{aligned}$$

However Justic Breyer made his case by citing a Binomial distribution. This isn't a perfect use of binomial, because the binomial assumes that each experiment has equal likelihood (p) of success. Because the jurors are selected without replacement, the probability of getting a minority juror changes slightly after each selection (and depending on what the selection was). However, as we will see, because the probabilities don't change too much the binomial distribution is not too far off.

$$X \sim \text{Binomial}(n = 12, p = 60/1000)$$

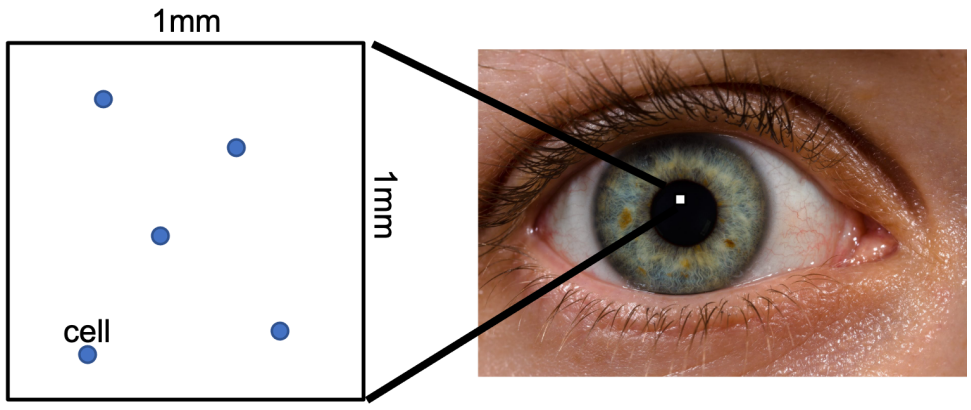
$$\begin{aligned} P(X \geq 1) &= 1 - P(X = 0) \\ &= 1 - \binom{60}{0}(1 - 0.06)^{12} \\ &\approx 0.5241 \end{aligned}$$

Acknowledgements: Problem posed and solved by Mehran Sahami

Grading Eye Inflammation

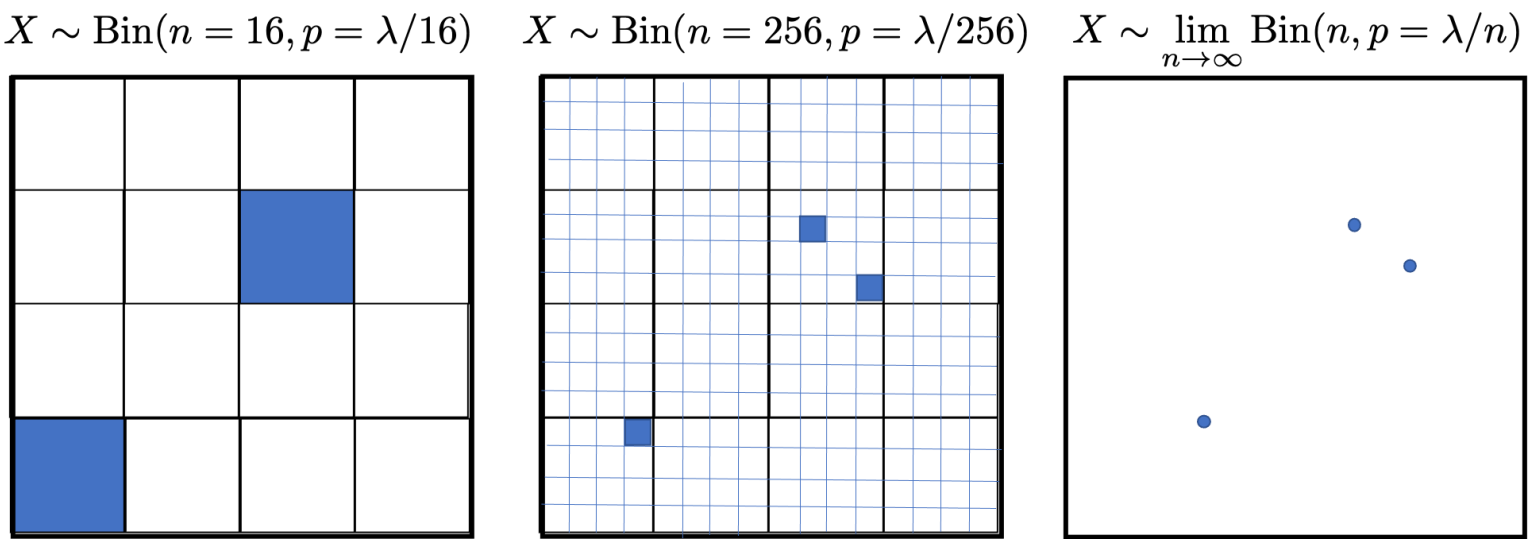
When a patient has eye inflammation, eye doctors "grade" the inflammation. When "grading" inflammation they randomly look at a single 1 millimeter by 1 millimeter square in the patient's eye and count how many "cells" they see.

There is uncertainty in these counts. If the true average number of cells for a given patient's eye is 6, the doctor could get a different count (say 4, or 5, or 7) just by chance. As of 2021, modern eye medicine does not have a sense of uncertainty for their inflammation grades! In this problem we are going to change that. At the same time we are going to learn about Poisson distributions over space.



Why is the number of cells observed in a 1x1 square governed by a Poisson process?

We can approximate a distribution for the count by discretizing the square into a fixed number of equal sized buckets. Each bucket either has a cell or not. Therefore, the count of cells in the 1x1 square is a sum of Bernoulli random variables with equal p, and as such can be modeled as a binomial random variable. This is an approximation because it doesn't allow for two cells in one bucket. Just like with time, if we make the size of each bucket infinitely small, this limitation goes away and we converge on the true distribution of counts. The binomial in the limit, i.e. a binomial as $n \to \infty$, is truly represented by a Poisson random variable. In this context, λ represents the average number of cells per 1×1 sample. See Figure 2.



For a given patient the true average rate of cells is 5 cells per 1x1 sample. What is the probability that in a single 1x1 sample the doctor counts 4 cells?

Let X denote the number of cells in the 1x1 sample. We note that $X \sim \text{Poi}(5)$. We want to find $P(X = 4)$.

$$P(X = 4) = \frac{5^4 e^{-5}}{4!} \approx 0.175$$

Multiple Observations

Heads up! This section uses concepts from Part 3. Specifically Independence in Variables

For a given patient the true average rate of cells is 5 cells per 1mm by 1mm sample. In an attempt to be more precise, the doctor counts cells in **two** different, larger **2mm by 2mm** samples. Assume that the occurrences of cells in one 2mm by 2mm samples are independent of the occurrences in any other 2mm by 2mm samples. What is the probability that she counts 20 cells in the first samples and 20 cells in the second?

Let Y_1 and Y_2 denote the number of cells in each of the 2x2 samples. Since there are 5 cells in a 1x1 sample, there are 20 samples in a 2x2 sample since the area quadrupled, so we have that $Y_1 \sim \text{Poi}(20)$ and $Y_2 \sim \text{Poi}(20)$. We want to find $P(Y_1 = 20 \wedge Y_2 = 20)$. Since the number of cells in the two samples are independent, this is equivalent to finding $\text{P}(Y_1 = 20)\,\text{P}(Y_2 = 20)$.

Estimating Lambda

Heads up! This section uses concepts from Part 5. Specifically Maximum A Posteriori

Inflammation prior: Based on millions of historical patients, doctors have learned that the prior probability density function of true rate of cells is:

$$f(\lambda) = K \cdot \lambda \cdot e^{-\frac{\lambda}{2}}$$

Where K is a normalization constant and λ must be greater than 0.

A doctor takes a single sample and counts 4 cells. Give an equation for the updated probability density of λ. Use the "Inflammation prior" as the prior probability density over values of λ. Your probability density may have a constant term.

Let θ be the random variable for true rate. Let X be the random variable for the count

$$\begin{aligned} f(\theta = \lambda | X = 4) &= \frac{P(X = 4 | \theta = \lambda) f(\theta = \lambda)}{P(X = 4)} \\ &= \frac{\frac{\lambda^4 e^{-\lambda}}{4!} \cdot K \cdot \lambda \cdot e^{\lambda/2}}{P(X = 4)} \\ &= \frac{K \cdot \lambda^5 e^{-\frac{3}{2}\lambda}}{4! P(X = 4)} \end{aligned}$$

A doctor takes a single sample and counts 4 cells. What is the Maximum A Posteriori estimate of λ?

Maximize the "posterior" of the parameter calculated in the previous section:

$$\arg\max_{\lambda} \frac{K \cdot \lambda^5 e^{-\frac{3}{2}\lambda}}{4! P(X = 4)} = \arg\max_{\lambda} \lambda^5 e^{-\frac{3}{2}\lambda}$$

Take logarithm (preserves argmax, and easier derivative):

$$\begin{aligned} &= \arg\max_{\lambda} \log\left(\lambda^5 e^{-\frac{3}{2}\lambda}\right) \\ &= \arg\max_{\lambda} \left(5 \log \lambda - \frac{3}{2}\lambda\right) \end{aligned}$$

Calculate the derivative with respect to the parameter, and set equal to 0

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \lambda}\left(5\log\lambda - \frac{3}{2}\lambda\right) \\
0 &= \frac{5}{\lambda} - \frac{3}{2} \\
\lambda &= \frac{10}{3}
\end{aligned}
$$

Explain, in words, the difference between the two estimates of lambda in the two previous parts.

The estimate in the first part is a ``distribution" (also called a soft estimate) whereas the estimate in the second part is a single value (also called a point estimate). The former contains information about confidence.

What is the MLE estimate of λ?

The MLE estimate doesn't use the prior belief. The MLE estimate for a poisson is simply the average of the observations. In this case the average of our single observation is 4. MLE is not a great tool for estimating our parameter from just one datapoint.

A patient comes on two separate days. The first day the doctor counts 5 cells, the second day the doctor counts 4 cells. Based only on this observation, and treating the true rates on the two days as independent, what is the probability that the patient's inflammation has gotten better (in other words, that their λ has decreased)?

Let θ_1 be the random variable for lambda on the first day and θ_2 be the random variable for lambda on the second day.

$$
\begin{aligned}
f(\theta_1 = \lambda | X = 5) &= K_1 \cdot \lambda^6 e^{-\frac{3}{2}\lambda} \\
f(\theta_2 = \lambda | X = 4) &= K_2 \cdot \lambda^5 e^{-\frac{3}{2}\lambda}
\end{aligned}
$$

The question is asking what is $P(\theta_1 > \theta_2)$? There are a few ways to calculate this exactly:

$$
\begin{aligned}
&\int_{\lambda_1=0}^{\infty}\int_{\lambda_2=0}^{\lambda_1} f(\theta_1 = \lambda_1, \theta_2 = \lambda_2) \\
&= \int_{\lambda_1=0}^{\infty}\int_{\lambda_2=0}^{\lambda_1} f(\theta_1 = \lambda_1) \cdot f(\theta_2 = \lambda_2) \\
&= \int_{\lambda_1=0}^{\infty} f(\theta_1 = \lambda_1) \int_{\lambda_2=0}^{\lambda_1} f(\theta_2 = \lambda_2) \\
&= \int_{\lambda_1=0}^{\infty} K_1 \cdot \lambda^6 e^{-\frac{3}{2}\lambda} \int_{\lambda_2=0}^{\lambda_1} K_2 \cdot \lambda^5 e^{-\frac{3}{2}\lambda}
\end{aligned}
$$

Grades are Not Normal

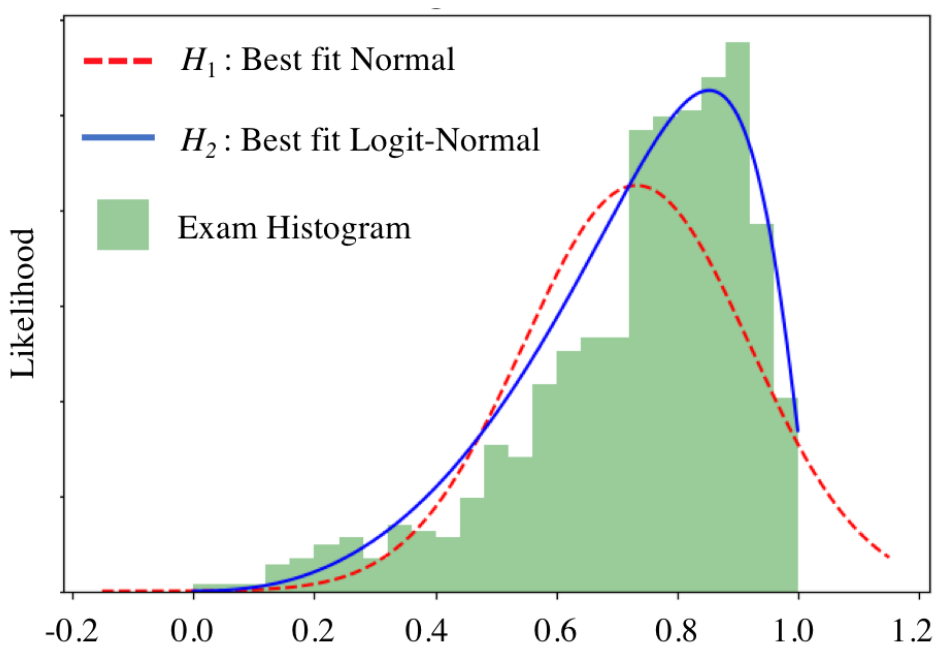
Sometimes you just feel like squashing normals:

Logit Normal
The logit normal is the continuous distribution that results from applying a special "squashing" function to a Normally distributed random variable. The squashing function maps all values the normal could take on onto the range 0 to 1. If $X \sim \text{LogitNormal}(\mu, \sigma^2)$ it has:

$$\begin{aligned}
\text{PDF:} \qquad & f_X(x) = \begin{cases} \frac{1}{\sigma(\sqrt{2\pi})x(1-x)} e^{-\frac{(\text{logit}(x)-\mu)^2}{2\sigma^2}} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \\
\text{CDF:} \qquad & F_X(x) = \Phi\Big(\frac{\text{logit}(x) - \mu}{\sigma}\Big) \\
\text{Where:} \qquad & \text{logit}(x) = \log\Big(\frac{x}{1-x}\Big)
\end{aligned}$$

A new theory shows that the Logit Normal better fits exam score distributions than the traditionally used Normal. Let's test it out! We have some set of exam scores for a test with min possible score 0 and max possible score 1, and we are trying to decide between two hypotheses:

H_1: our grade scores are distributed according to $X \sim \text{Normal}(\mu = 0.7, \sigma^2 = 0.2^2)$.
H_2: our grade scores are distributed according to $X \sim \text{LogitNormal}(\mu = 1.0, \sigma^2 = 0.9^2)$.



Under the normal assumption, H_1, what is $P(0.9 < X < 1.0)$? Provide a numerical answer to two decimal places.

$$P(0.9 < X < 1.0) = \Phi\left(\frac{1.0 - 0.7}{0.2}\right) - \Phi\left(\frac{0.9 - 0.7}{0.2}\right) = \Phi(1.5) - \Phi(1.0) = 0.9332 - 0.8413 = 0.09$$

Under the logit-normal assumption, H_2, what is $P(0.9 < X < 1.0)$?

$$F_X(1.0) - F_X(0.9) = \Phi\Big(\frac{\text{logit}(1.0) - 1.0}{0.9}\Big) - \Phi\Big(\frac{\text{logit}(0.9) - 1.0}{0.9}\Big)$$

Which we can solve for numerically:

$$\Phi\Big(\frac{\text{logit}(1.0) - 1.0}{0.9}\Big) - \Phi\Big(\frac{\text{logit}(0.9) - 1.0}{0.9}\Big) = 1 - \Phi(1.33) \approx 0.91$$

Under the normal assumption, H_1, what is the maximum value that X can take on?

$$\infty$$

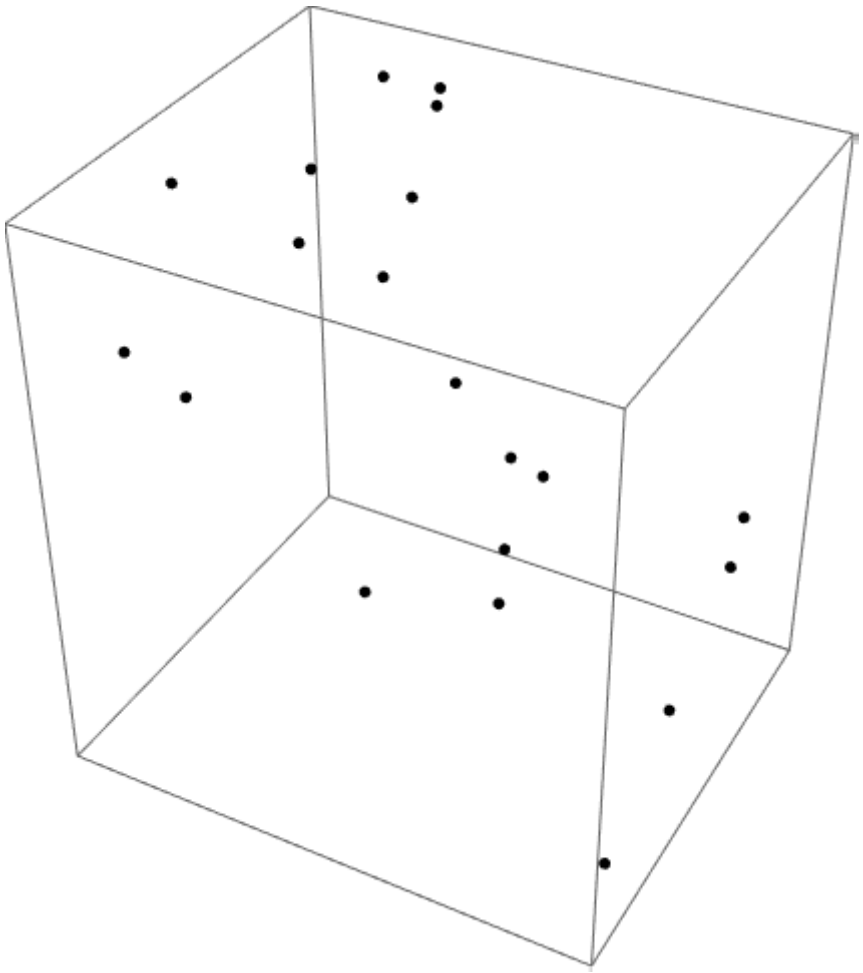
Before observing any test scores, you assume that (a) one of your two hypotheses is correct and (b) that initially, each hypothesis is equally likely to be correct, $P(H_1) = P(H_2) = \frac{1}{2}$. You then observe a single test score, $X = 0.9$. What is your updated probability that the Logit-Normal hypothesis is correct?

$$\begin{aligned} P(H_2|X=0.9) &= \frac{f(X=0.9|H_2)P(H_2)}{f(X=0.9|H_2)P(H_2) + f(X=0.9|H_1)P(H_1)} \\ &= \frac{f(X=0.9|H_2)}{f(X=0.9|H_2) + f(X=0.9|H_1)} \\ &= \frac{\frac{1}{\sigma(\sqrt{2\pi})0.9*(1-0.9)}e^{-\frac{(\text{logit}(0.9)-1.0)^2}{2*0.9^2}}}{\frac{1}{\sigma(\sqrt{2\pi})0.9*(1-0.9)}e^{-\frac{(\text{logit}(0.9)-1.0)^2}{2*0.9^2}} + \frac{1}{0.2\sqrt{2\pi}}e^{-\frac{(0.9-0.7)^2}{2*0.2^2}}} \end{aligned}$$

Curse of Dimensionality

In machine learning, like many fields of computer science, often involves high dimensional points, and high dimension spaces have some surprising probabilistic properties.

A random *value* X_i is a Uni(0, 1).
A random *point* of dimension d is a list of d random values: $[X_1 \dots X_d]$.



A random *value* X_i is close to an edge if X_i is less than 0.01 **or** X_i is greater than 0.99. What is the probability that a random value is close to an edge?

Let E be the event that a random value is close to an edge.
$P(E) = P(X_i < 0.01) + P(X_i > 0.99) = 0.02$

A random *point* $[X_1, X_2, X_3]$ of dimension 3 is close to an edge if *any* of its values are close to an edge. What is the probability that a 3 dimensional point is close to an edge?

The event is equivalent to the complement of none of the dimensions of the point is close to an edge, which is: $1 - (1 - P(E))^3 = 1 - 0.98^3 \approx 0.058$

A random *point* $[X_1, \dots X_{100}]$ of dimension 100 is close to an edge if *any* of its values are close to an edge. What is the probability that a 100 dimensional point is close to an edge?

Similarly, it is: $1 - (1 - P(E))^{100} = 1 - 0.98^{100} \approx 0.867$

There are many other phenomena of high dimensional points: such as, the euclidean distance between points starts to converge.

Probability and Babies

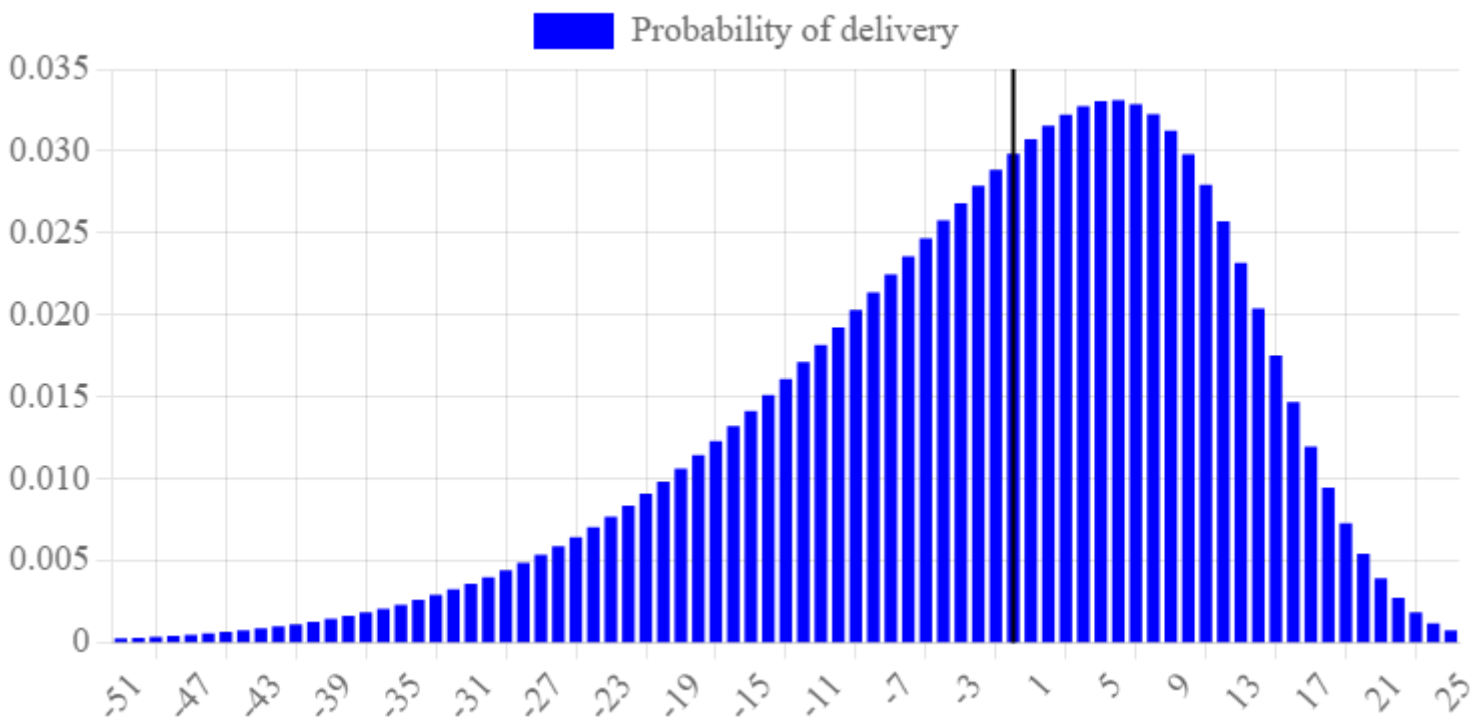
This demo used to be live. We now know that the delivery happened on Jan 23rd. Lets go back in time to Jan 1st and see what the probability looked like at that point.

What is the probability that Laura gives birth today (given that she hasn't given birth up until today)?

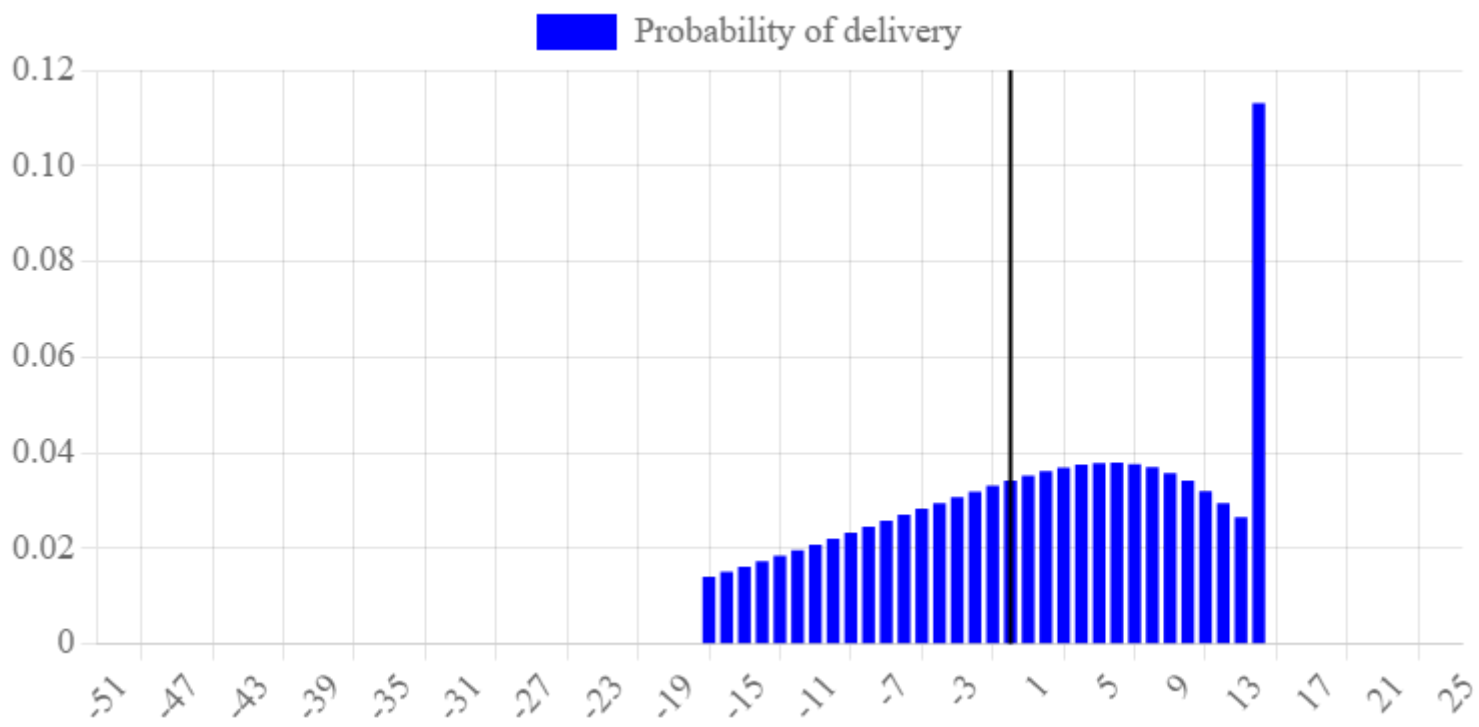
Today's Date	1/Jan/2021
Due Date	18/Jan/2021

Probability of delivery today: **0.014**
Probability of delivery in next 7 days: **0.144**
Current days past due date: **-17** days
Unconditioned probability mass before today: **0.128**

How likely is delivery, in humans, relative to the due date? There have been millions of births which gives us a relatively good picture [1]. The length of human pregnancy varies by quite a lot! Have you heard that it is 9 months? That is a rough, point estimate. The mean duration of pregnancy is 278.6 days, and pregnancy length has a standard deviation (SD) of 12.5 days. This distribution is not normal, but roughly matches a "skewed normal". This is a general probability mass function for the first pregnancy collected from hundreds of thousands of women (this PMF is very similar across demographics, but changes based on whether the woman has given birth before):



Of course, we have more information. Specifically, we know that Laura **hasn't** given birth up until today (we will update this example when that changes). We also know that babies which are over 14 days late are "induced" on day 14. How likely is delivery given that we haven't delivered up until today? Note that the y-axis is scalled differently:



Implementation notes: this calculation was performed by storing the PDF as a list of (day, probability) points. These values are sometimes called weighted samples, or "particles" and are the key component to a "particle filtering" approach. After we observe no-delivery, we set the probability of every point which has a day before today to be 0, and then re-normalize the remaining points (aka we "filter" the "particles"). This is convenient because the "posterior" belief doesn't follow a simple equation -- using particles means we never have to write that equation down in our code.

Three friends have the exact same due date (Really! this isn't a hypothetical) What is the probability that all three couples deliver on the exact same day?

Probability of three couples on the same day: **0.002**

How did we get that number? Let p_i be the probability that one baby is delivered on day i -- this number can be read off the probability mass function. Let D_i be the event that all three babies are delivered on day i. Note that the event D_i is mutually exclusive with the event that all three babies are born on another day (So for example, D_1 is mutually exclusive with D_2, D_3 etc). Let $N = 3$ be the event that all babies are born on the same day:

$$\begin{aligned} \mathrm{P}(N=3) &= \sum_i \mathrm{P}(D_i) && \text{Since days are mutually exclusive} \\ &= \sum_i p_i^3 && \text{Since the three couples are independent} \end{aligned}$$

[1] Predicting delivery date by ultrasound and last menstrual period in early gestation

Acknowledgements: This problem was first posed to me by Chris Gregg.

Joint Probability

Many interesting problems involve not one random variable, but rather several interacting with one another. In order to create interesting probabilistic *models* and to reason in real world situations, we are going to need to learn how to consider several random variables jointly.

In this section we are going to use disease prediction as a working example to introduce you to the concepts involved in probabilistic models. The general question is: a person has a set of observed symptoms. Given the symptoms what is the probability over each possible disease?

We have already considered events that co-occur and covered concepts such as independence and conditional probability. What is new about this section is (1) we are going to cover how to handle *random variables* which co-occur and (2) we are going to talk about how computers can reason under large probabilistic models.

Joint Probability Functions

For single random variables, the most important information was the PMF or, if the variable was continuous, the PDF. When dealing with two or more variables, the equivalent function is called the Joint function. For discrete random variables, it is a function which takes in a value for each variable and returns the probability (or probability density for continuous variables) that each variable takes on its value. For example if you had two *discrete* variables the Joint function is:

$$\mathrm{P}(X = x, Y = y) \qquad \text{Joint function for } X \text{ and } Y$$

You should read the comma as an "and" and as such this is saying the probability that $X = x$ and $Y = y$. Again like for single variables, as shorthand, we often write just the values and it implies that we are talking about the probability of the random variables taking on those values. This notation is convenient because it is shorter, and it makes it explicit that the function is operating over two parameters. It requires to recall that the event is a *random variable* taking on the given value.

$$\mathrm{P}(x, y) \qquad \text{Shorthand for } \mathrm{P}(X = x, Y = y)$$

If any of the variables are continuous we use different notation to make it clear that we need a probability density function, something we can integrate over to get a probability. We will cover this in detail:

$$f(X = x, Y = y) \qquad \text{Joint density function if } X \text{ or } Y \text{ are continuous}$$

The same idea extends to as many variables as you have in your model. For example if you had three discrete random variables X, Y, and Z, the joint probability function would state the likelihood of an assignment to all three: $\mathrm{P}(X = x, Y = y, Z = z)$.

Joint Probability Tables

> ***Definition***: Joint Probability Table
> A joint probability table is a way of specifying the "joint" distribution between multiple random variables. It does so by keeping a multi-dimensional lookup table (one dimension per variable) so that the probability mass of any assignment, eg $\mathrm{P}(X = x, Y = y, \ldots)$, can be directly looked up.

Let us start with an example. In 2020 the Covid-19 pandemic disrupted lives around the world. Many people were unable to get tested and had to determine whether or not they were sick based on home diagnosis. Let's build a very simple probabilistic model to enable us to make a tool which can predict the

probability of having the illness given observed symptoms. To make it clear that this is a pedagogical example, let's consider a made up illness called Determinitis. The two main symptoms are fever and loss of smell.

Variable	Symbol	Type
Has Determinitis	D	Bernoulli (1 indicates has Determinitis)
Fever	F	Categorical (none, low, high)
Can Smell	S	Bernoulli (1 indicates can smell)

A joint probability table is a brute force way to store the probability mass of a particular assignment of values to our variables. Here is a probabilistic model for our three random variables (aside: the values in this joint are realistic and based on reasearch, but are primarily for teaching. Consult a doctor before making medical decisions).

	$D = 0$	
	$S = 0$	$S = 1$
$F = \text{none}$	0.024	0.783
$F = \text{low}$	0.003	0.092
$F = \text{high}$	0.001	0.046

	$D = 1$	
	$S = 0$	$S = 1$
$F = \text{none}$	0.006	0.014
$F = \text{low}$	0.005	0.011
$F = \text{high}$	0.004	0.011

A few key observations:

- Each cell in this table represents the probability of one assignment of variables. For example the probability that someone can't smell, $S = 0$, has a low fever, $F = \text{low}$, and has the illness, $D = 1$, can be directly read off the table: $P(D = 1, S = 0, F = \text{low}) = 0.005$.
- These are joint probabilities *not* conditional probabilities. The value 0.005 is the value of illness, no smell and low fever. It is not the probability of no smell and low fever given illness. A table which stored conditional probabilities would be called a conditional probability table, this is a joint probability table.
- If you sum over all cells, the total will be 1. Each cell is a mutually exclusive combination of events and the cells are meant to span the entire space of possible outcomes.
- This table is large! We can count the number of cells using the step rule of counting. If n_i is the number of different values that random variable i can take on, the number of cells in the joint table is $\prod_i n_i$.

Properties of Joint Distributions

There are many properties of a random variable of any joint distribution some of which we will dive into extensively. Here is a brief summary. Each random variable has:

Property	Notation Example	Description
Distribution Function (PMF or PDF)	$\text{P}(X = x, Y = y, \ldots)$ or $f(X = x, Y = y, \ldots)$	A function which maps values the RV can take on to likelihood.
Cumulative Distribution Function (CDF)	$F(X < x, Y < y, \ldots)$	Probability that each variable is less than its corresponding parameter

Property	Notation Example	Description
Covariance	$\sigma_{X,Y}$	A measure of how much two random variables vary together.
Correlation	$\rho_{X,Y}$	Normalized co-variance

Multinomial

The multinomial is an example of a *parametric* distribution for multiple random variables.

Say you perform n independent trials of an experiment where each trial results in one of m outcomes, with respective probabilities: $p_1, p_2, \ldots, p_m$ (constrained so that $\sum_i p_i = 1$). Define X_i to be the number of trials with outcome i. A multinomial distribution is a closed form function that answers the question: What is the probability that there are c_i trials with outcome i. Mathematically:

$$P(X_1 = c_1, X_2 = c_2, \ldots, X_m = c_m) = \binom{n}{c_1, c_2, \ldots, c_m} \cdot p_1^{c_1} \cdot p_2^{c_2} \ldots p_m^{c_m}$$

Often people will use the product notation to write the exact same equation:

$$P(X_1 = c_1, X_2 = c_2, \ldots, X_m = c_m) = \binom{n}{c_1, c_2, \ldots, c_m} \cdot \prod_i p_i^{c_i}$$

Example: A 6-sided die is rolled 7 times. What is the probability that you roll: 1 one, 1 two, 0 threes, 2 fours, 0 fives, 3 sixes (disregarding order).

$$\begin{aligned} P(X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 2, X_5 = 0, X_6 = 3) \\ &= \frac{7!}{2!3!}\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^2\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^3 \\ &= 420\left(\frac{1}{6}\right)^7 \end{aligned}$$

The multinomial is especially popular because of its use as a model of language. For a full example see the Federalist Paper Authorship example.

Continuous Joint

Random variables X and Y are Jointly Continuous if there exists a joint Probability Density Function (PDF) f such that:

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = \int_{a_1}^{a_2} \int_{b_1}^{b_2} f(X = x, Y = y)\, dy\ dx$$

Using the PDF we can compute marginal probability densities:

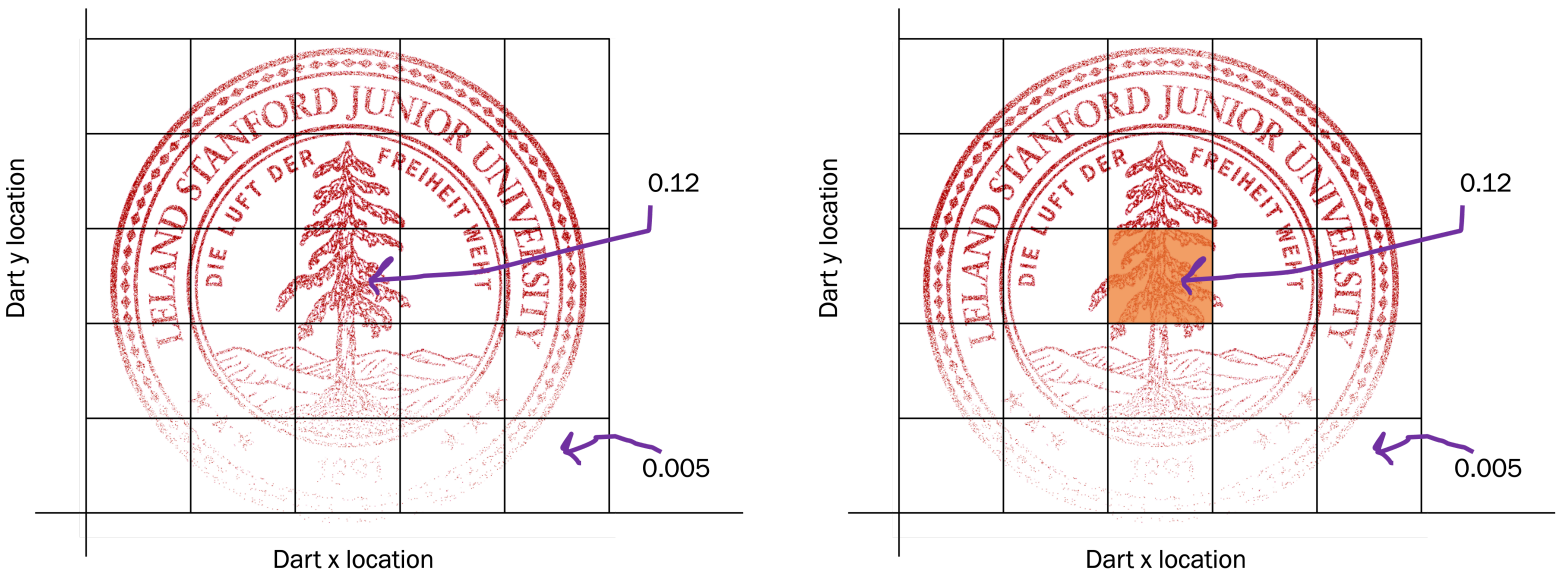
$$f(X = a) = \int_{-\infty}^{\infty} f(X = a, Y = y)\, dy$$
$$f(Y = b) = \int_{-\infty}^{\infty} f(X = x, Y = b)\, dx$$

Let $F(x, y)$ be the Cumulative Density Function (CDF):

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = F(a_2, b_2) - F(a_1, b_2) + F(a_1, b_1) - F(a_2, b_1)$$

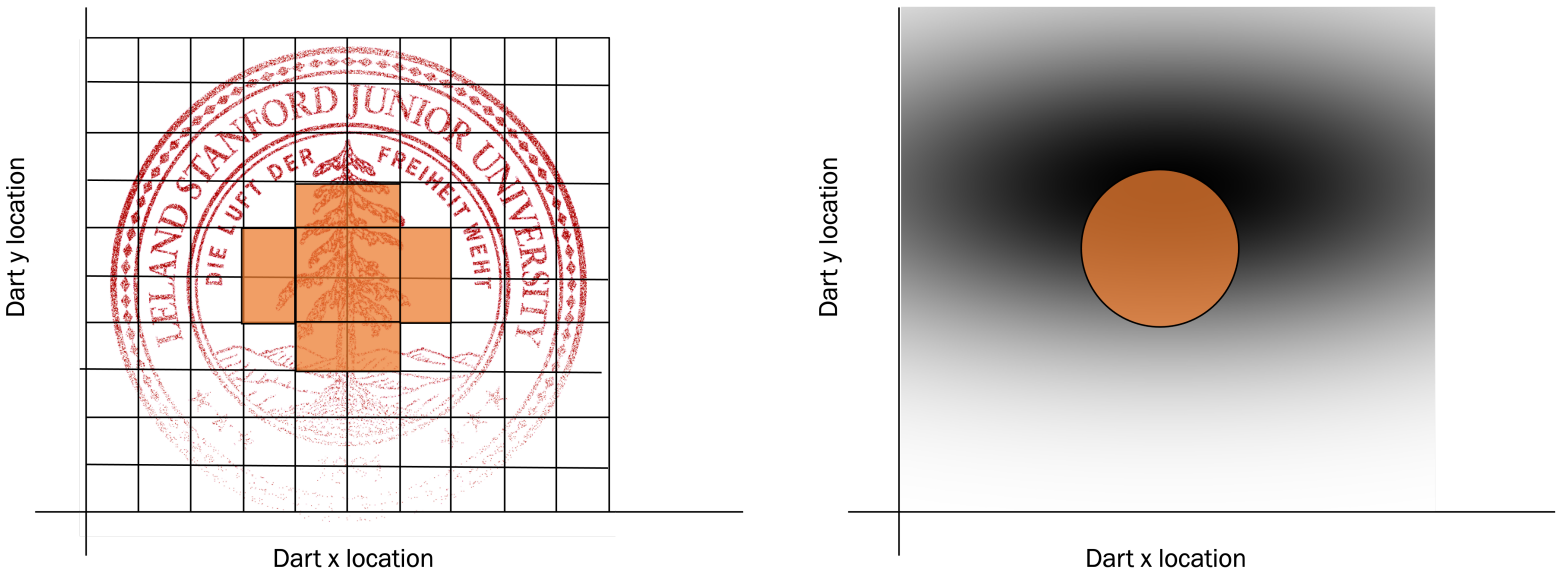
From Discrete Joint to Continuous Joint

Thinking about multiple continuous random variables jointly can be unintuitive at first blush. But we can turn to our helpful trick that we can use to understand continuous random variables: start with a discrete approximation. Consider the example of creating the CS109 seal. It was generated by throwing half a million darts at an image of the stanford logo (keeping all the pixels that get hit by at least one dart). The darts could hit any continuous location on the logo, and, the locations are not equally likely. Instead, the location a dart hits is goverened by a joint continuous distribution. In this case there are only two simultaneous random variables, the x location of the dart and the y location of the dart. Each random variable is continuous (it takes on real numbers). Thinking about the joint probability density function is easier by first considering a discretization. I am going to break the dart landing area into 25 discrete buckets:

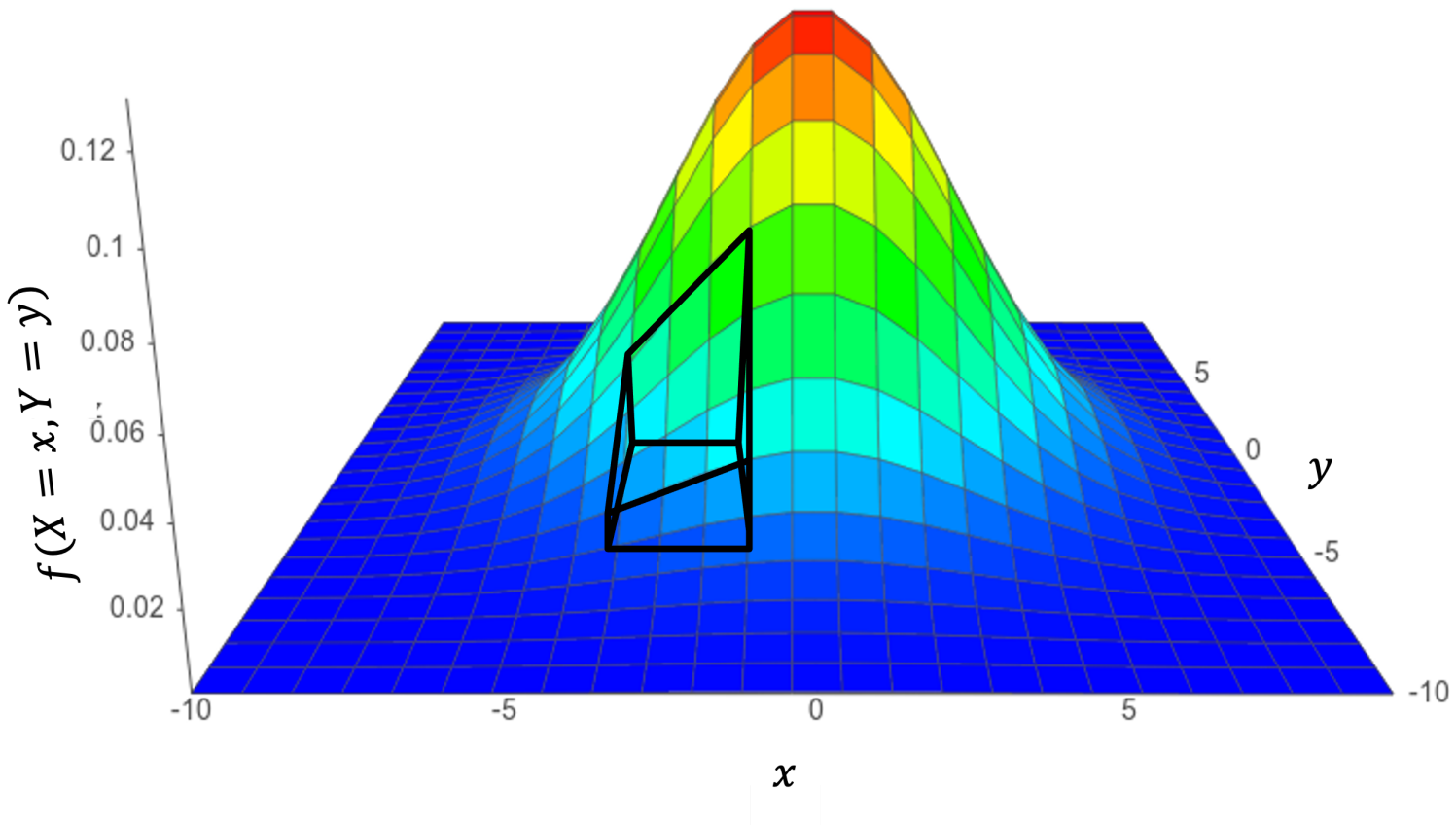


On the left is a visualization of the probability mass of this joint distribution, and on the right is a visualization of how we could answer the question: what is the probability that a dart hits within a certain distance of the center. For each bucket there is a single number, the probability that a dart will fall into that particular bucket (these probabilities are mutually exclusive and sum to 1).

Of course this discretization only *approximates* the joint probability distribution. In order to get a better approximation we could create more fine-grained discretizations. In the limit we can make our buckets infinitely small, and the value associated with each bucket becomes a second derivative of probability.



To represent the 2D probability density in a graph, we use the darkness of a value to represent the density (darker means more density). Another way to visualize this distribution is from an angle. This makes it easier to realize that this is a function with two inputs and one output. Below is an different visualization of the exact same density function:



Just like in the single random variable case, we are now represending our belief in the continuous random variables as *densities* rather than probabilities. Recall that a density represents a relative belief. If the density of $f(X = 1.1, Y = 0.9)$ is twice as high as the density that $f(X = 1.1, Y = 1.1)$ the function is expressing that it is twice as likely to find the particular combination of $X = 1.1$ and $Y = 0.9$.

Multivariate Gaussian

The density that is depicted in this example happens to be a particular of joint continuous distribution called Multivariate Gaussian. In fact it is a special case where all of the constituent variables are independent.

Def: Independent Multivariate Gaussian . An Independent Multivariate Gaussian can model a collection of continuous joint random variables $\vec{X} = (X_1 \dots X_n)$ as being a composition of independent normals with means $\vec{\mu} = (\mu_1 \dots \mu_n)$ and standard deviations $\vec{\sigma} = (\sigma_1 \dots \sigma_n)$. Notice how we now have variables in vectors (similar to a list in python). The notation for the multivariate uses vector notation:

$$\vec{X} \sim \vec{\mathrm{N}}(\vec{\mu}, \vec{\sigma})$$

The joint PDF is:

$$\begin{aligned} f(\vec{x}) &= \prod_{i=1}^{n} f(x_i) \\ &= \prod_{i=1}^{n} \frac{1}{\sigma_i \sqrt{2\pi}} e^{\frac{-(x-\mu_i)^2}{2\sigma_i^2}} \end{aligned}$$

And the joint CDF is

$$
\begin{aligned}
F(\vec{x}) &= \prod_{i=1}^{n} F(x_i) \\
&= \prod_{i=1}^{n} \Phi(\frac{x_i - \mu_i}{\sigma_i})
\end{aligned}
$$

Example: Gaussian Blur

In the same way that many single random variables are assumed to be gaussian, many joint random variables may be assumed to be Multivariate Gaussian. Consider this example of Gaussian Blur:

In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise. A Gaussian blur works by convolving an image with a 2D independent multivariate gaussian (with means of 0 and equal valued standard deviations).

In order to use a Gaussian blur, you need to be able to compute the probability mass of that 2D gaussian in the space of pixels. Each pixel is given a weight equal to the probability that X and Y are both within the pixel bounds. The center pixel covers the area where $-0.5 \leq x \leq 0.5$ and $-0.5 \leq y \leq 0.5$. Let's do one step in computing the Gaussian function discretized over image space. What is the weight of the center pixel for gaussian blur with a multivariate gaussian which has means of 0 and standard deviation of 3?

Let $\vec{B}$ be the multivariate gaussian, $\vec{B} \sim \text{N}(\vec{\mu} = [0, 0], \vec{\sigma} = [3, 3])$. Let's compute the CDF of this multivariate gaussian $F(x_1, x_2)$:

$$
\begin{aligned}
F(x_1, x_2) &= \prod_{i=1}^{n} \Phi(\frac{x_i - \mu_i}{\sigma_i}) \\
&= \Phi(\frac{x_1 - \mu_1}{\sigma_1}) \cdot \Phi(\frac{x_2 - \mu_2}{\sigma_2}) \\
&= \Phi(\frac{x_1}{3}) \cdot \Phi(\frac{x_2}{3})
\end{aligned}
$$

Now we are ready to calculate the weight of the center pixel:

$$
\begin{aligned}
&\text{P}(-0.5 < X_1 \leq 0.5, -0.5 < X_2 \leq 0.5) \\
&= F(0.5, 0.5) - F(-0.5, 0.5) + F(-0.5, -0.5) - F(0.5, -0.5) \\
&= \Phi(\frac{0.5}{3}) \cdot \Phi(\frac{0.5}{3}) - \Phi(\frac{-0.5}{3}) \cdot \Phi(\frac{0.5}{3}) + \Phi(\frac{-0.5}{3}) \cdot \Phi(\frac{-0.5}{3}) - \Phi(\frac{0.5}{3}) \cdot \Phi(\frac{-0.5}{3}) \\
&\approx 0.026
\end{aligned}
$$

How can this 2D gaussian blur the image? Wikipedia explains: "Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components; a Gaussian blur is a low pass filter" [2].

Inference

So far we have set the foundation for how we can represent probabilistic models with multiple random variables. These models are especially useful because they let us perform a task called "inference" where we update our belief about one random variable in the model, conditioned on new information about another. Inference in general is hard! In fact, it has been proven that in the worst case, the inference task, can be NP-Hard where n is the number of random variables [1].

First we are going to practice it with two random variables (in this section). Then, later in this unit we are going to talk about inference in the general case, with many random variables.

Earlier we looked at conditional probabilities for events. The first task in inference is to understand how to combine conditional probabilities and random variables. The equations for both the discrete and continuous case are intuitive extensions of our understanding of conditional probability:

The Discrete Conditional

The discrete case, where every random variable in your model is discrete, is a straightforward combination of what you know about conditional probability (which you learned in the context of events). Recall that every relational operator applied to a random variable defines an event. As such the rules for conditional probability directly apply: The conditional probability mass function (PMF) for the discrete case:

Let X and Y be discrete random variables.
Def: Conditional definition with discrete random variables.

$$\mathrm{P}(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

Def: Bayes' Theorem with discrete random variables.

$$\mathrm{P}(X = x|Y = y) = \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)}$$

In the presence of multiple random variables, it becomes increasingly useful to use shorthand! The above definition is identical to this notation where a lowercase symbol such as x is short hand for the event $X = x$:

$$\mathrm{P}(x|y) = \frac{P(x, y)}{P(y)}$$

The conditional definition works for any event and as such we can also write conditionals using cumulative density functions (CDFs) for the discrete case:

$$\begin{aligned}\mathrm{P}(X \leq a|Y = y) &= \frac{\mathrm{P}(X \leq a, Y = y)}{\mathrm{P}(Y = y)} \\ &= \frac{\sum_{x \leq a} \mathrm{P}(X = x, Y = y)}{\mathrm{P}(Y = y)}\end{aligned}$$

Here is a neat result: this last term can be rewritten, by a clever manipulation. We can make the sum extend over the whole fraction:

$$
\begin{aligned}
\mathrm{P}(X \leq a | Y = y) &= \frac{\sum_{x \leq a} \mathrm{P}(X = x, Y = y)}{\mathrm{P}(Y = y)} \\
&= \sum_{x \leq a} \frac{\mathrm{P}(X = x, Y = y)}{\mathrm{P}(Y = y)} \\
&= \sum_{x \leq a} \mathrm{P}(X = x | Y = y)
\end{aligned}
$$

In fact it becomes straight forward to translate the rules of probability (such as Bayes' Theorem, law of total probability, etc) to the language of discrete random variables: we simply need to recall that every relational operator applied to a random variable defines an event.

Mixing Discrete and Continuous

What happens when we want to reason about *continuous* random variables using our rules of probability (such as Bayes' Theorem, law of total probability, chain rule, etc)? There is a simple practical answer: the rules still apply, but we have to replace probability terminology with probability *density* functions. As a concrete example let's look at Bayes' Theorem with one continuous random variable.

Def: Bayes' Theorem with mixed discrete and continuous.

Let X be a continuous random variable and let N be a discrete random variable. The conditional probabilities of X given N and N given X respectively are:

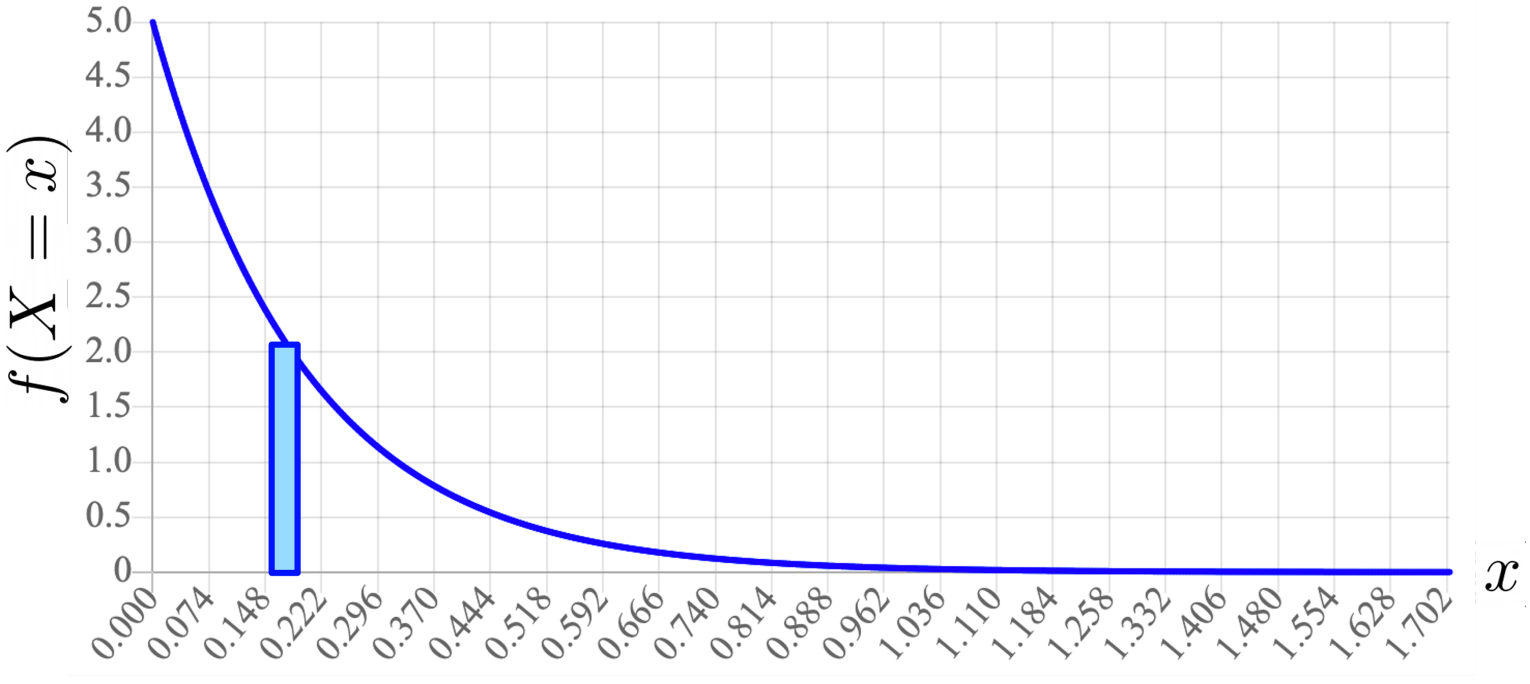
$$
f(X = x | N = n) = \frac{\mathrm{P}(N = n | X = x) f(X = x)}{\mathrm{P}(N = n)}
$$

$$
\mathrm{P}(N = n | X = x) = \frac{f(X = x | N = n)\, \mathrm{P}(N = n)}{f(X = x)}
$$

These equations might seem complicated since they mix probability densities and probabilities. Why should we believe that they are correct? First, observe that anytime the random variable on the left hand side of the conditional is continuous, we use a density, whenever it is discrete, we use a probability. This result can be derived by making the observation:

$$
\mathrm{P}(X = x) = f(X = x) \cdot \epsilon_x
$$

In the limit as $\epsilon_x \to 0$. In order to obtain a probability from a density function is to integrate under the function. If you wanted to approximate the probability that $X = x$ you could consider the area created by a rectangle which has height $f(X = x)$ and some very small width. As that width gets smaller, your answer becomes more accurate:



A value of ϵ_x is problematic if it is left in a formula. However, if we can get them to cancel, we can arrive at a working equation. This is the key insight used to derive the rules of probability in the context of one or more continuous random variables. Again, let X be continuous random variable and let N be a discrete random variable:

$$
\begin{aligned}
\mathrm{P}(N = n|X = x) &= \frac{P(X = x|N = n)\,\mathrm{P}(N = n)}{P(X = x)} && \text{Bayes' Theorem} \\
&= \frac{f(X = x|N = n) \cdot \epsilon_x \cdot \mathrm{P}(N = n)}{f(X = x) \cdot \epsilon_x} && \mathrm{P}(X = x) = f(X = x) \cdot \epsilon_x \\
&= \frac{f(X = x|N = n) \cdot \mathrm{P}(N = n)}{f(X = x)} && \text{Cancel } \epsilon_x
\end{aligned}
$$

This strategy applies beyond Bayes' Theorem. For example here is a version of the Law of Total Probability when X is continuous and N is discrete:

$$f(X = x) = \sum_{n \in N} f(X = x|N = n)\,\mathrm{P}(N = n)$$

Probability Rules with Continuous Random Variables

The strategy used in the above section can be used to derive the rules of probability in the presence of continuous random variables. The strategy also works when there are multiple continuous random variables. For example here is Bayes' Theorem with two continuous random variables.

Def: Bayes' Theorem with continuous random variables.

Let X and Y be continuous random variables.

$$f(X = x|Y = y) = \frac{f(X = x, Y = y)}{f(Y = y)}$$

Example: Inference with a Continuous Variable

Consider the following question:

Question: At birth, girl elephant weights are distributed as a Gaussian with mean 160kg, and standard deviation 7kg. At birth, boy elephant weights are distributed as a Gaussian with mean 165kg, and standard deviation of 3kg. All you know about a newborn elephant is that it is 163kg. What is the probability that it is a girl?

Answer: Let G be an indicator that the elephant is a girl. G is Bern(p = 0.5) Let X be the distribution of weight of the elephant.

$X|G = 1$ is $N(\mu = 160, \sigma^2 = 7^2)$
$X|G = 0$ is $N(\mu = 165, \sigma^2 = 3^2)$

$$\mathrm{P}(G = 1|X = 163) = \frac{f(X = 163|G = 1)\,\mathrm{P}(G = 1)}{f(X = 163)} \qquad \text{Bayes}$$

If we can solve this equation we will have our answer. What is $f(X = 163|G = 1)$? It is the probability density function of a gaussian X which has $\mu = 160, \sigma^2 = 7^2$ at the point x is 163:

$$\begin{aligned} f(X = 163|G = 1) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} && \text{PDF Gauss} \\ &= \frac{1}{7\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{163-160}{7}\right)^2} && \text{PDF } X \text{ at } 163 \end{aligned}$$

Next we note that $\mathrm{P}(G = 0) = \mathrm{P}(G = 1) = \frac{1}{2}$. Putting this all together, and using the law of total probability to compute the denominator we get:

$$\begin{aligned} &\mathrm{P}(G = 1|X = 163) \\ &= \frac{f(X = 163|G = 1)\,\mathrm{P}(G = 1)}{f(X = 163)} \\ &= \frac{f(X = 163|G = 1)\,\mathrm{P}(G = 1)}{f(X = 163|G = 1)\,\mathrm{P}(G = 1) + f(X = 163|G = 0)\,\mathrm{P}(G = 0)} \\ &= \frac{\frac{1}{7\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{163-160}{7}\right)^2} \cdot \frac{1}{2}}{\frac{1}{7\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{163-160}{7}\right)^2} \cdot \frac{1}{2} + \frac{1}{3\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{163-165}{3}\right)^2} \cdot \frac{1}{2}} \\ &= \frac{\frac{1}{7} e^{-\frac{1}{2}\left(\frac{9}{49}\right)}}{\frac{1}{7} e^{-\frac{1}{2}\left(\frac{9}{49}\right)} + \frac{1}{3} e^{-\frac{1}{2}\left(\frac{4}{9}\right)^2}} \\ &\approx 0.328 \end{aligned}$$

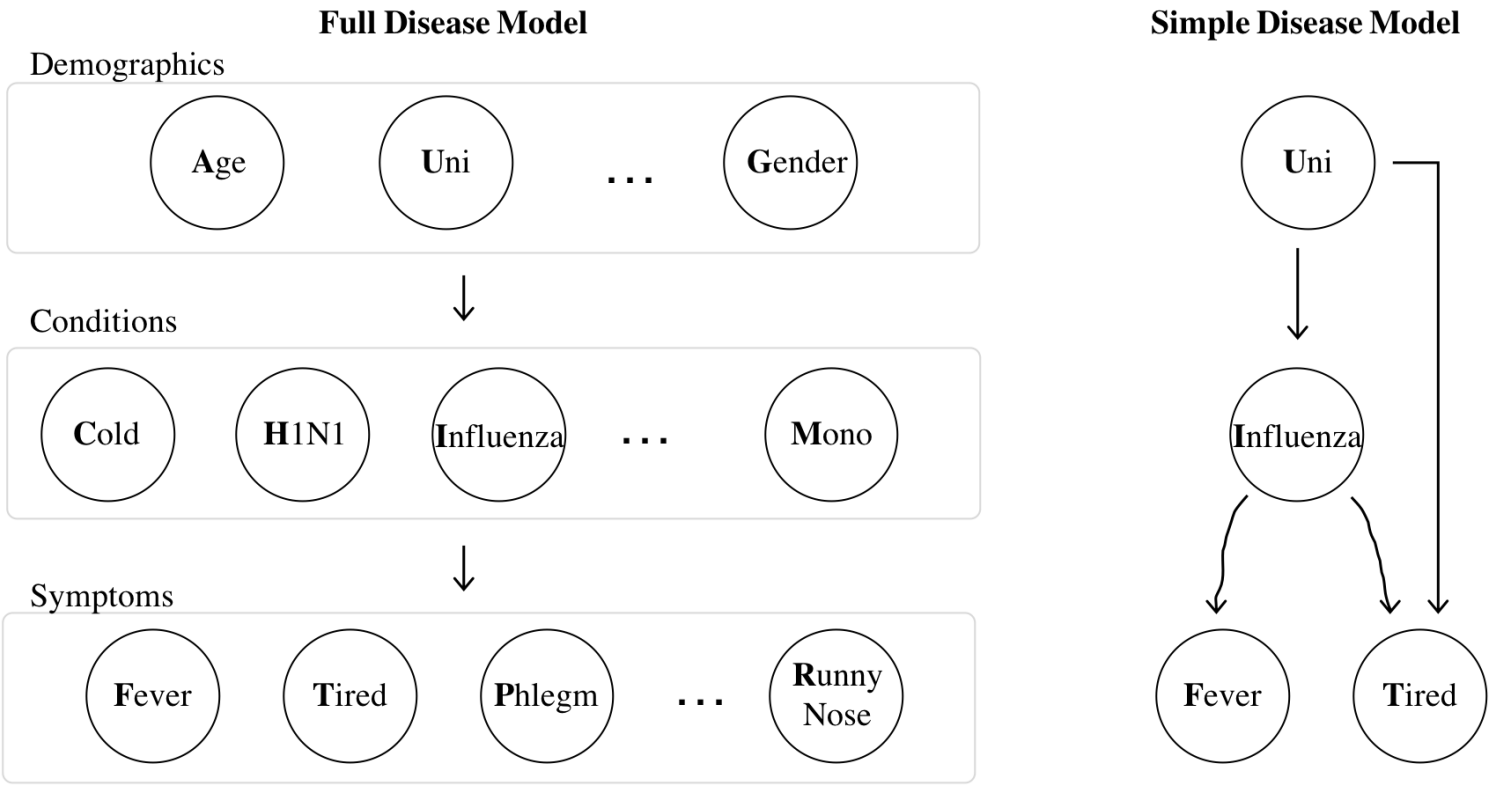
Bayesian Networks

At this point in the reader we have developed tools for analytically solving for probabilities. We can calculate the likelihood of random variables taking on values, even if they are interacting with other random variables (which we have called multi-variate models, or we say the random variables are jointly distributed). We have also started to study samples and sampling.

Consider the WebMD Symptom Checker. WebMD has built a probabilistic model with random variables which roughly fall under three categories: symptoms, risk factors and diseases. For any combination of observed symptoms and risk factors, they can calculate the probability of any disease. For example, they can calculate the probability that I have influenza given that I am a 21-year-old female who has a fever and who is tired: $P(I = 1|A = 21, G = 1, T = 1, F = 1)$. Or they could calculate the probability that I have a cold given that I am a 30-year-old with a runny nose: $P(C = 1|A = 30, R = 1)$. At first blush this might not seem difficult. But as we dig deeper we will realize just how hard it is. There are two challenges: (1) Modelling: sufficiently specifying the probabilistic model and (2) Inference: calculating any desired probability.

Bayesian Networks

Before we jump into how to solve probability (aka inference) questions, let's take a moment to go over how an expert doctor could specify the relationship between so many random variables. Ideally we could have our expert sit down and specify the entire "joint distribution" (see the first lecture on multi-variable models). She could do so either by writing a single equation that relates all the variables (which is as impossible as it sounds), or she could come up with a joint distribution table where she specifies the probability of any possible combination of assignments to variables. It turns out that is not feasible either. Why? Imagine there are $N = 100$ binary random variables in our WebMD model. Our expert doctor would have to specify a probability for each of the $2^N > 10^{30}$ combinations of assignments to those variables, which is approaching the number of atoms in the universe. Thankfully, there is a better way. We can simplify our task if we know the "generative" process that creates a joint assignment. Based on the generative process we can make a data structure known as a **Bayesian Network**. Here are two networks of random variables for diseases:



For diseases the flow of influence is directed. The states of "demographic" random variables influence whether someone has particular "conditions", which influence whether someone shows particular "symptoms". On the right is a simple model with only four random variables. Though this is a less interesting model it is easier to understand when first learning Bayesian Networks. Being in university

(binary) influences whether or not someone has influenza (binary). Having influenza influences whether or not someone has a fever (binary) and the state of university and influenza influences whether or not someone feels tired (also binary).

In a Bayesian Network an arrow from random variable X to random variable Y articulates our assumption that X directly influences the likelihood of Y. We say that X is a *parent* of Y. To fully define the Bayesian network we must provide a way to compute the probability of each random variable (X_i) conditioned on knowing the value of all their parents: $P(X_i = k|\text{Parents of } X_i \text{ take on specified values})$. Here is a concrete example of what needs to be defined for the simple disease model. Recall that each of the random variables is binary:

$$P(\text{Uni} = 1) = 0.8$$
$$P(\text{Influenza} = 1|\text{Uni} = 1) = 0.2 \qquad P(\text{Fever} = 1|\text{Influenza} = 1) = 0.9$$
$$P(\text{Influenza} = 1|\text{Uni} = 0) = 0.1 \qquad P(\text{Fever} = 1|\text{Influenza} = 0) = 0.05$$
$$P(\text{Tired} = 1|\text{Uni} = 0, \text{Influenza} = 0) = 0.1 \qquad P(\text{Tired} = 1|\text{Uni} = 0, \text{Influenza} = 1) = 0.9$$
$$P(\text{Tired} = 1|\text{Uni} = 1, \text{Influenza} = 0) = 0.8 \qquad P(\text{Tired} = 1|\text{Uni} = 1, \text{Influenza} = 1) = 1.0$$

Let's put this in programming terms. All that we need to do in order to code up a Bayesian network is to define a function: `getProbXi(i, k, parents)` which returns the probability that X_i (the random var with index `i`) takes on the value `k` given a value for each of the parents of X_i encoded by the list `parents`: $P(X_i = x_i|\text{Values of parents of } X_i)$

Deeper understanding: The reason that a Bayes Net is so useful is that the "joint" probability can be expressed in exponentially less space as the product of the probabilities of each random variable conditioned on its parents! Without loss of generality, let X_i refer to the ith random variable (such that if X_i is a parent of X_j then $i < j$):

$$P(\text{Joint}) = P(X_1 = x_1, \ldots, X_n = x_n) = \prod_i P(X_i = x_i|\text{Values of parents of } X_i)$$

What assumptions are implicit in a Bayes Net? Using the chain rule we can decompose the **exact** joint probability for n random variables. To make the following math easier to digest I am going to use x_i as shorthand for the event that $X_i = x_i$:

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

By looking at the difference in the two equations, we can see that a Bayes Net is assuming that

$$P(x_i|x_{i-1}, \ldots, x_1) = P(x_i|\text{Values of parents of } X_i)$$

This is a conditional independence statement. It is saying that 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. Formally we say that X_i is conditionally independent of its non-descendents, given its parents. What is a non-descendent again? In a graph, a descendent of X_i is anything which is in the subtree that starts at X_i. Everything else is a non-descendent. Non-descendents include the "ancestor" nodes of X_i as well as nodes which are totally unconnected to X_i. When designing Bayes Nets you don't have to think about this assumption directly. It turns out to be a naturally good assumption if the arrows between your nodes follow a causal path.

Designing a Bayes Net

There are several steps to designing a Bayes Net.

1. Choose your random variables, and make them nodes.
2. Add edges, often based off your assumptions about which nodes directly cause which others.
3. Define $P(X_i = x_i|\text{Values of parents of } X_i)$ for all nodes.

As you might have guessed, we can do step (2) and (3) by hand, or, we can have computers try and perform those tasks based on data. The first task is called "structure learning" and the second is an instance of "machine learning." There are fully autonomous solutions to structure learning—but they

only work well if you have a massive amount of data. Alternatively people will often compute a statistic called correlation between all pairs of random variables to help in the art form of designing a Bayes Net.

In the next part of the reader we are going to talk about how we could learn $P(X_i = x_i | \text{Values of parents of } X_i)$ from data. For now let's start with the (reasonable) assumption that an expert can write down these functions in equations or as python: `getProbXi`.

Next Steps

Great! We have a feasible way to define a large network of random variables. First challenge complete. We haven't talked about continuous or multinomial random variables in Bayes Nets. None of the theory changes: the expert will just have to define `getProbXi` to handle more values of `k` than 0 or 1.

A Bayesian network is not very interesting to us unless we can use it to solve different conditional probability questions. How can we perform "inference" on a network as complex as a Bayesian network?

Independence in Variables

Discrete

Two discrete random variables X and Y are called independent if:

$$\mathrm{P}(X = x, Y = y) = \mathrm{P}(X = x)\,\mathrm{P}(Y = y) \text{ for all } x, y$$

Intuitively: knowing the value of X tells us nothing about the distribution of Y. If two variables are not independent, they are called dependent. This is a similar conceptually to independent events, but we are dealing with multiple *variables*. Make sure to keep your events and variables distinct.

Continuous

Two continuous random variables X and Y are called independent if:

$$\mathrm{P}(X \leq a, Y \leq b) = \mathrm{P}(X \leq a)\,\mathrm{P}(Y \leq b) \text{ for all } a, b$$

This can be stated equivalently using either the CDF or the PDF:

$$F_{X,Y}(a, b) = F_X(a)F_Y(b) \text{ for all } a, b$$
$$f(X = x, Y = y) = f(X = x)f(Y = y) \text{ for all } x, y$$

More generally, if you can factor the joint density function then your random variable are independent (or the joint probability function for discrete random variables):

$$f(X = x, Y = y) = h(x)g(y)$$
$$\mathrm{P}(X = x, Y = y) = h(x)g(y)$$

Example: Showing Independence

Let N be the # of requests to a web server/day and that $N \sim \mathrm{Poi}(\lambda)$. Each request comes from a human with probability = p or from a "bot" with probability = $(1–p)$. Define X to be the # of requests from humans/day and Y to be the # of requests from bots/day. Show that the number of requests from humans, X, is independent of the number of requests from bots, Y.

Since requests come in independently, the probability of X conditioned on knowing the number of requests is a Binomial. Specifically:

$$(X|N) \sim \mathrm{Bin}(N, p)$$
$$(Y|N) \sim \mathrm{Bin}(N, 1 - p)$$

To get started we need to first write an expression for the joint probability of X and Y. To do so, we use the chain rule:

$$\mathrm{P}(X = x, Y = y) = \mathrm{P}(X = x, Y = y|N = x + y)\,\mathrm{P}(N = x + y)$$

We can calculate each term in this expression. The first term is the PMF of the binomial $X|N$ having x "successes". The second term is the probability that the Poisson N takes on the value $x + y$:

$$\mathrm{P}(X = x, Y = y|N = x + y) = \binom{x + y}{x} p^x (1 - p)^y$$
$$\mathrm{P}(N = x + y) = e^{-\lambda}\frac{\lambda^{x+y}}{(x + y)!}$$

Now we can put those together we have an expression for the joint:

$$\mathrm{P}(X = x, Y = y) = \binom{x+y}{x} p^x (1-p)^y e^{-\lambda} \frac{\lambda^{x+y}}{(x+y)!}$$

At this point we have derived the joint distribution over X and Y. In order to show that these two are independent, we need to be able to factor the joint:

$$\begin{aligned}
&\mathrm{P}(X = x, Y = y) \\
&\quad = \binom{x+y}{x} p^x (1-p)^y e^{-\lambda} \frac{\lambda^{x+y}}{(x+y)!} \\
&\quad = \frac{(x+y)!}{x! \cdot y!} p^x (1-p)^y e^{-\lambda} \frac{\lambda^{x+y}}{(x+y)!} \\
&\quad = \frac{1}{x! \cdot y!} p^x (1-p)^y e^{-\lambda} \lambda^{x+y} && \text{Cancel (x+y)!} \\
&\quad = \frac{p^x \cdot \lambda^x}{x!} \cdot \frac{(1-p)^y \cdot \lambda^y}{y!} \cdot e^{-\lambda} && \text{Rearrange}
\end{aligned}$$

Because the joint can be factored into a term that only has x and a term that only has y, the random variables are independent.

Symmetry of Independence

Independence is symmetric. That means that if random variables X and Y are independent, X is independent of Y and Y is independent of X. This claim may seem meaningless but it can be very useful. Imagine a sequence of events $X_1, X_2, \ldots$. Let A_i be the event that X_i is a "record value" (eg it is larger than all previous values). Is A_{n+1} independent of A_n? It is easier to answer that A_n is independent of A_{n+1}. By symmetry of independence both claims must be true.

Expectation of Products

Lemma: Product of Expectation for Independent Random Variables:
If two random variables X and Y are independent, the expectation of their product is the product of the individual expectations.

$$\begin{aligned}
E[X \cdot Y] &= E[X] \cdot E[Y] && \text{if } X \text{ and } Y \text{ are independent} \\
E[g(X)h(Y)] &= E[g(X)]E[h(Y)] && \text{where } g \text{ and } h \text{ are functions}
\end{aligned}$$

Note that this assumes that X and Y are independent. Contrast this to the sum version of this rule (expectation of sum of random variables, is the sum of individual expectations) which does **not** require the random variables to be independent.

Correlation

Covariance

Covariance is a quantitative measure of the extent to which the deviation of one variable from its mean matches the deviation of the other from its mean. It is a mathematical relationship that is defined as:

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

That is a little hard to wrap your mind around (but worth pushing on a bit). The outer expectation will be a weighted sum of the inner function evaluated at a particular (x, y) weighted by the probability of (x, y). If x and y are both above their respective means, or if x and y are both below their respective means, that term will be positive. If one is above its mean and the other is below, the term is negative. If the weighted sum of terms is positive, the two random variables will have a positive correlation. We can rewrite the above equation to get an equivalent equation:

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

> ***Lemma: Correlation of Independent Random Variables***:
> If two random variables X and Y are independent, than their covariance must be 0.
>
> $$\begin{aligned} \text{Cov}(X, Y) &= E[XY] - E[Y]E[X] && \text{Def of Cov} \\ &= E[X]E[Y] - E[Y]E[X] && \text{Lemma Product of Expectation} \\ &= 0 \end{aligned}$$
>
> Note that the reverse claim is not true. Covariance of 0 does not prove independence.

Using this equation (and the product lemma) it is easy to see that if two random variables are independent their covariance is 0. The reverse is *not* true in general.

Properties of Covariance

Say that X and Y are arbitrary random variables:

$$\begin{aligned} &\text{Cov}(X, Y) = \text{Cov}(Y, X) \\ &\text{Cov}(X, X) = E[X^2] - E[X]E[X] = \text{Var}(X) \\ &\text{Cov}(aX + b, Y) = a\text{Cov}(X, Y) \end{aligned}$$

Let $X = X_1 + X_2 + \cdots + X_n$ and let $Y = Y_1 + Y_2 + \cdots + Y_m$. The covariance of X and Y is:

$$\text{Cov}(X, Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} \text{Cov}(X_i, Y_j)$$

$$\text{Cov}(X, X) = \text{Var}(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Cov}(X_i, X_j)$$

That last property gives us a third way to calculate variance. We can use it to, again, show how to get the variance of a Binomial.

Correlation

We left off last class talking about covariance. Covariance was interesting because it was a quantitative measurement of the relationship between two variables. Today we are going to extend that concept to correlation. Correlation between two random variables, $\rho(X, Y)$ is the covariance of the two variables normalized by the variance of each variable. This normalization cancels the units out:

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

Correlation measures linearity between X and Y.

$$\begin{array}{ll} \rho(X,Y) = 1 & Y = aX + b \text{ where } a = \sigma_y/\sigma_x \\ \rho(X,Y) = -1 & Y = aX + b \text{ where } a = -\sigma_y/\sigma_x \\ \rho(X,Y) = 0 & \text{absence of linear relationship} \end{array}$$

If $\rho(X,Y) = 0$ we say that X and Y are "uncorrelated."

When people use the term correlation, they are actually referring to a specific type of correlation called "Pearson" correlation. It measures the degree to which there is a linear relationship between the two variables. An alternative measure is "Spearman" correlation which has a formula almost identical to your regular correlation score, with the exception that the underlying random variables are first transformed into their rank. "Spearman" correlation is outside the scope of CS109.

General Inference

A Bayesian Network gives us a reasonable way to specify the joint probability of a network of many random variables. Before we celebrate, realize that we still don't know how to use such a network to answer probability questions. There are many techniques for doing so. I am going to introduce you to one of the great ideas in probability for computer science: we can use sampling to solve inference questions on Bayesian networks. Sampling is frequently used in practice because it is relatively easy to understand and easy to implement.

Rejection Sampling

As a warmup consider what it would take to sample an assignment to each of the random variables in our Bayes net. Such a sample is often called a "joint sample" or a "particle" (as in a particle of sand). To sample a particle, simply sample a value for each random variable one at a time based on the value of the random variable's parents. This means that if X_i is a parent of X_j, you will have to sample a value for X_i before you sample a value for X_j.

Let's work through an example of sampling a "particle" for the Simple Disease Model in the Bayes Net section:

1. Sample from $P(\text{Uni} = 1)$: $\text{Bern}(0.8)$. Sampled value for Uni is 1.
2. Sample from $P(\text{Influenza} = 1 | \text{Uni} = 1)$: $\text{Bern}(0.2)$. Sampled value for Influenza is 0.
3. Sample from $P(\text{Fever} = 1 | \text{Influenza} = 0)$: $\text{Bern}(0.05)$. Sampled value for Fever is 0.
4. Sample from $P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 0)$: $\text{Bern}(0.8)$. Sampled value for Tired is 0.

Thus the sampled particle is: [Uni = 1, Influenza = 0, Fever = 0, Tired = 0]. If we were to run the process again we would get a new particle (with likelihood determined by the joint probability).

Now our strategy is simple: we are going to generate N samples where N is in the hundreds of thousands (if not millions). Then we can compute probability queries by counting. Let $N(\mathbf{X} = \mathbf{k})$ be notation for the number of particles where random variables $\mathbf{X}$ take on values $\mathbf{k}$. Recall that the bold notation $\mathbf{X}$ means that $\mathbf{X}$ is a vector with one or more elements. By the "frequentist" definition of probability:

$$P(\mathbf{X} = \mathbf{k}) = \frac{N(\mathbf{X} = \mathbf{k})}{N}$$

Counting for the win! But what about conditional probabilities? Well using the definition of conditional probabilities, we can see it's still some pretty straightforward counting:

$$P(\mathbf{X} = \mathbf{a} | \mathbf{Y} = \mathbf{b}) = \frac{P(\mathbf{X} = \mathbf{a}, \mathbf{Y} = \mathbf{b})}{P(\mathbf{Y} = \mathbf{b})} = \frac{\frac{N(\mathbf{X}=\mathbf{a}, \mathbf{Y}=\mathbf{b})}{N}}{\frac{N(\mathbf{Y}=\mathbf{b})}{N}} = \frac{N(\mathbf{X} = \mathbf{a}, \mathbf{Y} = \mathbf{b})}{N(\mathbf{Y} = \mathbf{b})}$$

Let's take a moment to recognize that this is straight-up fantastic. General inference based on analytic probability (math without samples) is hard even given a Bayesian network (if you don't believe me, try to calculate the probability of flu conditioning on one demographic and one symptom in the Full Disease Model). However if we generate enough samples we can calculate any conditional probability question by reducing our samples to the ones that are consistent with the condition ($\vec{Y} = \vec{b}$) and then counting how many of those are also consistent with the query ($\vec{X} = \vec{a}$). Here is the algorithm in pseudocode:

```
N = 10000
# "query" is the assignment to variables we want probabilities for
# condition" is the assignments to variables we will condition on
def get_any_probability(query, condition):
  particles = generate_many_joint_samples(N)
  cond_particles = reject_non_consistent_samples(particles, condition)
  K = count_consistent_samples(cond_particles, query)
  return K / len(cond_particles)
```

This algorithm is sometimes called "Rejection Sampling" because it works by generating many particles from the joint distribution and rejecting the ones that are not consistent with the set of assignments we are conditioning on. Of course this algorithm is an approximation, though with enough samples it often works out to be a very good approximation. However, in cases where the event we're conditioning on is rare enough that it doesn't occur after millions of samples are generated, our algorithm will not work. The last line of our code will result in a divide by 0 error. See the next section for solutions!

General Inference when Conditioning on Rare Events

Joint Sampling is a powerful technique that takes advantage of computational power. But it doesn't always work. In fact it doesn't work any time that the probability of the event we are conditioning is rare enough that we are unlikely to ever produce samples that exactly match the event. The simplest example is with continuous random variables. Consider the Simple Disease Model. Let's change Fever from being a binary variable to being a continuous variable. To do so the only thing we need to do is re-specify the likelihood of fever given assignments to its parents (influenza). Let's say that the likelihoods come from the normal PDF:

$$\text{if Influenza} = 0\text{, then Fever} \sim N(\mu = 98.3, \sigma = 0.7)$$
$$\therefore f(\text{Fever} = x) = \frac{1}{\sqrt{2\pi \cdot 0.7}} e^{-\frac{(x-98.3)^2}{2 \cdot 0.7}}$$

$$\text{if Influenza} = 1\text{, then Fever} \sim N(\mu = 100.0, \sigma = 1.8)$$
$$\therefore f(\text{Fever} = x) = \frac{1}{\sqrt{2\pi \cdot 1.8}} e^{-\frac{(x-100.0)^2}{2 \cdot 1.8}}$$

Drawing samples (aka particles) is still straightforward. We apply the same process until we get to the step where we sample a value for the Fever random variable (in the example from the previous section that was step 3). If we had sampled a 0 for influenza we draw a value for fever from the normal for healthy adults (which has $\mu = 98.3$). If we had sampled a 1 for influenza we draw a value for fever from the normal for adults with the flu (which has $\mu = 100.0$). The problem comes in the "rejection" stage of joint sampling.

When we sample values for fever we get numbers with infinite precision (eg 100.819238 etc). If we condition on someone having a fever equal to 101 we would reject every single particle. Why? No particle will have exactly a fever of 101.

There are several ways to deal with this problem. One especially easy solution is to be less strict when rejecting particles. We could round all fevers to whole numbers.

There is an algorithm called "Likelihood Weighting" which sometimes helps, but which we don't cover in CS109. Instead, in class we talked about a new algorithm called Markov Chain Monte Carlo (MCMC) that allowed us to sample from the "posterior" probability: the distribution of random variables after (post) us fixing variables in the conditioned event. The version of MCMC we talked about is called Gibbs Sampling. While I don't require that students in CS109 know how to implement Gibbs Sampling, I wanted everyone to know that it exists and that it isn't beyond your capabilities. If you need to use it, you can learn it given the knowledge you have now.

MCMC does require more math than Joint Sampling. For every random variable you will need to specify how to calculate the likelihood of assignments given the variable's: parents, children and parents of its children (a set of variables cozily called a "blanket"). Want to learn more? Take CS221 or CS228!

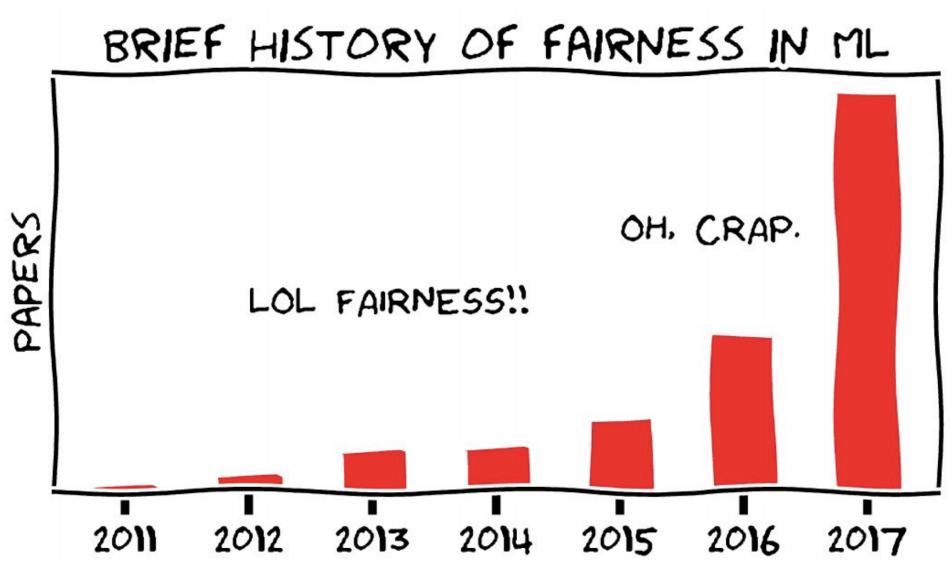
Thoughts

While there are slightly-more-powerful "general inference algorithms" that you will get to learn in the future, it is worth recognizing that at this point we have reached an important milestone in CS109. You can take very complicated probability models (encoded as Bayesian networks) and can answer general inference queries on them. To get there we worked through the concrete example of predicting disease. While the WebMD website is great for home users, similar probability models are being used in thousands of hospitals around the world. As you are reading this general inference is being used to improve health care (and sometimes even save lives) for real human beings. That's some probability for computer scientists that is worth learning. What if we don't have an expert? Could we learn those probabilities from data? Jump to part 5 to answer that question.

Fairness in Artificial Intelligence

Artificial Intelligence often gives the impression that it is objective and "fair". However, algorithms are made by humans and trained by data which may be biased. There are several examples of deployed AI algorithms that have been shown to make decisions that were biased based on gender, race or other protected demographics — even when there was no intention for it.

These examples have also led to a necessary research into a growing field of algorithmic fairness. How can we demonstrate, or prove, that an algorithm is behaving in a way that we think is appropriate? What is fair? Clearly these are complex questions and are deserving of a complete conversation. This example is simple for the purpose of giving an introduction to the topic.



ML stands for Machine Learning. Solon Barocas and Moritz Hardt, "Fairness in Machine Learning", NeurIPS 2017

What is Fairness?

An artificial intelligence algorithm is going to be used to make a binary prediction (G for guess) for whether a person will repay a loan. The question has come up: is the algorithm "fair" with respect to a binary protected demographic (D for demographic)? To answer this question we are going to analyze predictions the algorithm made on historical data. We are then going to compare the predictions to the true outcome (T for truth). Consider the following joint probability table from the history of the algorithm's predictions:

$D = 0$

	$G = 0$	$G = 1$
$T = 0$	0.21	0.32
$T = 1$	0.07	0.28

$D = 1$

	$G = 0$	$G = 1$
$T = 0$	0.01	0.01
$T = 1$	0.02	0.08

Recall that cell $D = i, G = j, T = k$ contains the probability $\mathrm{P}(D = i, G = j, T = k)$. A joint probability table gives the probability of all combination of events. Recall that since each cell is mutually exclusive, the $\sum_i \sum_j \sum_k \mathrm{P}(D = i, G = j, T = k) = 1$. Note that this assumption of mutual exclusion could be problematic for demographic variables (some people are mixed ethnicity, etc) which gives you a hint that we are just scratching the surface in our conversation about fairness. Let's use this joint probability to learn about some of the common definitions of fairness.

Practice with joint marginalization

What is $\mathrm{P}(D = 0)$? What is $\mathrm{P}(D = 1)$?

Probabilities with assignments to a subset of the random variables in the joint distribution can be

calculated by a process called *marginalization*: sum the probability from all cells where that assignment is true.

$$\begin{aligned}\mathrm{P}(D=1) &= \sum_{j\in\{0,1\}}\sum_{k\in\{0,1\}}\mathrm{P}(D=1,G=j,T=k)\\ &= 0.01+0.01+0.02+0.08 = 0.12\\ \mathrm{P}(D=0) &= \sum_{j\in\{0,1\}}\sum_{k\in\{0,1\}}\mathrm{P}(D=0,G=j,T=k)\\ &= 0.21+0.32+0.07+0.28 = 0.88\end{aligned}$$

Note that $\mathrm{P}(D=0)+\mathrm{P}(D=1)=1$. That implies that the demographics are mututally exclusive.

Fairness definition #1: Parity
An algorithm satisfies "parity" if the probability that the algorithm makes a positive prediction ($G=1$) is the same regardless of being conditioned on demographic variable.

Does this algorithm satisfy parity?

$$\begin{aligned}P(G=1|D=1) &= \frac{P(G=1,D=1)}{P(D=1)} && \text{Cond. Prob.}\\ &= \frac{P(G=1,D=1,T=0)+P(G=1,D=1,T=1)}{P(D=1)} && \text{Prob or}\\ &= \frac{0.01+0.08}{0.12} = 0.75 && \text{From joint}\\ P(G=1|D=0) &= \frac{P(G=1,D=0)}{P(D=0)} && \text{Cond. Prob.}\\ &= \frac{P(G=1,D=0,T=0)+P(G=1,D=0,T=1)}{P(D=0)} && \text{Prob or}\\ &= \frac{0.32+0.28}{0.88} \approx 0.68 && \text{From joint}\end{aligned}$$

No. Since $P(G=1|D=1)\neq P(G=1|D=0)$ this algorithm does not satisfy parity. It is more likely to guess 1 when the demographic indicator is 1.

Fairness definition #2: Calibration
An algorithm satisfies "calibration" if the probability that the algorithm is correct ($G=T$) is the same regardless of demographics.

Does this algorithm satisfy calibration?

The algorithm satisfies calibration if $P(G=T|D=0)=P(G=T|D=1)$

$$\begin{aligned}P(G=T|D=0) &= P(G=1,T=1|D=0)+P(G=0,T=0|D=0)\\ &= \frac{0.28+0.21}{0.88} \approx 0.56\\ P(G=T|D=1) &= P(G=1,T=1|D=1)+P(G=0,T=0|D=1)\\ &= \frac{0.08+0.01}{0.12} = 0.75\end{aligned}$$

No: $P(G=T|D=0)\neq P(G=T|D=1)$

Fairness definition #3: Equality of Odds
An algorithm satisfies "equality of odds" if the probability that the algorithm *predicts a positive outcome* ($G=1$) is the same regardless of demographics *given* that the outcome will occur ($T=1$).

Does this algorithm satisfy equality of odds?

The algorithm satisfies equality of odds if $P(G=1|D=0,T=1)=P(G=1|D=1,T=1)$

$$
\begin{aligned}
P(G=1|D=1,T=1) &= \frac{P(G=1,D=1,T=1)}{P(D=1,T=1)} \\
&= \frac{0.08}{0.08+0.02} = 0.8 \\
P(G=1|D=0,T=1) &= \frac{P(G=1,D=0,T=1)}{P(D=0,T=1)} \\
&= \frac{0.28}{0.28+0.07} = 0.8
\end{aligned}
$$

Yes: $P(G=1|D=0,T=1) = P(G=1|D=1,T=1)$

Which of these definitions seems right to you? It turns out, it can actually be proven that these three cannot be jointly optimized, and this is called the Impossibility Theorem of Machine Fairness. In other words, any AI system we build will necessarily violate some notion of fairness. For a deeper treatment of the subject, here is a useful summary of the latest research Pessach et al. Algorithmic Fairness.

Gender Shades

In 2018, Joy Buolamwini and Timnit Gebru had a breakthrough result called "gender shades" published in the first conference on Fairness, Accountability and Transparency in ML [1]. They showed that facial recognition algorithms, which had been deployed to be used by Facebook, IBM and Microsoft, were substantially better at making predicitons (in this case classifying gender) when looking at lighter skinned men than darker skinned women. Their work exposed several shortcomings in production AI: biased datasets, optimizing for average accuracy (which means that the majority demographic gets most weight) lack of awareness of intersectionality, and more. Let's take a look at some of their results.

Figure by Joy Buolamwini and Timnit Gebru. Facial recognition algorithms perform very differently depending on who they are looking at. [1]

Timnit and Joy looked at three classifiers trained to predict gender, and computed several statistics. Let's take a look at one statistic, accuracy, for one of the facial recognition classifiers, IBMs:

	Women	**Men**	**Darker**	**Lighter**
Accuracy	79.7	94.4	77.6	96.8

Using the language of fairness, accuracy measures $\mathrm{P}(G=T)$. The cell in the table above under "Women" says the accuracy when looking at photos of women $\mathrm{P}(G=T|D=\mathrm{Women})$. It is easy to show that these production level systems are terribly "uncalibrated":

$$\mathrm{P}(G=T|D=\mathrm{Woman}) \neq \mathrm{P}(G=T|D=\mathrm{Man})$$

$$\mathrm{P}(G=T|D=\mathrm{Lighter}) \neq \mathrm{P}(G=T|D=\mathrm{Darker})$$

Why should we care about calibration and not the other definitions of fairness? In this case the classifier was making a prediction of gender where a positive prediction (say predicting women) doesn't have a directly associated reward as in our above example, where we were predicting if someone should receive a loan. As such the most salient idea is: is the algorithm just as accurate for different genders (calibration)?

The lack of calibration between men/women and lighter/darker skinned photos is an issue. What Joy and Timnit showed next was that the problem becomes even worse when you look at intersectional demographics.

	Darker Men	**Darker Women**	**Lighter Men**	**Lighter Women**
Accuracy	88.0	65.3	99.7	92.9

If the algorithms were "fair" according to the calibration you would expect the accuracy to be the same regardless of demographics. Instead there is almost a 34.2 percentage point difference! $\mathrm{P}(G = T|D = \text{Darker Woman}) = 65.3$ compared to $\mathrm{P}(G = T|D = \text{Ligher Man}) = 99.7$

[1] Buolamwini, Gebru. Gender Shades. 2018

Ways Forward?

Wadsworth et al. Achieving Fairness through Adversarial Learning

Federalist Paper Authorship

Let's write a program to decide whether or not James Madison or Alexander Hamilton wrote Federalist Paper 49. Both men have claimed to have written it, and hence the authorship is in dispute. First we used historical essays to estimate p_i, the probability that Hamilton generates the word i (independent of all previous and future choices of words). Similarly we estimated q_i, the probability that Madison generates the word i. For each word i we observe the number of times that word occurs in Federalist Paper 49 (we call that count c_i). We assume that, given no evidence, the paper is equally likely to be written by Madison or Hamilton.

Define three events: H is the event that Hamilton wrote the paper, M is the event that Madison wrote the paper, and D is the event that a paper has the collection of words observed in Federalist Paper 49. We would like to know whether $P(H|D)$ is larger than $P(M|D)$. This is equivalent to trying to decide if $P(H|D)/P(M|D)$ is larger than 1.

The event $D|H$ is a multinomial parameterized by the values p. The event $D|M$ is also a multinomial, this time parameterized by the values q.

Using Bayes Rule we can simplify the desired probability.

$$\frac{P(H|D)}{P(M|D)} = \frac{\frac{P(D|H)P(H)}{P(D)}}{\frac{P(D|M)P(M)}{P(D)}} = \frac{P(D|H)P(H)}{P(D|M)P(M)} = \frac{P(D|H)}{P(D|M)}$$
$$= \frac{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i p_i^{c_i}}{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i q_i^{c_i}} = \frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}$$

This seems great! We have our desired probability statement expressed in terms of a product of values we have already estimated. However, when we plug this into a computer, both the numerator and denominator come out to be zero. The product of many numbers close to zero is too hard for a computer to represent. To fix this problem, we use a standard trick in computational probability: we apply a log to both sides and apply some basic rules of logs.

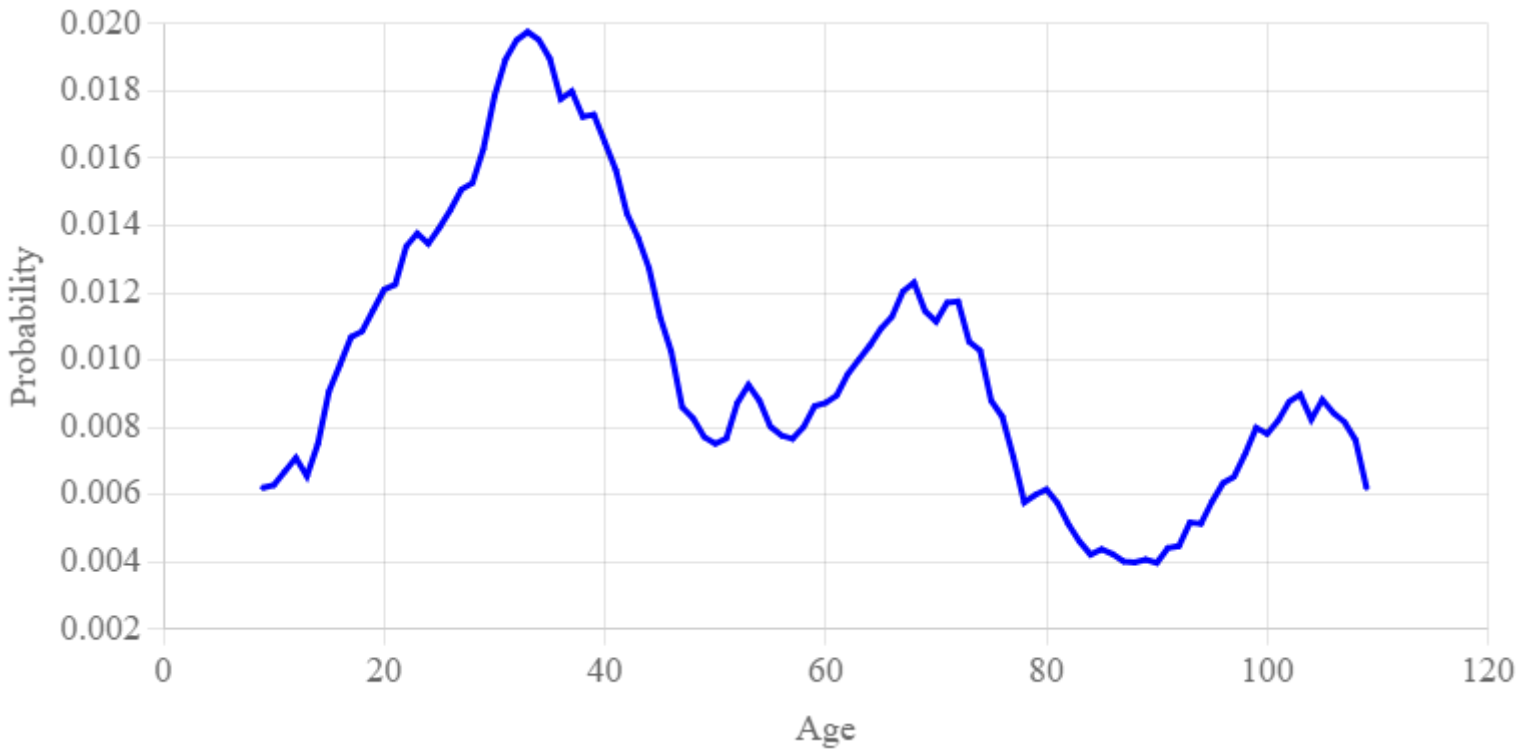
$$\log\Big(\frac{P(H|D)}{P(M|D)}\Big) = \log\Big(\frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}\Big)$$
$$= \log(\prod_i p_i^{c_i}) - \log(\prod_i q_i^{c_i})$$
$$= \sum_i \log(p_i^{c_i}) - \sum_i \log(q_i^{c_i})$$
$$= \sum_i c_i \log(p_i) - \sum_i c_i \log(q_i)$$

This expression is "numerically stable" and my computer returned that the answer was a negative number. We can use exponentiation to solve for $P(H|D)/P(M|D)$. Since the exponent of a negative number is a number smaller than 1, this implies that $P(H|D)/P(M|D)$ is smaller than 1. As a result, we conclude that Madison was more likely to have written Federalist Paper 49. That is the standing assumption currently made by historians!

Name to Age

Because of shifting patterns in name popularity, a person's name is a hint as to their age. The United States publishes a data which contains counts of how many US residents were born with a given name in a given year, based off Social Security applications. We can use inference to compute the reverse probability distribution: an updated belief in a person's age, given their name. As a reminder, if I know the year someone was born, I can calculate their age within one year.

Query Name: Katherine ✔



Records with name: 589753

This demo is based on real data from US Social Security applications between 1914 and 2014. Thank you to https://www.kaggle.com/kaggle/us-baby-names for compiling the data. Download Data.

Computation

The US Social Security applications data provides you with a function: `count(year, name)` which returns the number of US citizens, born in a given year with a given name. You also have access to a list `names` which has each name ever given in the US and `years` which has all the years. This function is implicitly giving us the joint probability over names and birth year. The probability of a joint assignment to name and birth year can be estimated as the count of people with that name, born on that year, over the total number of people in the dataset. Let B be the year someone is born, and let N be their name:

$$P(B = b, N = n) \approx \frac{\text{count}(b, n)}{\sum_{i \in \text{names}} \sum_{j \in \text{years}} \text{count}(i, j)}$$

The question we would really like to answer is: what is your belief that a resident was born in 1950, given that their name is Gary?

We can get started by applying the definition of conditional probability for random variables:

$$\text{P}(B = 1950 | N = \text{Gary}) = \frac{\text{P}(N = \text{Gary}, B = 1950)}{\text{P}(N = \text{Gary})}$$

But this leaves one term to compute: $\text{P}(N = \text{Gary})$ which we can compute using marginalization:

$$\begin{aligned}\mathrm{P}(N = \text{Gary}) &= \sum_{y \in \text{years}} P(B = y, N = \text{Gary}) \\ &\approx \frac{\sum_{y \in \text{years}} \text{count}(y, \text{Gary})}{\sum_{i \in \text{names}} \sum_{j \in \text{years}} \text{count}(i, j)}\end{aligned}$$

Putting this all together we have:

$$\begin{aligned}\mathrm{P}(B = 1950 | N = \text{Gary}) &= \frac{\mathrm{P}(N = \text{Gary}, B = 1950)}{\mathrm{P}(N = \text{Gary})} \\ &\approx \frac{\left(\frac{\text{count}(1950, \text{Gary})}{\sum_{i \in \text{names}} \sum_{j \in \text{years}} \text{count}(i, j)}\right)}{\left(\frac{\sum_{y \in \text{years}} \text{count}(y, \text{Gary})}{\sum_{i \in \text{names}} \sum_{j \in \text{years}} \text{count}(i, j)}\right)} \\ &\approx \frac{\text{count}(1950, \text{Gary})}{\sum_{y \in \text{years}} \text{count}(y, \text{Gary})}\end{aligned}$$

More generally, for any name, we can compute the conditional probability mass function over birth year B:

$$\mathrm{P}(B = b | N = n) \approx \frac{\text{count}(b, n)}{\sum_{y \in \text{years}} \text{count}(y, n)}$$

From Birth Year to Age

Of course, if B is the birth year of a person, their age, A is approximately the current year minus B. This could be off by one if someone has a birth day later in the year, but we will ignore this small deviation for now. So for example, if we think that a person was born in 1988, since the current year is 2023 then their age is 2023 - 1988 = 35

Assumptions

This problem makes many assumptions which are worth highlighting. In fact, any time we make generalizations (especially about demographics) based on sparse information we should tread lightly. Here are the assumptions that I can think of:

1. This data only is accurate for names of people in the US. The probability of age given names could be very different in other countries.
2. The US census is not perfect. It does not capture all people who are resident in the US, and there are demographics which are underrepresented. This will also skew our results.

Names that Give Away Your Age

Some names have certain years where they were exceptionally popular. These names provide quite a lot of information about birth year. Let's look at some of the names with the highest max probability.

Medium Popularity (>10,000 people with the name)

Name	Age with max prob	Prob of most likely age
Katina	49	0.245
Marquita	38	0.233
Ashanti	19	0.250
Miley	13	0.250
Aria	7	0.247

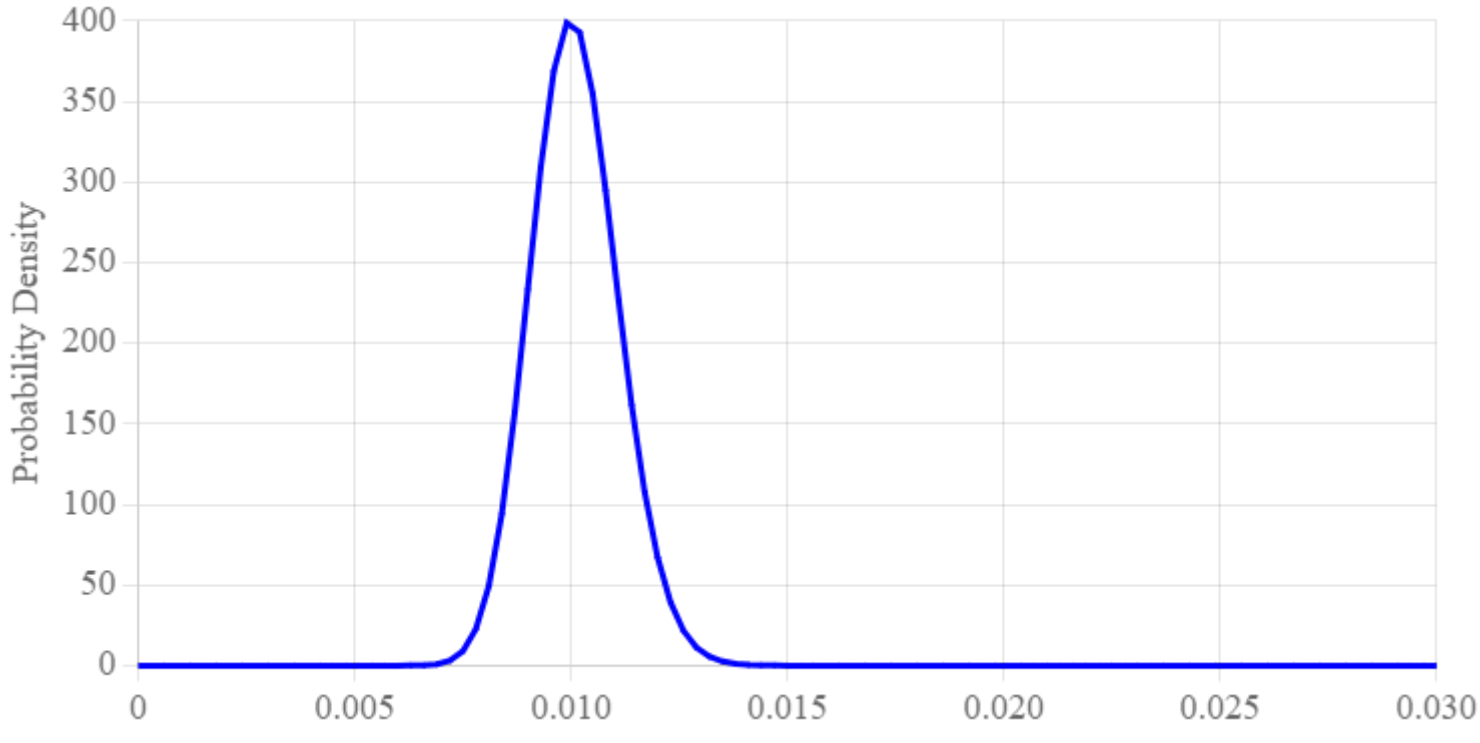
High Popularity (>100,000 people with the name)

Name	Age with max prob	Prob of most likely age
Debbie	62	0.104
Whitney	35	0.098
Chelsea	29	0.103
Aidan	18	0.098
Addison	14	0.112

A search for "Katina 1972" brought up this interesting article about a baby named Katina in a 1972 CBS Soap Opera. Marquita's popularity was likely from a 1983 toothpase add. Ashanti Douglas and Miley Cirus were popular singers in 2002 and 2008 respectively.

Futher Reading

Some names don't seem to have enough data to make good probability estimates. Can we quantify our uncertainty in such probability estimates? For example, if a name has only 10,000 entries in the database, of which only 100 were born in the year 1950, how confident are we that the true probability for 1950 is $\frac{100}{10000} = 0.01$? One way to express our uncertainty would be through a Beta Distribution. In this scenario we could represent our belief in the probability for 1950 as $X \sim \text{Beta}(a = 101, b = 9901)$ reflecting that we have seen 100 people born in 1950 and 9900 people who were not. We can plot that belief, zoomed into the range [0, 0.03]:



We can now ask questions such as, what is the probability that X is within 0.002 of 0.01?

$$
\begin{aligned}
&P(0.008 < X < 0.012) \\
&\quad = P(X < 0.012) - P(X < 0.008) \\
&\quad = F_X(0.012) - F_X(0.008) \\
&\quad = 0.966 - 0.013 \\
&\quad = 0.953
\end{aligned}
$$

Semantically this leads to the claim that, after observing 100 births with a name in 1950, out of 10,000 births with that name over the whole dataset, there is a 95% chance that the probability of someone being born in 1950 is 0.010 ± 0.002.

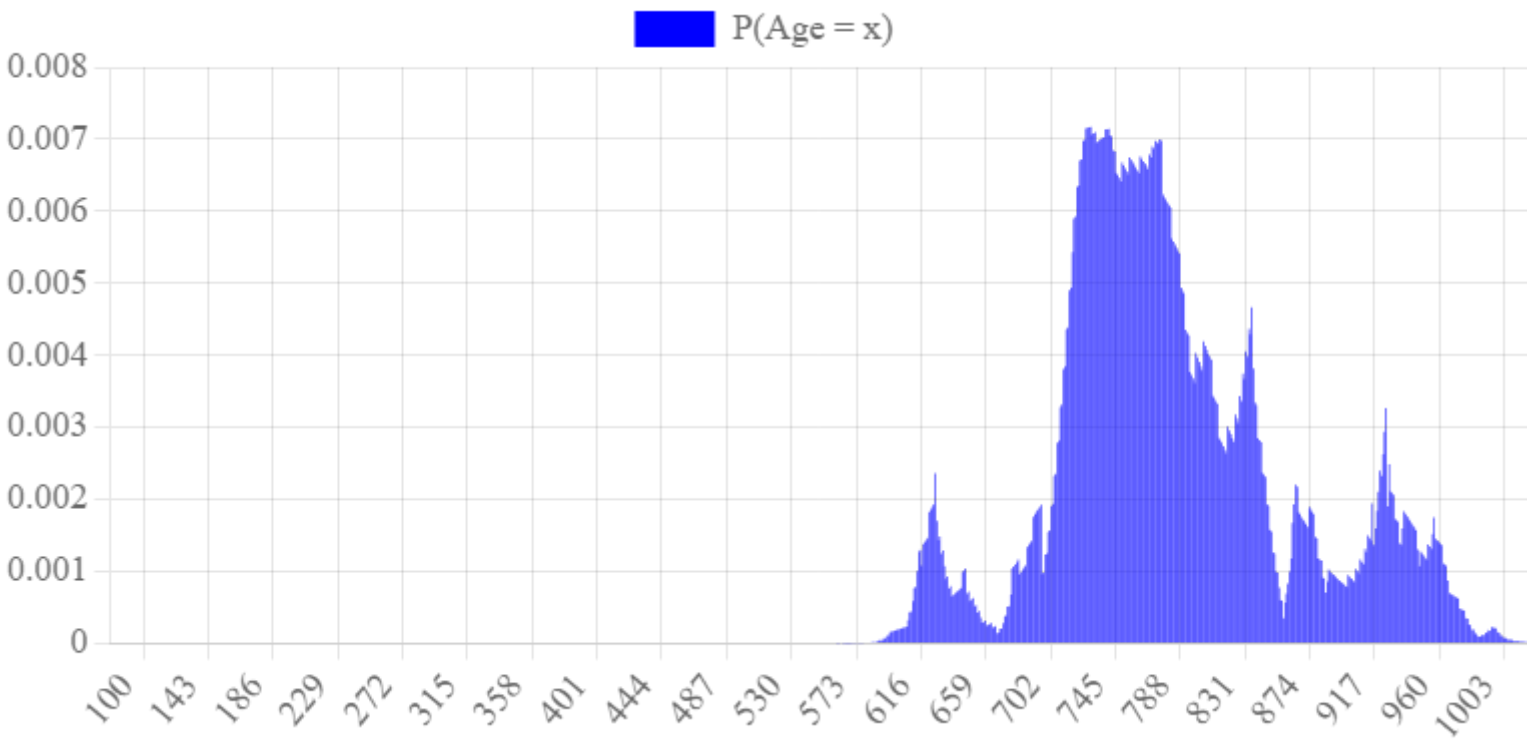
Bayesian Carbon Dating

We are able to know the age of ancient artefacts using a process called carbon dating. This process involves a lot of uncertainty! You observe a measurement of 90% of natural C14 molecules in a sample. What is your belief distribution over the age of the sample? This task requires probabilistic models because we have to think about two random variables together: the age A of the sample, and M the remaining C14 molecules.

Carbon Dating Demo

Imagine you have just taken a sample from your artifact. For the sample size you took, a living organism would have had 1000 molecules of C14. Use this demo to explore the relationship between how much C14 is left and your belief ditribution for how old your artifact is.

Remaining C14: 900



Note: this demo was created in 2023 and the age reported is relative to that year! This chapter only has historical C14 rates from 10,000 years ago and as such is not able to estimate age when there are fewer than 350 molecules of C14 in the sample.

Carbon dating allows us to know the age of things that used to be alive, like dinosaur bones.

Understanding Decay of C14 molecule

All living things have a constant proportion of a radioactive molecule called C14 in them. When living things die those molecules start to decay radioactively. Specifically, the time to decay in years, T, of a single C14 molecule is distributed as an exponential, $T \sim \text{Exp}(\lambda = 1/8267)$ where 8267 is the mean life of C14.

Consider a single C14 molecule. What is the probability that it decays within 750 years?

$$\begin{aligned} P(T \leq 750) &= 1 - e^{-\lambda \cdot 750} && \text{Exp. CDF} \\ &= 1 - e^{-\frac{1}{8267} \cdot 750} \\ &= 0.0867 \end{aligned}$$

That is only for a single molecule. Since C14 molecules decay independently, it is not much harder to think of how many are left out of a larger initial count of C14. A particular sample started with 1000 molecules. What is the probability that exactly 900 are left after 750 years? This is equivalently to the event that 100 molecules have decayed.

$$\begin{aligned} X &\sim \text{Bin}(n = 1000, p = 0.0867) \\ \text{P}(X = 100) &= \binom{1000}{100} 0.0867^{100} (1 - 0.0867)^{900} \\ &\approx 0.0144 \end{aligned}$$

Let's generalize. Define M to be a random variable for the number of molecules left, and A to be the age of the sample. The probability $\text{P}(M = m | A = i)$ of having m remaining C14 molecules given that the artifact is i years old will be equal to $\text{P}(X = n - m)$ where n is the starting number of C14 molecules, $p = 1 - e^{-i/8267}$ and $X \sim \text{Bin}(n, p)$ is the count of decayed C14 molecules.

Inferring Age From C14

You observe a measurement of 900 C14 molecules in a sample. You assume that the sample originally had 1000 C14 molecules when it died. Infer $P(A = i|M = 900)$ where $A = i$ is the event that the sample organism died i years ago. Note that age is a discrete random variable which takes on whole numbers of years. You will need use $P(M = m|A = i)$ from the previous part.

For your prior belief you know that the sample must be between $A = 100$ and $A = 10000$ inclusive and you assume that every year in that range is equally likely.

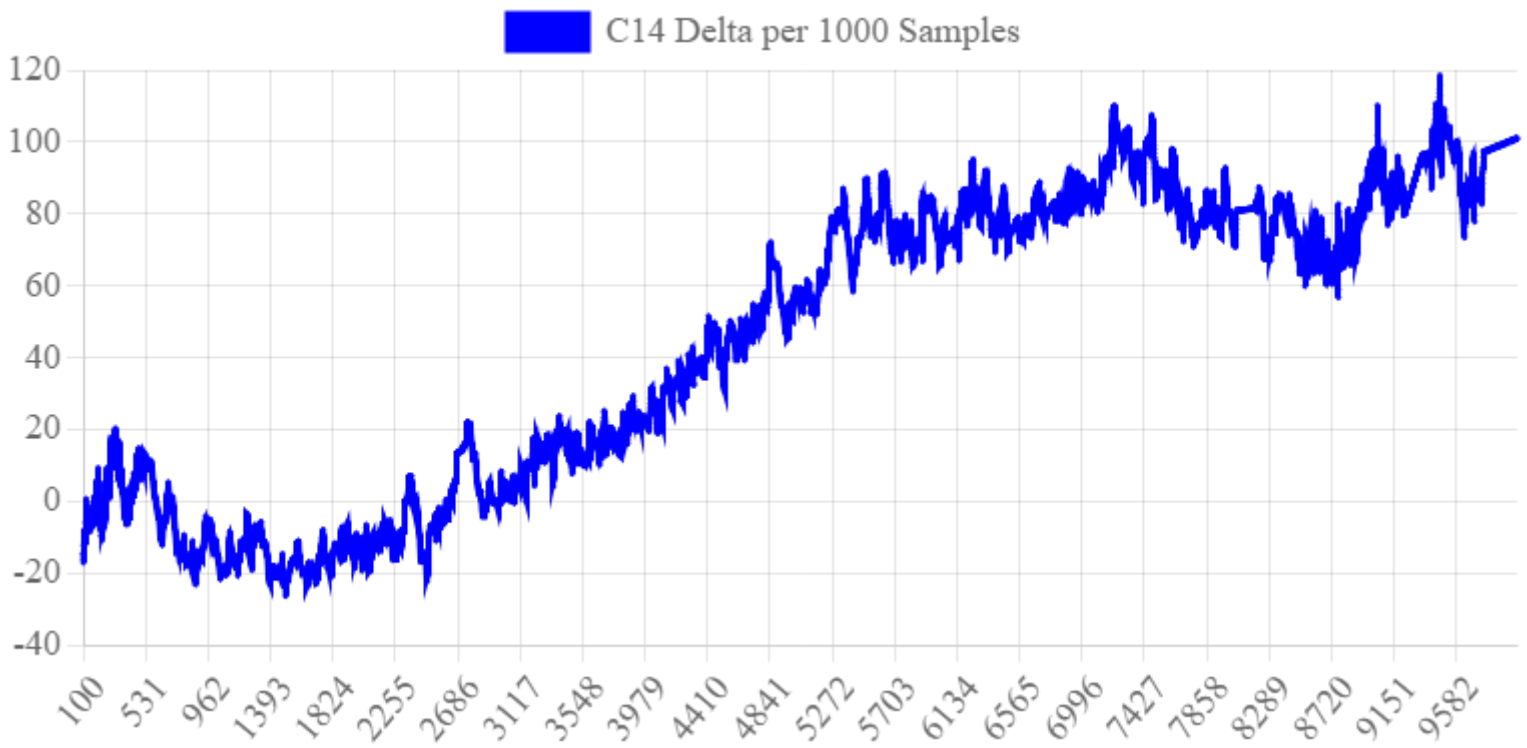
This is a perfect case for Bayes theorem. However instead of updating our belief in an event, like we did in Part 1, we are updating the belief over all the values that a random variable can take on, a process called inference. Here is the generalized version of Bayes' theorem for infering age, A:

$$\begin{aligned} P(A = i|M = 900) &= \frac{\mathrm{P}(M = 900|A = i)\,\mathrm{P}(A = i)}{\mathrm{P}(M = 900)} \\ &= \mathrm{P}(M = 900|A = i) \cdot \frac{\mathrm{P}(A = i)}{\mathrm{P}(M = 900)} \\ &= \mathrm{P}(M = 900|A = i) \cdot K \end{aligned}$$

The critical part of the last line was to recognize that $\frac{\mathrm{P}(A=i)}{\mathrm{P}(M=900)}$ is a constant with respect to i. The term $\mathrm{P}(A = i)$ is constant as our prior over A was uniform. We could compute the value of $\mathrm{P}(M = 900)$ explicitely using the law of total probability. In code this is most easily implemented by computing all values of $\mathrm{P}(M = 900|A = i)$ and normalizing as K will be the value that makes all of the values $\mathrm{P}(A = i|M = 900)$ sum to 1.

Fluctuating History

The amount of C14 in the atmosphere fluctuates over time; it is not a constant baseline! Here is the delta C14 (per 1000 molecules) that you would have found if the object died different number of years ago. To incorporate this information we simply start our binomial with 1000 molecules plus the delta for the year, downloaded from a public dataset [1]:



This offset is archeology theory not probability theory. We include it in this chapter because otherwise our code will give an incorrect preiction. Also, it gives the posterior a really interesting shape (see the demo).

Python Code

The math, derived above, leads to the following python code for a function `inference(m)` which returns the probability mass function for age A, given an observation of m C14 molecules in a sample that should have 1000 molecules, were it alive today. Notice the use of normalization to avoid explicitely computing the prior or $\mathrm{P}(M = m)$ from Bayes Theorem.

```python
from scipy import stats
import math

C14_MEAN_LIFE = 8267

def inference(m = 900):
    """
    Returns a dictionary A, where A[i] contains the
    corresponding probability, P(A = i| M = m).
    m is the number of C14 molecules remaining and i
    is age in years. i is in the range 100 to 10000
    """
    A = {}
    for i in range(100,10000+1):
        A[i] = calc_likelihood(m, i) # P(M = m | A = i)
    # implicitly computes the normalization constant
    normalize(A)
    return A

def calc_likelihood(m, age):
    """
    Computes P(M = m | A = age), the probability of
    having m molecules left given the sample is age
    years old. Uses the exponential decay of C14
    """
    n_original = 1000 + delta_start(age)
    n_decayed = n_original - m
    p_single = 1 - math.exp(-age/C14_MEAN_LIFE)
    return stats.binom.pmf(n_decayed, n_original, p_single)

def normalize(prob_dict):
    # first compute the sum of the probability
    sum = 0
    for key, pr in prob_dict.items():
        sum += pr
    # then divide each probability by that sum
    for key, pr in prob_dict.items():
        prob_dict[key] = pr / sum
    # now the probabilities sum to 1 (aka are normalized)

def delta_start(age):
    """
    The amount of atmospheric C14 is not the same every
    year. If the sample died "age" years ago, then it would
    have started with slightly more, or slightly less than
    1000 C14 molecules. We can look this value up from the
    IntCal database. See the next section!
    """
    return historical_c14_delta[age]
```

Industry Strength Bayesian Carbon Dating

There are other sources of uncertainty in Carbon Dating which we have not considered. For example a common source of uncertainty is laboraty error: if the same sample were sent to a lab several times, different results would come back.

Perhaps the most fascinating extension is modelling "stratigraphic" relationships. Often in archeological sites, you can know relative age of artifacts based on their position in sediment. This requires a joint model of the age of each artifact with the constraint that you *know* some are older than others. Inference can then be performed using a General Inference technique (often MCMC) and will be much more accurate.

Binomial Approximations?

Could we have used an approximation for the binomial PMF calculation? In Bayesian Carbon dating *both* a normal and a poisson approximation are appropriate. The decay binomial $X \sim \text{Bin}(n, p)$ is well approximated by either a Poisson with $\lambda = n \cdot p$ or a Gaussian with $\mu = n \cdot p$ and $\sigma^2 = n \cdot p \cdot (1 - p)$. This could be used to speed up calculations. Let's rework the example where we had $X \sim \text{Bin}(n = 1000, p = 0.0867)$. We computed that $\text{P}(X = 100) = 0.0144$

Poisson Approximation:

$$\begin{aligned} Y &\sim \text{Poi}(\lambda = 86.7) \\ \text{P}(X = 100) &\approx \text{P}(Y = 100) \\ &= \texttt{scipy.stats.poi.pmf}(100, 86.7) \\ &\approx 0.0151 \end{aligned}$$

Normal Approximation:

$$\begin{aligned} Y &\sim \text{N}(\mu = 86.7, \sigma^2 = 79.2) \\ \text{P}(X = 100) &\approx \text{P}(Y > 100.5) - \text{P}(Y > 99.5) \\ &\approx 0.0146 \end{aligned}$$

---

[1] IntCal Historical Atmospheric C14

Digital Vision Test

The Story: This problem was initially posed as a CS109 Final exam problem in Spring 2017. This grew into a collaboration with a former student, Ali Malik, as well as a Stanford Ophthalmology doctor Charles Lin. We realized that it was actually a much more accurate way of measuring visual acuity. The algorithm, which is called the Stanford Acuity Test, or StAT, has since been published as an article for AAAI and covered by Science Magazine and The Lancet. To the best of our knowledge, the algorithm is still the most acurate way to infer ability to see from an optotype based test.

You can find a demo of the Stanford Acuity Test here: https://myeyes.ai/. Look out for the bar icon to see the belief distribution change as the test progresses.

Digital Vision Tests

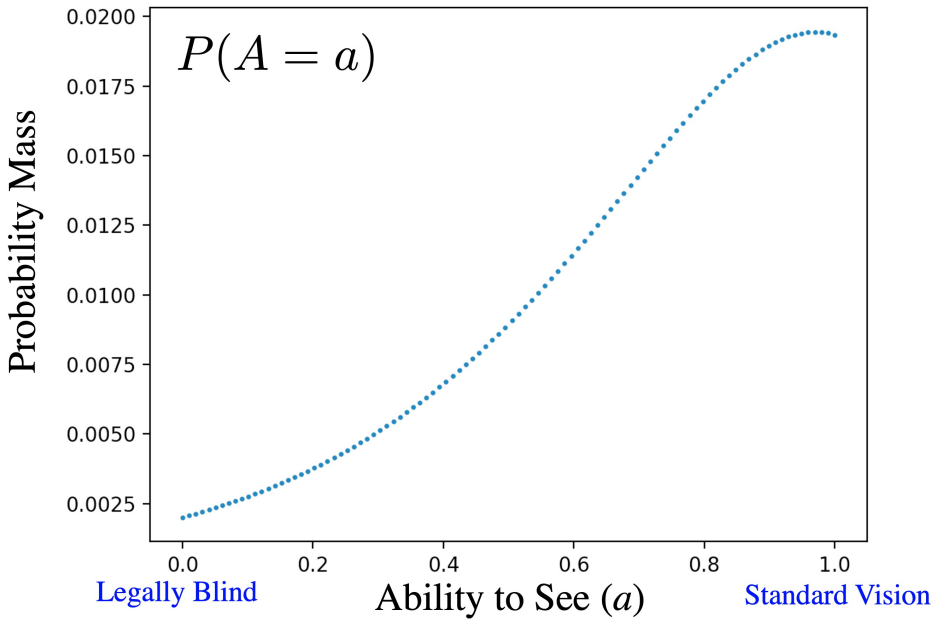
The goal of a digital vision test is to estimate how well a patient can see. You can give the patient a series of vision tests, observe their responses and then based on those responses eventually make your diagnosis. In this chapter we consider the Tumbling E tasks. The patient is presented an E at a chosen font size. The E will be randomly written up, down, left or right and the patient must say which direction it is facing. Their guess will either be correct, or incorrect. The patient will have a series of 20 of these tasks. Vision tests as useful for people who need glasses, but can be critical for folks with eye disease who need to closely monitor for subtle decreases in vision.

There are two primary tasks in a digital vision test: (1) based on the patient responses, infer their ability to see and (2) select the next font size to show to the patient.

How to Represent Ability to See?

Ability is a random variable! We define A to represent the ability of someone to see. A takes on values between 0.0 (representing legal blindness) and 1.0 (representing standard vision). While ability to see is in theory a continuous random variable, we are going to represent ability to see as discretized into one hundreths. As such $A \in \{0.00, 0.01, \ldots, 0.99\}$. As a small aside, visual acuity can be represented in many different units (such as a log based unit called LogMAR). We chose this 0 to 1 scale as it makes the math easier to explain.

The prior probability mass function for A, written as $\mathrm{P}(A = a)$, represents our belief that A takes on the value of a, *before* we have seen any observations about the patient. This prior belief comes from the natural distribution of how well people can see. To make our algorithm most accurate, the prior should best reflect our patient population. Since our eye test is built for doctors in an eye hospital, we used historical data from eye hospital visits to build our prior. Here is $P(A = a)$ as a graph:



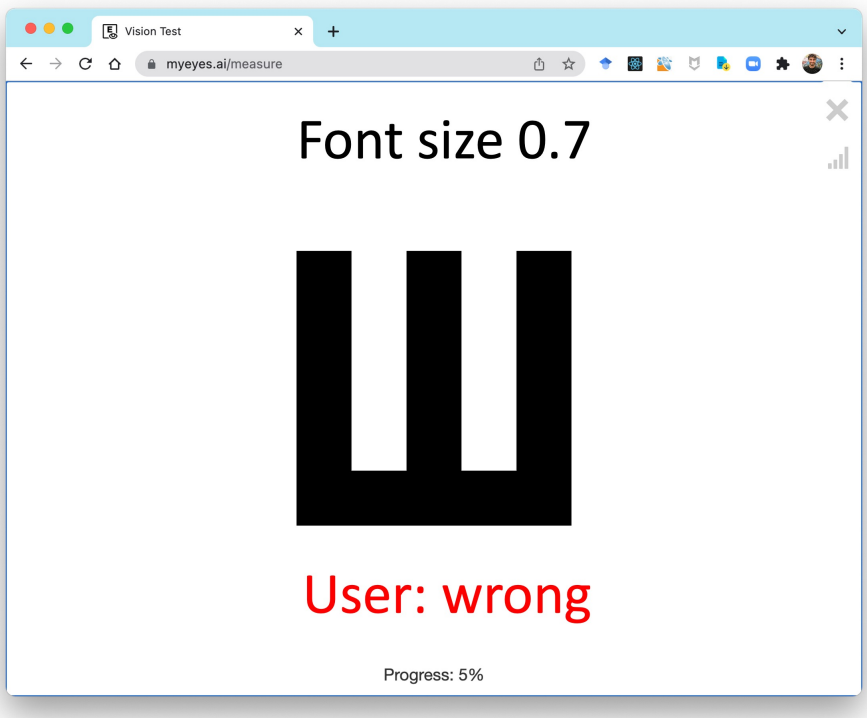
The prior belief on ability to see

Here is that exact same probability mass function written as a table. A table representation is possible because of our choice to discretize A. In code we can access $\mathrm{P}(A = a)$ as a dictionary lookup, `belief[a]` where `belief` stores the whole probability mass function:

a	$\mathrm{P}(A = a)$	a	$\mathrm{P}(A = a)$	a	$\mathrm{P}(A = a)$
0.00	0.0020	0.20	0.0037		⋯
0.01	0.0021	0.21	0.0038	0.81	0.0171
0.02	0.0021	0.22	0.0040	0.82	0.0173
0.03	0.0022	0.23	0.0041	0.83	0.0175
0.04	0.0023	0.24	0.0042	0.84	0.0177
0.05	0.0023	0.25	0.0043	0.85	0.0180
0.06	0.0024	0.26	0.0045	0.86	0.0181
0.07	0.0025	0.27	0.0046	0.87	0.0183
0.08	0.0026	0.28	0.0048	0.88	0.0185
0.09	0.0026	0.29	0.0049	0.89	0.0186
0.10	0.0027	0.30	0.0050	0.90	0.0188
0.11	0.0028	0.31	0.0052	0.91	0.0189
0.12	0.0029	0.32	0.0054	0.92	0.0190
0.13	0.0030	0.33	0.0055	0.93	0.0191
0.14	0.0031	0.34	0.0057	0.94	0.0192
0.15	0.0032	0.35	0.0058	0.95	0.0192
0.16	0.0033	0.36	0.0060	0.96	0.0192
0.17	0.0034	0.37	0.0062	0.97	0.0193
0.18	0.0035	0.38	0.0064	0.98	0.0192
0.19	0.0036	0.39	0.0066	0.99	0.0192

Observations

Once the patient starts the test, you will begin collecting observations. Consider this first observation, obs_1 where the patient was shown a letter with font size 0.7 and answered the question incorrectly:



We can represent this observation as a tuple with font size and correctness. Mathematically this could be written as $\text{obs}_1 = [0.7, \text{False}]$. In code this observation could be stored as a dictionary

```
obs_1 = {
   "font_size":0.7,
   "is_correct":False
}
```

Eventually we will have 20 of these observations: $[\text{obs}_1, \text{obs}_2, \ldots, \text{obs}_{20}]$.

Infering Ability

Our first major task is to write code which can update our probability mass function for A based on observations. First let us consider how to update our belief in ability to see from a single observation, obs (aside: formally this the event that random variable Obs takes on the value obs). We can use Bayes' Theory for random variables:

$$\text{P}(A = a|\text{obs}) = \frac{\text{P}(\text{obs}|A = a)P(A = a)}{\text{P}(\text{obs})}$$

This will be computed inside a `for` loop for each assignment a to ability to see. How can we compute each term in the Bayes' Theorem expression? We already have values for the prior $\text{P}(A = a)$ and we can compute the denominator $\text{P}(\text{obs})$ using the Law of Total Probability:

$$\begin{aligned}\text{P}(\text{obs}) &= \sum_x \text{P}(\text{obs}, A = x) && \text{LOTP} \\ &= \sum_x P(\text{obs}|A = x)P(A = x) && \text{Chain Rule}\end{aligned}$$

Notice how the terms in this new expression for $\text{P}(\text{obs})$ already show up in the numerator of our Bayes' Theorem equation. As such, in code we are going to (1) compute the numerator for every value of a, store it as the value of belief, (2) compute the sum of all of those terms and (3) devide each value of belief by the sum. The process of doing steps 2 and 3 is also known as normalization:

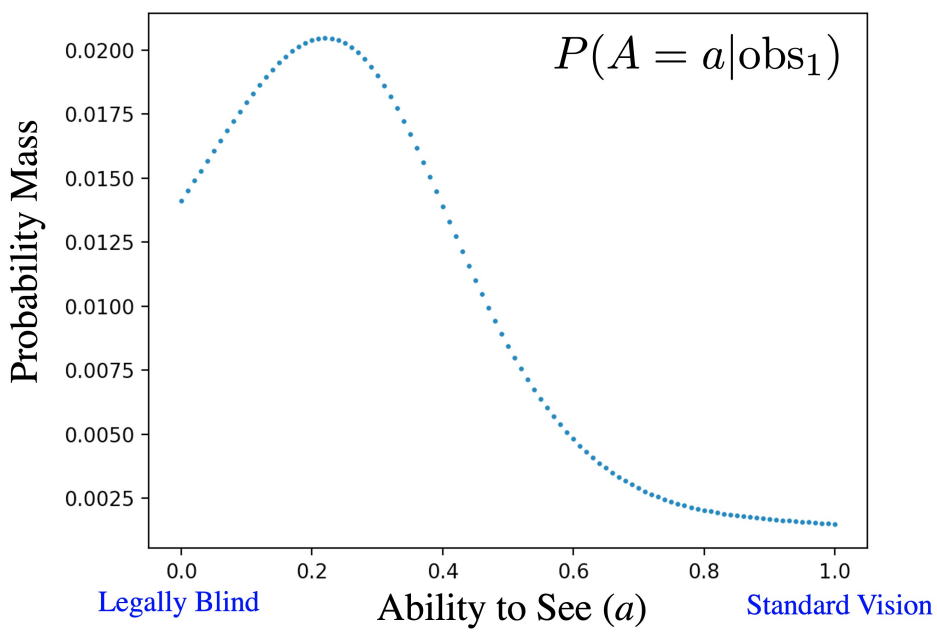
```python
def update_belief(belief, obs):
    """
    Take in a prior belief (stored as a dictionary) for a random
    variable representing how well someone can see based on a single
    observation (obs). Update the belief based using Bayes' Theorem
    """
    # loop over every value in the support of the belief RV
    for a in belief:
        # the prior belief P(A = a)
        prior_a = belief[a]
        # the obs probability P(obs | A = a)
        likelihood = calc_likelihood(a, obs)
        # numerator of Bayes' Theorem
        belief[a] = prior_a * likelihood
    # calculate the denominator of Bayes' Theorem
    normalize(belief)

  def normalize(belief):
    # in place normalization of a belief dictionary
    total = belief_sum(belief)
    for key in belief:
        belief[key] /= total

  def belief_sum(belief):
    # get the sum of probability mass for a discrete belief
    total = 0
    for key in belief:
        total += belief[key]
    return total
```

At this point we have an expression, and corresponding code, to update our belief in ability to see given an observation. However we are **missing** a way to compute $\mathrm{P}(\text{obs}|A = a)$. In our code this expression is the currently undefined `calc_likelihood(a, obs)` function. In the next section we will go over how to compute this "likelihood" function. Before we do so, let's take a look at the result of applying `update_belief` for a patient with the single observation `obs_1` defined above.

`obs_1` says that this patient got a rather large letter (font-size of 0.7) incorrect. As such in our posterior we think they can't see very well, though we have a lot of uncertainty as it has only been one observation. This belief is expressed in our updated probability mass function for A, $\mathrm{P}(A = a|\text{obs}_1)$, called the posterior. Here is what the posterior looks like for `obs_1`. Note that the posterior $\mathrm{P}(A = a|\text{obs}_1)$ is still represented in code as a dictionary, as in the prior, $\mathrm{P}(A = a)$:



The posterior belief on ability to see given a patient incorrectly identified a letter with font size 0.7. It shows a belief that the patient can't see very well.

Likelihood Function

We are not done yet! We have not yet said how we will compute $\mathrm{P}(\text{obs}|A = a)$. In Bayes' Theorem this term is called the "likelihood." The likelihood for our eye exam will be function that returns back probabilities for inputs of a and obs. In python this will be a function `calc_likelihood(a, obs)`. In this function `obs` is a single observation such as `obs_1` described above. Imagine a concrete call to the likelihood function bellow. This call will return back the probability a person who has true ability to see of 0.5 would get a letter of font-size 0.7 incorrect.

```
# get an observation
obs = {
  "font_size":0.7,
  "is_correct":False
}
# calculate likelihood for obs given a, P(obs | A = a)
calc_likelihood(a = 0.5, obs)
```

Before going any further, let's make two critical notes about the likelihood function:

Note 1: When computing the likelihood term, $\mathrm{P}(\text{obs}|A = a)$, we do not have to estimate A as it shows up on the right hand size of the conditional. In the likelihood term we are told *exactly* how well the person can see. Their vision is truly a. Do not be fooled by the fact that a is a (non-random) variable. When computing the likelihood function this varaible will have a numeric value.

Note 2: The variable `obs` represents a single patient interaction. It has two parts: a font-size and a boolean for whether the patient got the letter correct. However, we don't think of font-size as being a random variable. Instead we think of it as a contant which has been fixed by the computer. As such $\mathrm{P}(\text{obs}|A = a)$ can be simplified to $\mathrm{P}(\text{correct}|A = a)$. "correct" is short hand for the event that a random variable Correct takes on the `True` or `False` value correct:

$$\begin{aligned}
&\mathrm{P}(\text{obs}|A = a) \\
&\quad = \mathrm{P}(\text{correct}, f|A = a) && \text{obs is a tuple} \\
&\quad = \mathrm{P}(\text{correct}|A = a) && f \text{ is a constant}
\end{aligned}$$

Defining the likelihood function $\mathrm{P}(\text{correct}|A = a)$ involves more medical and education theory than probability theory. You don't need to know either for this course! But it is still neat to learn and without the likelihood function we won't have complete code. So, let's dive in.

A very practical starting point for the likelihood function for a vision test comes from a classic education model called "Item Response Theory", also known as IRT. IRT *assumes* the probability that a student with ability a gets a question with difficulty d correct is governed by the easy to compute function:

$$\begin{aligned}
&\mathrm{P}(\text{Correct} = \text{True}|a) \\
&\quad = \text{sigmoid}(a - d) && d \text{ is difficulty} \\
&\quad = \frac{1}{1 + e^{-(a-d)}}
\end{aligned}$$

where e is the natural base constant and $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$. The sigmoid function is a handy function which takes in any real valued input and returns a corresponding value in the range $[0, 1]$.

This IRT model introduces a new constant: difficulty of a letter d. How difficult is it to correctly respond to a letter with a given font size? The simplest way to model difficulty, while accounting for the fact that large font sizes are easier than small ones, is to define the difficulty of a letter with font size f to be $d = 1 - f$. Plugging this in:

$$\begin{aligned}
&\mathrm{P}(\text{Correct} = \text{True}|a) \\
&\quad = \text{sigmoid}(a - [1 - f]) \\
&\quad = \text{sigmoid}(a - 1 + f) \\
&\quad = \frac{1}{1 + e^{-(a-1+f)}}
\end{aligned}$$

We now have a complete, if simplistic, liklihood function! In code it would look like this:

```python
def calc_likelihood(a, obs):
    # returns P(obs | A = a) using Item Response Theory
    f = obs["font_size"]
    p_correct_true = sigmoid(a + f - 1)
    if obs["is_correct"]:
        return p_correct_true
    else:
        return 1 - p_correct_true

def sigmoid(x):
    # the classic squashing function. All outputs are [0,1]
    return 1 / (1 + math.exp(-x))
```

Note that Item Response Theory returns the probability that a a patient answers a letter *correctly*. In the code above, notice what we do if the patient instead guesses the letter incorrectly:

$$\mathrm{P}(\mathrm{Correct} = \mathrm{False}|a, f) = 1 - \mathrm{P}(\mathrm{Correct} = \mathrm{True}|a, f)$$

In the published version of the Stanford Acuity Test we extend Item Response Theory in several ways. We have a term for the probability that a patient gets the answer correct by random guessing as well as a term that they make a mistake, aka "slip", even though they know the correct answer. We also observed that a Floored Exponential seems to be a more accurate function than the sigmoid. These extensions are beyond the scope of this chapter as they are not central to the probability insight. For more details see the original paper [1].

Multiple Observations

What if you have multiple observations? For multiple observations the only term that will change will be the likelihood term $\mathrm{P}(\mathrm{Observations}|A = a)$. We assume that each observation is independent, conditioned on ability to see. Formally

$$\mathrm{P}(\mathrm{obs}_1, \ldots, \mathrm{obs}_{20}|A = a) = \prod_i \mathrm{P}(\mathrm{obs}_i|A = a)$$

As such the likelihood of all observations will be the product of the likelihood of each observation on its own. This is equivalent mathematically to calculating the posterior for one observation and calling the posterior your new prior.

The Full Code

Here is the full code for inference of ability to see given observations, minus the user interface functions and the file reading for the prior belief:

```
def main():
    """
    Compute your belief in how well someone can see based
    off an eye exam with 20 questions at different fonts
    """
    belief_a = load_prior_from_file()
    observations = get_observations()
    for obs in observations:
        update_belief(belief_a,obs)
    plot(belief_a)

def update_belief(belief, obs):
    """
    Take in a prior belief (stored as a dictionary) for a random
    variable representing how well someone can see based on a single
    observation (obs). Update the belief based using Bayes' Theorem
    """
    # loop over every value in the support of the belief RV
    for a in belief:
        # the prior belief P(A = a)
        prior_a = belief[a]
        # the obs probability P( obs | A = a)
        likelihood = calc_likelihood(a, obs)
        # numerator of Bayes' Theorem
        belief[a] = prior_a * likelihood
    # calculate the denominator of Bayes' Theorem
    normalize(belief)

def calc_likelihood(a, obs):
    # returns P(obs | A = a) using Item Response Theory
    f = obs["font_size"]
    p_correct = sigmoid(a + f - 1)
    if obs["is_correct"]:
        return p_correct
    else:
        return 1 - p_correct

# ----------- Helper Functions -----------

def sigmoid(x):
    # the classic squashing function. All outputs are [0,1]
    return 1 / (1 + math.exp(-x))

def normalize(belief):
    # in place normalization of a belief dictionary
    total = belief_sum(belief)
    for key in belief:
        belief[key] /= total

def belief_sum(belief):
    # get the sum of probability mass for a discrete belief
    total = 0
    for key in belief:
        total += belief[key]
    return total
```

Chosing the Next Font Size

At this point, we have a way to calculate a probability mass function of our belief in how well the patient can see, at any point in our test. This leaves one more task for us to perform: In a digital eye test, we *select* the next font size to show to the patient. Instead of showing a predetermined set, we should make a choice which is informed by our current belief in how well the patient can see. We were inspired by Thompson Sampling, an algorithm which is able to balance exploring uncertainty and narrowing in on

your most confident belief. When chosing a font size we simply take a sample from our current belief A and then chose the font size that we think a person with ability with that sampled value could see with 80% accuracy. We chose the 80% constant so that the eye test would not be too painful.

One of the neat take aways from this application is that there are many problems where you could take the knowledge learned from this course and improve on the current state of the art! Often the most creative task is to recognize where computer based probability could be usefully applied. Even for eye tests this is not the end of the story. The Stanford Eye Test, which started in CS109, is just a step on the journey to a more accurate digital eye test. There is ./always a better way. Have an idea?

Publications and press coverage:

[1] The Stanford Acuity Test: A Precise Vision Test Using Bayesian Techniques and a Discovery in Human Visual Response. Association for the Advancement of Artificial Intelligence

[2] Digitising the vision test. The Lancet Journal.

[3] Eye, robot: Artificial intelligence dramatically improves accuracy of classic eye exam. Science Magazine.

Special thanks to Ali Malik who co-invented the Stanford Acuity Test.

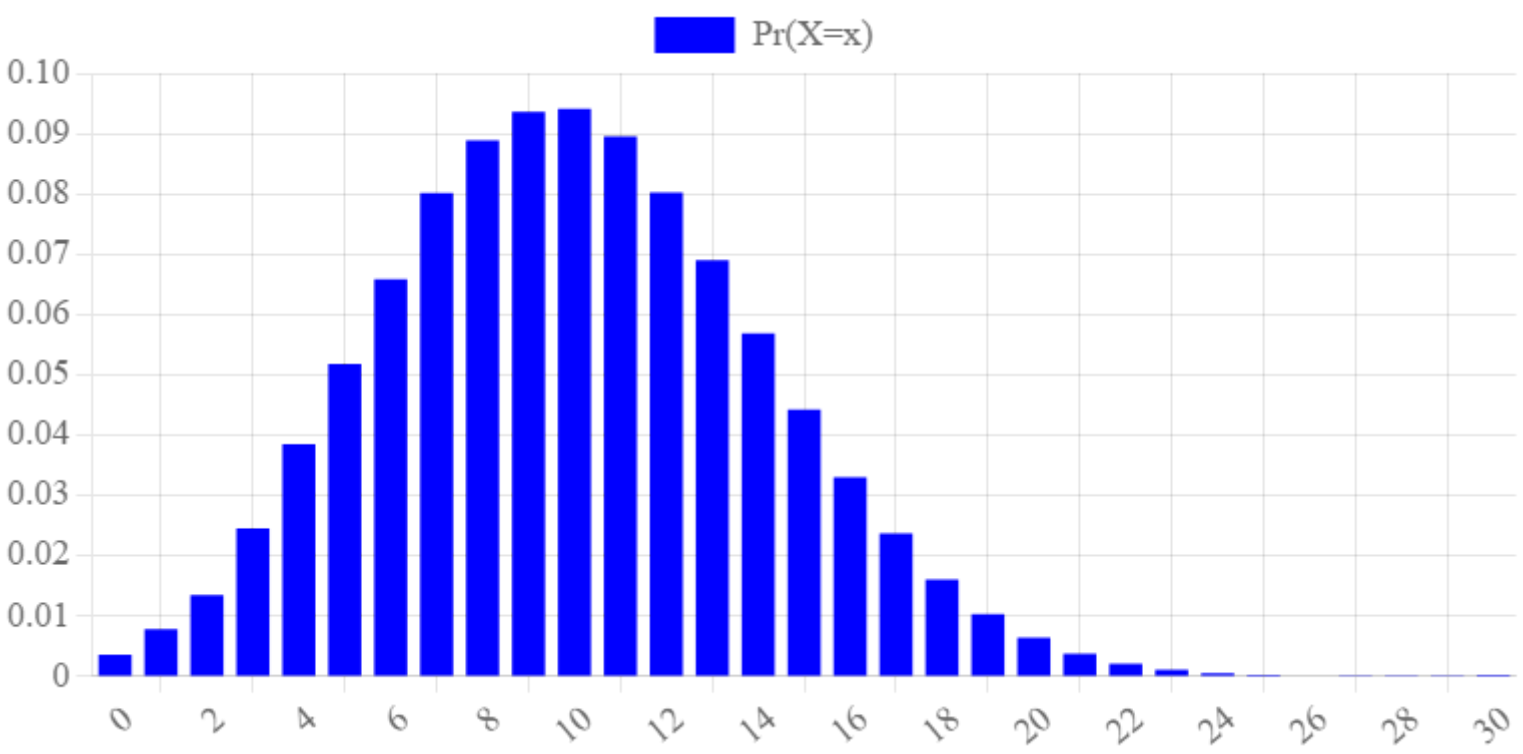
Bridge Card Game

Bridge is one of the most popular collaborative card games. It is played with four players in two teams. A few interesting probability problems come up in this game. You do not need to know the rules of bridge to follow this example. I focus on a set of probability problems which are most important for game strategy.

Distribution of Hand Strength

The way folks play bridge is that they make a calculation about their "hand strength" and then make decisions based off that number. The strength of your hand is a number which is equal to 4 times the number of "aces", 3 times the number of "kings", 2 times the number of "queens" and 1 times the number of "jacks" in your hand. No other cards contribute to your hand strength. Lets consider your hand strength to be a random variable and compute its distribution. It seems complex to compute by hand -- but perhaps we could run a simulation? Here we simulate a million deals of bridge hands and calculate the hand strengths. Let X be the strength of a hand. From the Definition of Probability:

$$\mathrm{P}(X = x) \approx \frac{\mathrm{count}(x)}{100000}$$



Wait! Is that a Poisson?

If you pay very close attention might notice that this PMF looks a lot like a poisson PMF with rate $\lambda = 10$. There is a nice explanation for why the rate might be 10. Let H be the value of your hand. Let X_i be the points of the ith card in your hand, which has 13 cards. $H = \sum_{i=1}^{13} X_i$.

First we compute $\mathrm{E}[X_i]$, the expectation of points for the ith card in your hand — without considering the other cards . A card can take on four non zero values $X_i \in \{1, 2, 3, 4\}$. For each value there are four cards out of 52 with that value, eg $\mathrm{P}(X_i = 1) = \frac{4}{52}$. Thus

$$\begin{aligned}
\mathrm{E}[X_i] &= \sum_x x \cdot \mathrm{P}(X_i = x) \\
&= (1 + 2 + 3 + 4)\frac{1}{13} \\
&= \frac{10}{13}
\end{aligned}$$

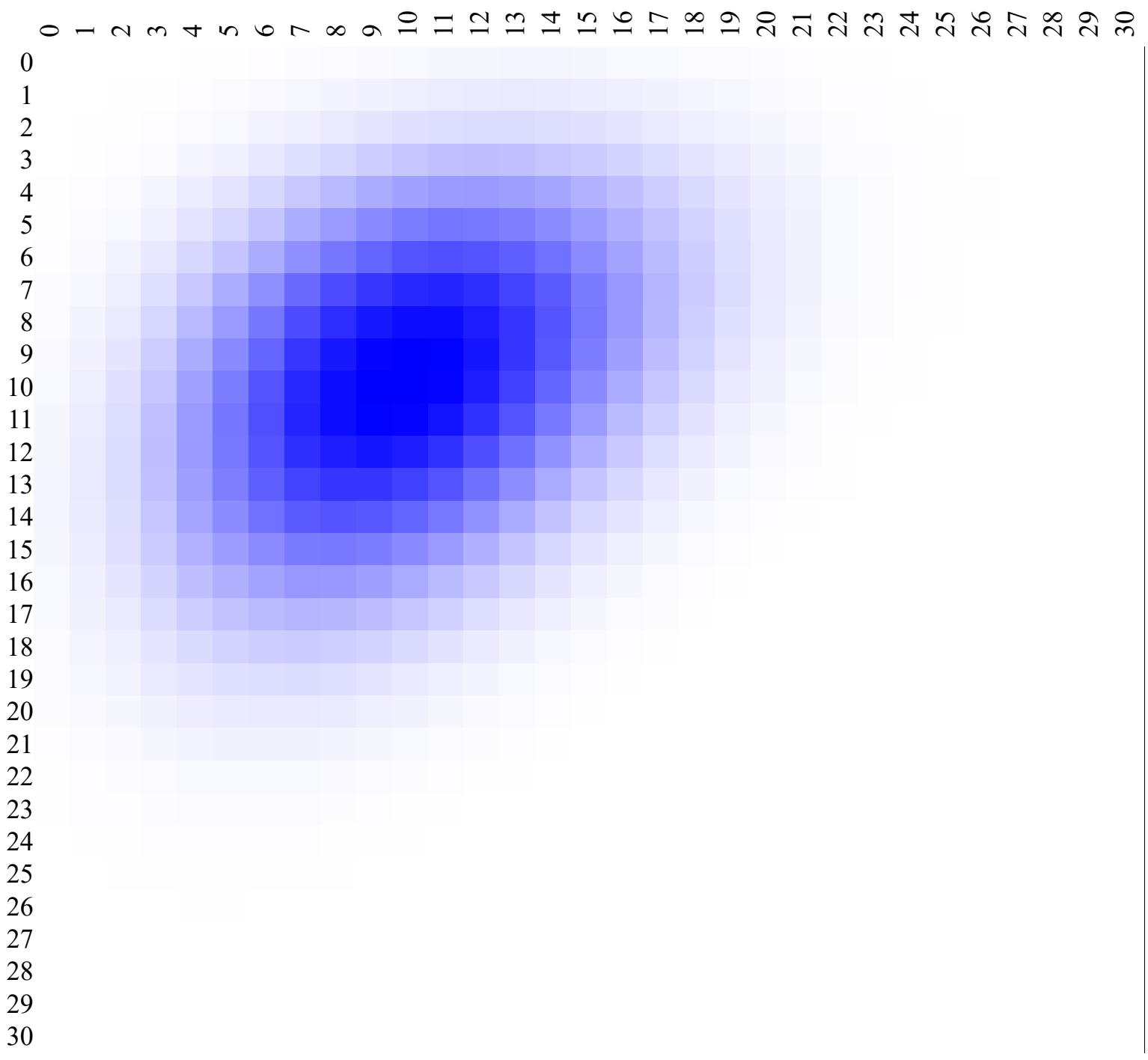
We can then calculate $\mathrm{E}[H]$ by using the fact that the expectation of the sum of random variables is the sum of expectations, regardless of independence:

$$
\begin{aligned}
\mathrm{E}[H] &= \sum_{i=1}^{13} \mathrm{E}[X_i] \\
&= 13 \cdot \mathrm{E}[X_i] \\
&= 13 \cdot \frac{10}{13} \\
&= 10
\end{aligned}
$$

Saying that H is approximately $\sim \mathrm{Poi}(\lambda = 10)$ is an interesting claim. It suggests that points in a hand come at a constant rate, and that the next point in your hand is independent of when you got your last point. Of course this second part of the assumption is mildly violated. There are a fixed set of cards so getting one card changes the probabilities of others. For this reason the poisson is a close, but not perfect approximation.

Joint Distribution of Hand Strength Among Two Hands

It doesn't just matter how strong your hand is, but the relative strength of your hand and your partners hand (recall that in Bridge you play with a partner). We know that the two hands are not independent of each other. If I tell you that your partner has a strong hand, that means there are fewer "high value" cards that can be in your hand, and as such my belief in your strength has changed. If you think about each player's hand strength as a random variable, we care about the joint distribution of hand strength. In the joint distribution bellow the x-axis is your partner's hand strength and on the y-axis is your hand strength. The value is $\mathrm{P}(\mathrm{Partner} = x, \mathrm{YourPoints} = y)$. This joint distribution was calculated by simulating a million randomly dealt hands:

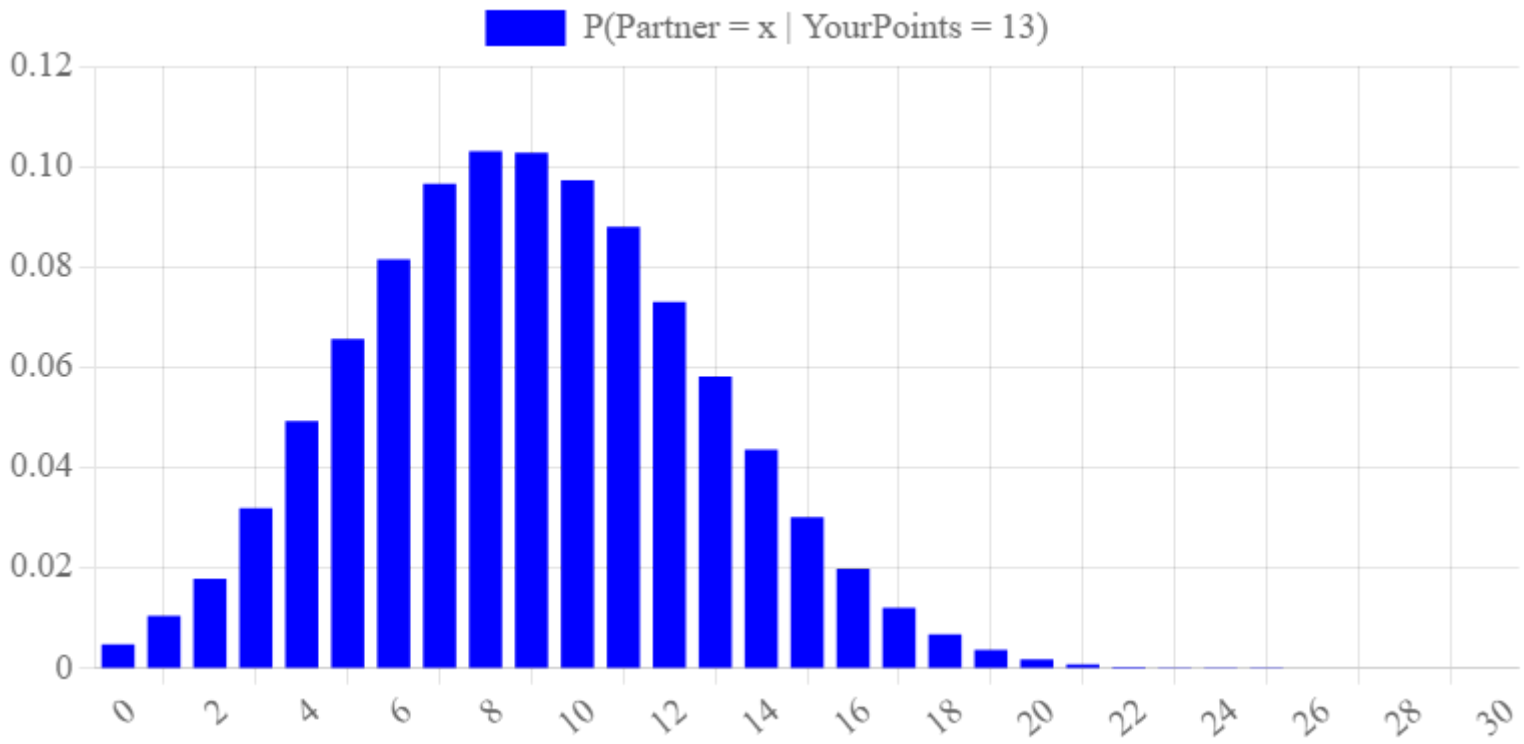


From this joint distribution we can compute conditional probabilities. For example we can compute the conditional distribution of your partner's points given your points using lookups from the joint:

$$
\begin{aligned}
&\mathrm{P}(\text{Partner} = x | \text{YourPoints} = y) \\
&= \frac{\mathrm{P}(\text{Partner} = x, \text{YourPoints} = y)}{\mathrm{P}(\text{YourPoints} = y)} && \text{Cond. Prob.} \\
&= \frac{\mathrm{P}(\text{Partner} = x, \text{YourPoints} = y)}{\sum_z \mathrm{P}(\text{Partner} = z, \text{YourPoints} = y)} && \text{LOTP}
\end{aligned}
$$

Here is a working demo of the result

Your points: 13



Distribution of Suit Splits

When playing the game there are many times when one player will know *exactly* how many cards there are of a certain suit between their two opponents hands (call the opponents A and B). However, the player won't know the "split": how many of that particular suit are in opponent A's hand and how many cards of that suit are in opponent B's hand.

Both opponents have equal sized hands with k cards left. Across the two hands there are a known number of cards of a particular suit (eg spades) n, and you want to know how many are in one hand and how many are in the other. A split is represented as a tuple. For example $(0, 5)$ would mean 0 cards of the suit in opponent A's hands and 5 in opponent B's. Feel free to chose specific values for k and n:

k, the number of cards in each player's hand: 13

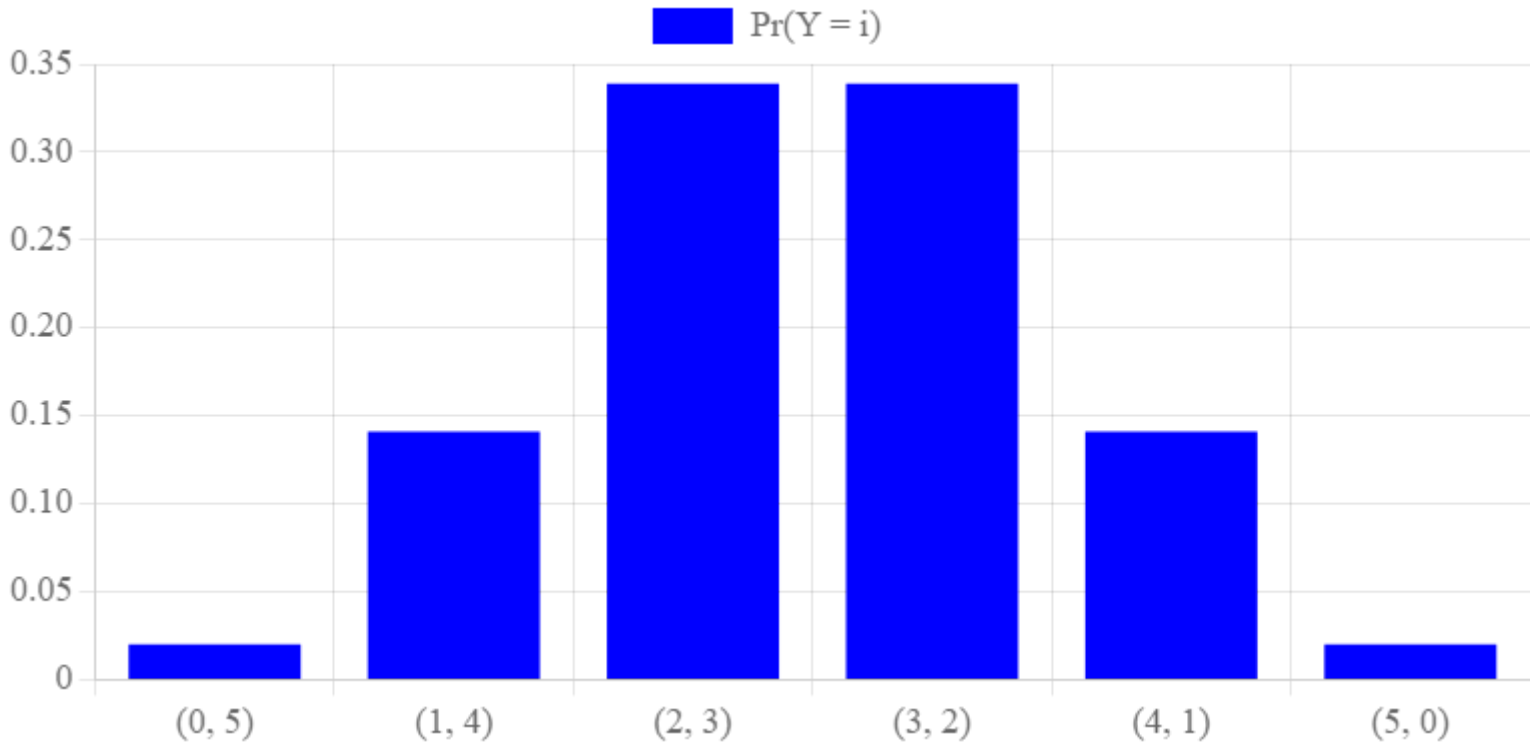
n, the number of cards of particular suit among the two hands: 5

A few notes: If there are k cards in each of the 2 hands there are $2k$ cards total bewteen the two players. At the start of a game of bridge $k = 13$. It must be the case that $n \leq 2k$ because you can't have more cards of the suit left than number of cards! If there are n of a suit, then there are $2k - n$ of other suits. This problem assumes that the cards are properly shuffled.

Probability of different splits of the suit:
Let Y be a random variable representing the number of the suit in opponent A's hand. We can calculate the probability that Y equals different values i by counting equally likely outcomes.

$$
\mathrm{P}(Y = i) = \frac{\binom{n}{i} \cdot \binom{2 \cdot k - n}{k - i}}{\binom{2 \cdot k}{k}}
$$

Each outcome in the sample set is a chosen set of k distinct cards to be dealt to one player (out of the $2k$ cards). To create an outcome in the event space we first chose the i cards from the n cards of the given suit. We then chose $k - i$ from the cards of other suits. For k =13 and n =5 here is the PMF over splits:

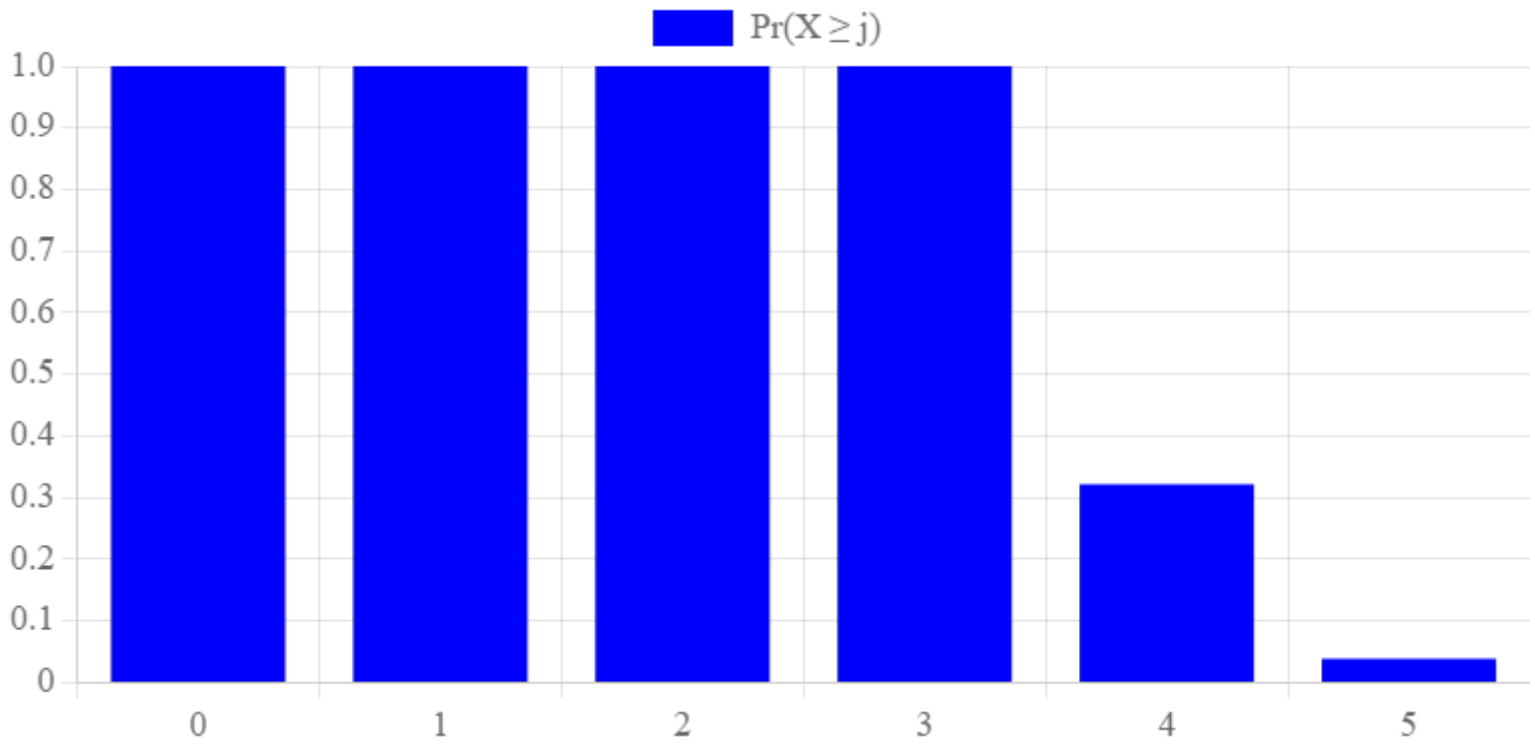


If we want to think about the probability of a given split, it is sufficient to chose one hand (call it "hand one"). If I tell you how many of a suit are in one hand, you can automatically figure out how many of the suit are in the other hand: recall that the number of the suit sums to n.

Probability that either hand has at least j cards of suit

Let X be a random variable representing the highest number of cards of the suit in *either* hand. We can calculate the probability by using probability of or.

$$\mathrm{P}(X \geq j) = 1 - \sum_{i=n-j+1}^{j-1} \mathrm{P}(Y = i)$$

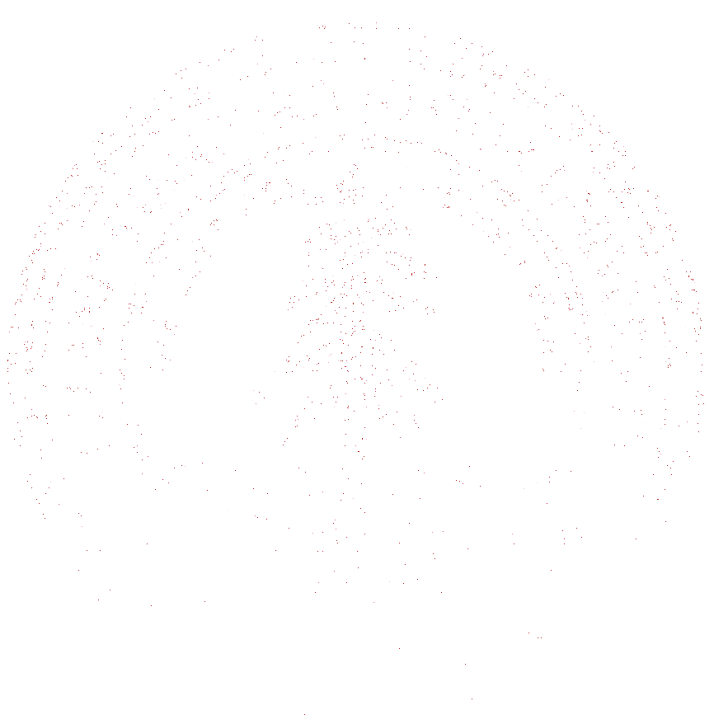


CS109 Logo

To generate the CS109 logo, we are going to throw half a million darts at a picture of the Stanford seal. We only keep the pixels that are hit by at least one dart. Each dart has it's x-pixel and y-pixel chosen at random from gaussian distributions. Let X be a random variable which represent the x-pixel, Y be a random variable which represents the y-pixel and S be a constant that equals the size of the logo (its width is equal to its height). $X \sim \mathcal{N}\left(\frac{S}{2}, \frac{S}{2}\right)$ and $Y \sim \mathcal{N}\left(\frac{S}{3}, \frac{S}{5}\right)$

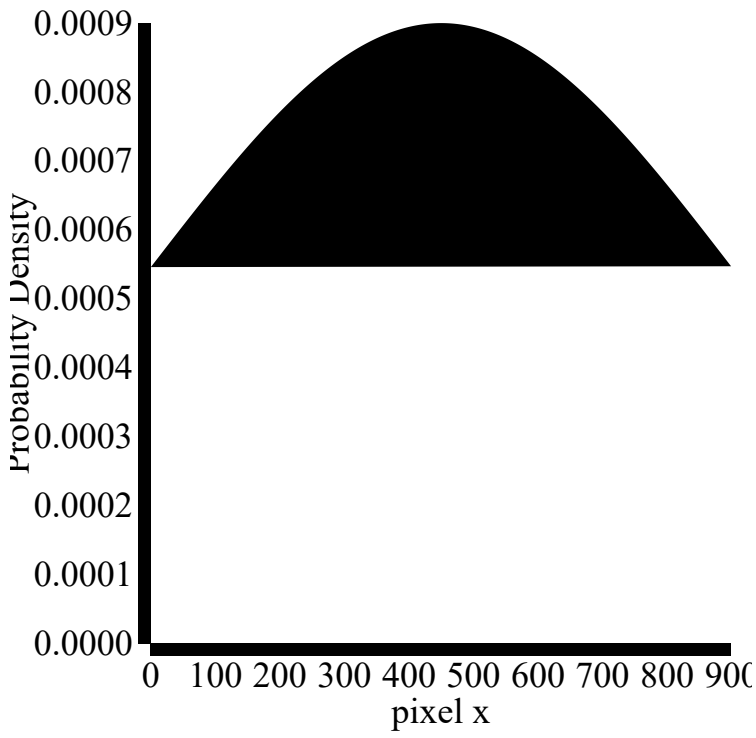
Darts thrown: 7000

Dart Results

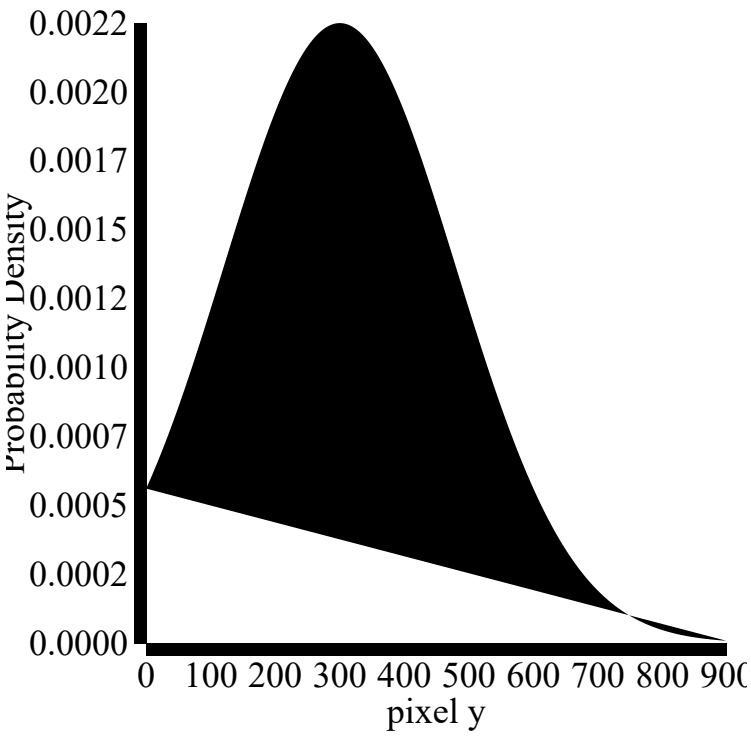


Dart Probability Density

X Distribution



Y Distribution



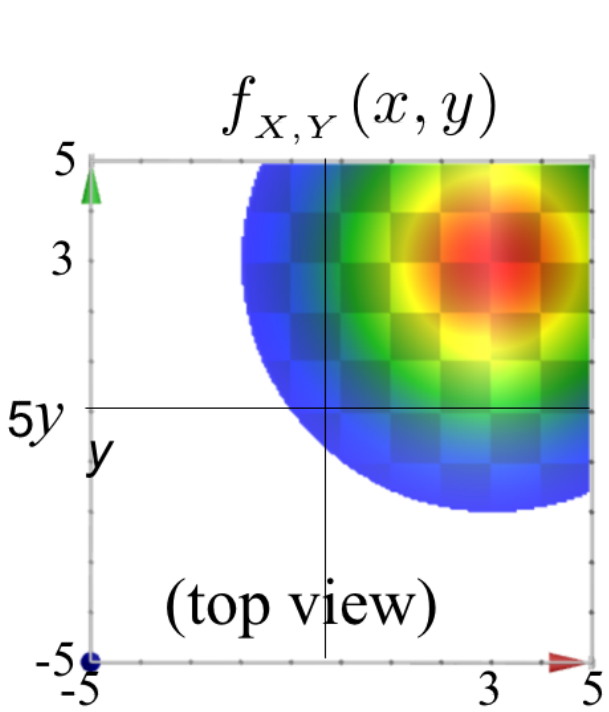
Tracking in 2D

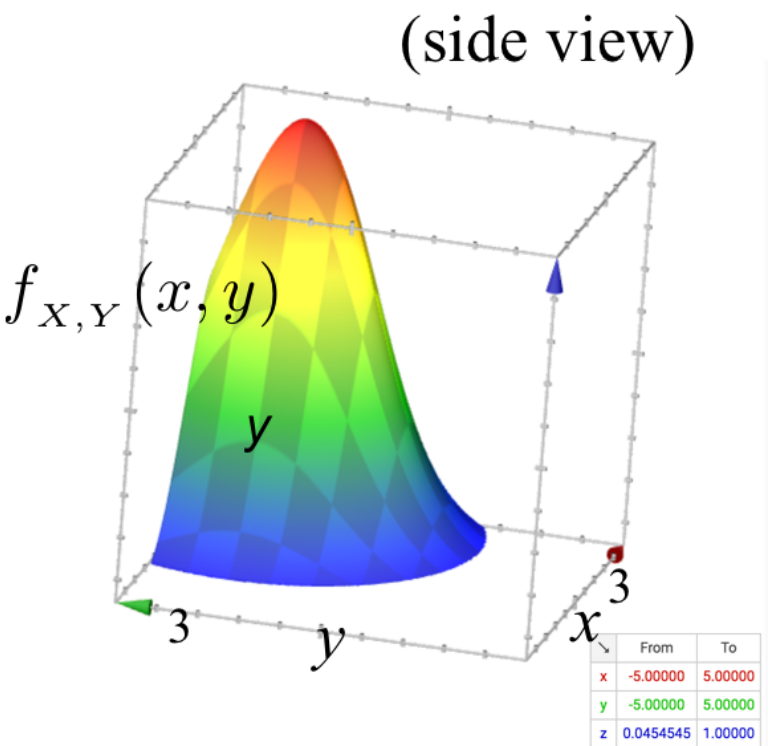
Warning: After learning about joint distributions, and about inference, you have all the technical abilities necessary to follow this example. However this is very very difficult. Primarily because you need to understand three complex things at the same time: (1) how to represent a continuous joint distribution, (2) inference in probabilistic models and (3) a rather complex probability of observation calculation.

In this example we are going to explore the problem of tracking an object in 2D space. The object exists at some (x, y) location, however we are not sure exactly where! Thus we are going to use random variables X and Y to represent location.

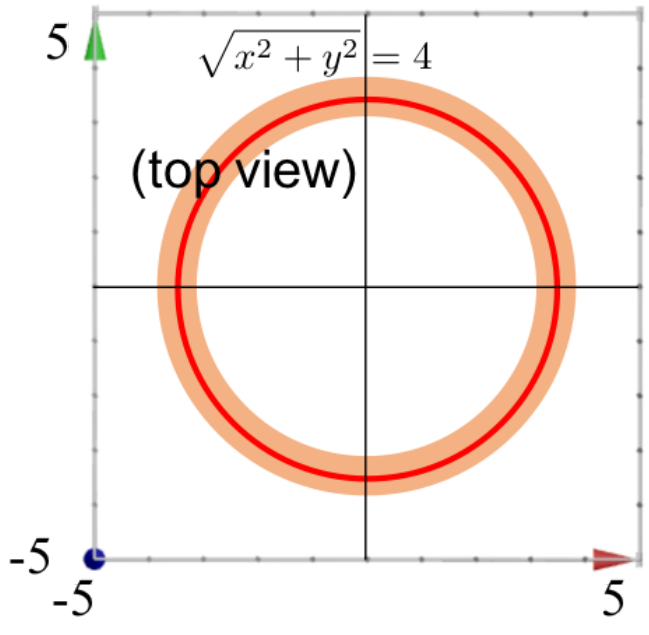
$$
\begin{aligned}
f(X = x, Y = y) &= f(X = x) \cdot f(Y = y) && \text{In the prior X and Y are independen} \\
&= \frac{1}{\sqrt{2 \cdot 4 \cdot \pi}} \cdot e^{-\frac{(x-3)^2}{2 \cdot 4}} \cdot \frac{1}{\sqrt{2 \cdot 4 \cdot \pi}} \cdot e^{-\frac{(y-3)^2}{2 \cdot 4}} && \text{Using the PDF equation for normals} \\
&= K_1 \cdot e^{-\frac{(x-3)^2+(y-3)^2}{8}} && \text{All constants are put into } K_1
\end{aligned}
$$

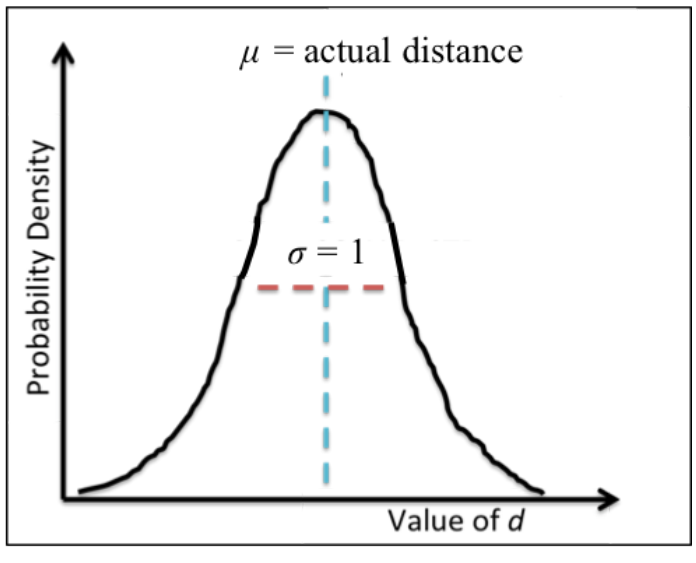
This combinations of normals is called a bivariate distribution. Here is a visualization of the PDF of our prior.





The interesting part about tracking an object is the process of updating your belief about it's location based on an observation. Let's say that we get an instrument reading from a sonar that is sitting on the origin. The instrument reports that the object is 4 units away. Our instrument is not perfect: if the true distance was t units away, than the instrument will give a reading which is normally distributed with mean t and variance 1. Let's visualize the observation:





Based on this information about the noisiness of our prior, we can compute the conditional probability of seeing a particular distance reading D, given the true location of the object X, Y. If we knew the object was at location (x, y), we could calculate the true distance to the origin $\sqrt{x^2+y^2}$ which would give us the mean for the instrument Gaussian:

$$
\begin{aligned}
f(D=d|X=x,Y=y) &= \frac{1}{\sqrt{2\cdot 1\cdot \pi}}\cdot e^{-\frac{(d-\sqrt{x^2+y^2})^2}{2\cdot 1}} && \text{Normal PDF where } \mu=\sqrt{x^2+y^2} \\
&= K_2\cdot e^{-\frac{(d-\sqrt{x^2+y^2})^2}{2\cdot 1}} && \text{All constants are put into } K_2
\end{aligned}
$$

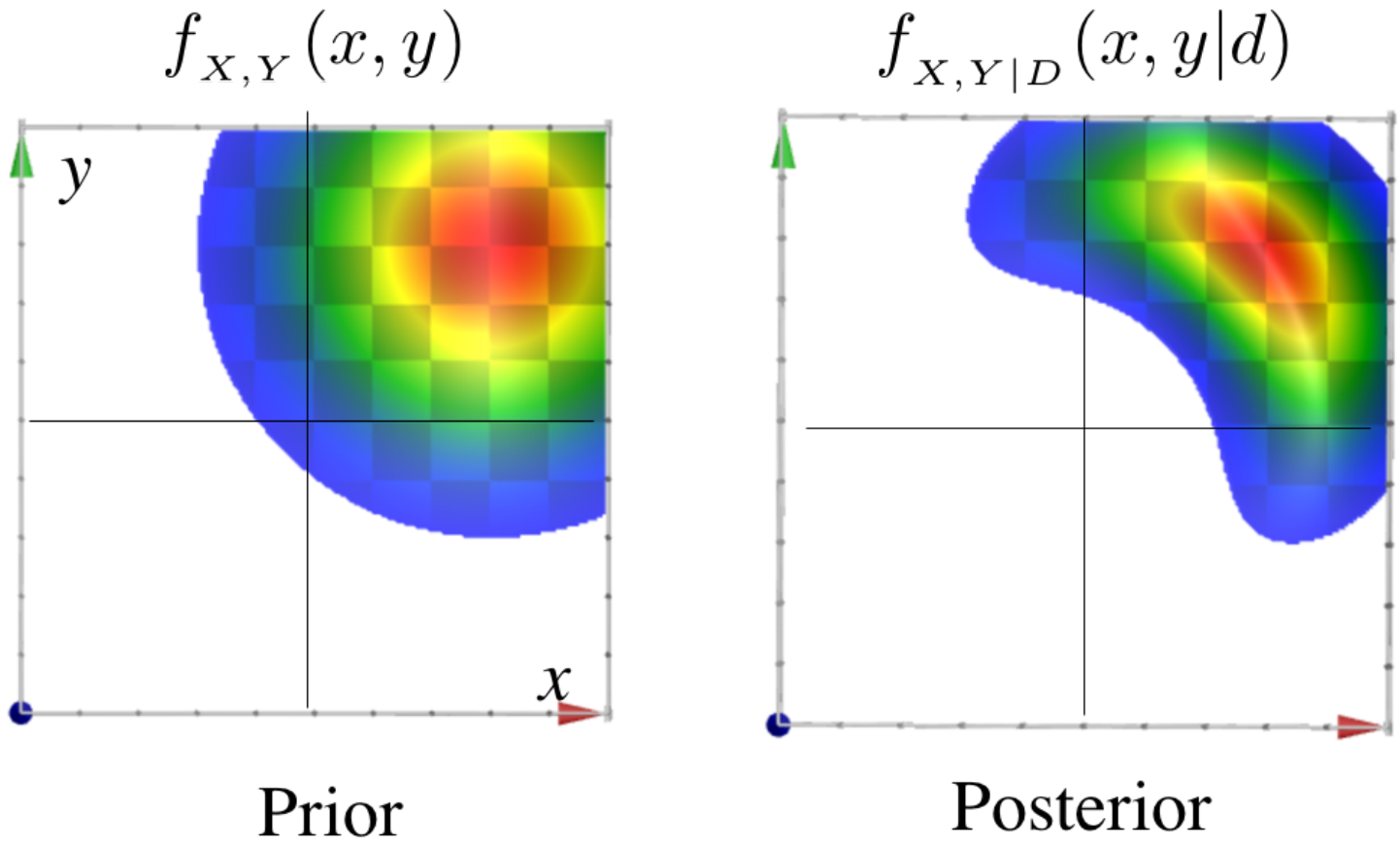
How about we try this out on actual numbers. How much more likely is an instrument reading of 1 compared to 2, given that the location of the object is at (1, 1)?

$$
\begin{aligned}
\frac{f(D=1|X=1,Y=1)}{f(D=2|X=1,Y=1)} &= \frac{K_2\cdot e^{-\frac{(1-\sqrt{1^2+1^2})^2}{2\cdot 1}}}{K_2\cdot e^{-\frac{(2-\sqrt{1^2+1^2})^2}{2\cdot 1}}} && \text{Substituting into the conditional PDF of D} \\
&= \frac{e^0}{e^{-1/2}} \approx 1.65 && \text{Notice how the } K_2 \text{ cancel out}
\end{aligned}
$$

At this point we have a prior belief and we have an observation. We would like to compute an updated belief, given that observation. This is a classic Bayes' formula scenario. We are using joint continuous variables, but that doesn't change the math much, it just means we will be dealing with densities instead of probabilities:

$$
\begin{aligned}
& f(X=x,Y=y|D=4) \\
&= \frac{f(D=4|X=x,Y=y)\cdot f(X=x,Y=y)}{f(D=4)} && \text{Bayes using densities} \\
&= \frac{K_1\cdot e^{-\frac{[4-\sqrt{x^2+y^2})^2]}{2}}\cdot K_2\cdot e^{-\frac{[(x-3)^2+(y-3)^2]}{8}}}{f(D=4)} && \text{Substitute} \\
&= \frac{K_1\cdot K_2}{f(D=4)}\cdot e^{-\left[\frac{[4-\sqrt{x^2+y^2})^2]}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} && f(D=4) \text{ is a constant w.r.t. } (x,y) \\
&= K_3\cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} && K_3 \text{ is a new constant}
\end{aligned}
$$

Wow! That looks like a pretty interesting function! You have successfully computed the updated belief. Let's see what it looks like. Here is a figure with our prior on the left and the posterior on the right:



How beautiful is that! Its like a 2D normal distribution merged with a circle. But wait, what about that constant! We do not know the value of K_3 and that is not a problem for two reasons: the first reason is that if we ever want to calculate a relative probability of two locations, K_3 will cancel out. The second reason is that if we really wanted to know what K_3 was, we could solve for it. This math is used every day in millions of applications. If there are multiple observations the equations can get truly complex (even worse than this one). To represent these complex functions often use an algorithm called particle filtering.

Beta Distribution

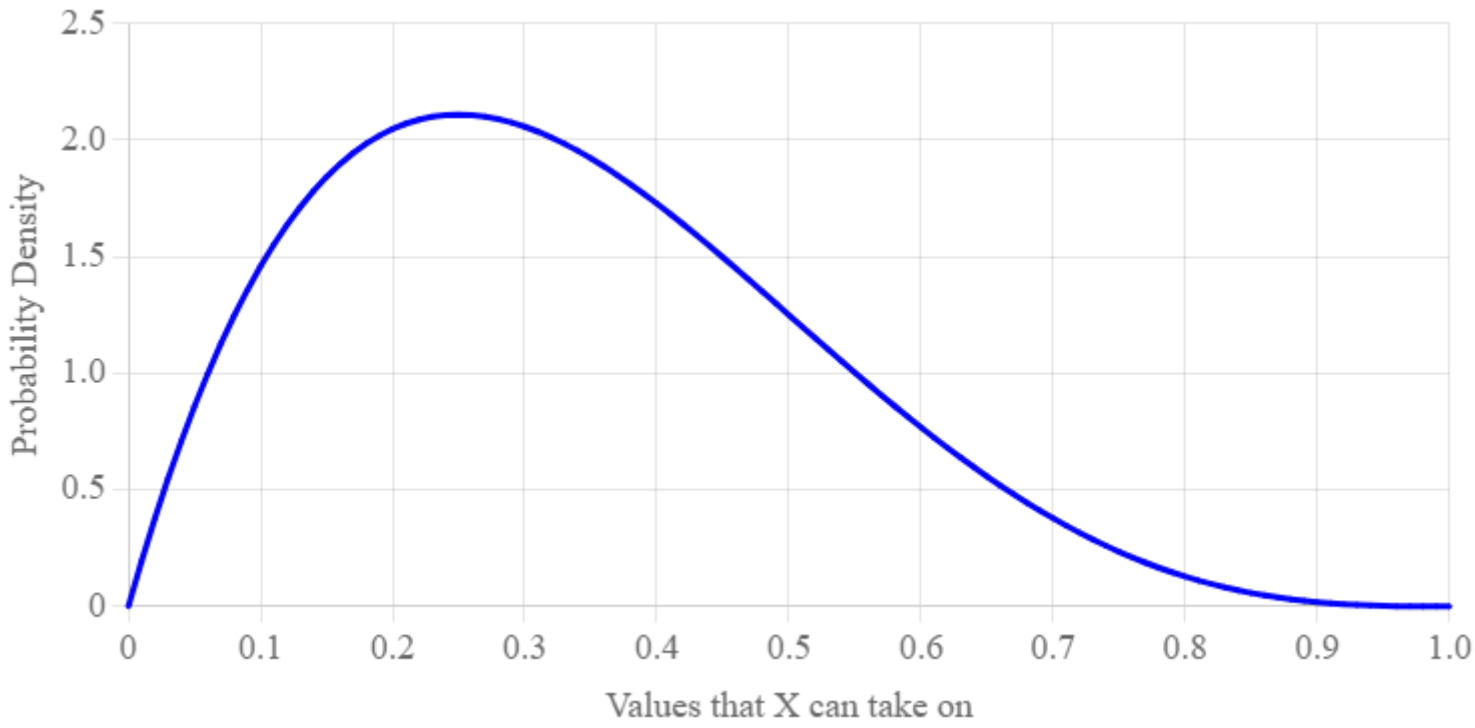
The Beta distribution is the distribution most often used as the distribution of probabilities. In this section we are going to have a very meta discussion about how we represent probabilities. Until now probabilities have just been numbers in the range 0 to 1. However, if we have uncertainty about our probability, it would make sense to represent our probabilities as random variables (and thus articulate the relative likelihood of our belief).

Beta Random Variable

Notation:	$X \sim \text{Beta}(a, b)$
Description:	A belief distribution over the value of a probability p from a Binomial distribution after observing $a-1$ successes and $b-1$ fails.
Parameters:	$a > 0$, the number successes + 1 $b > 0$, the number of fails + 1
Support:	$x \in [0, 1]$
PDF equation:	$f(x) = B \cdot x^{a-1} \cdot (1-x)^{b-1}$
CDF equation:	No closed form
Expectation:	$\text{E}[X] = \frac{a}{a+b}$
Variance:	$\text{Var}(X) = \frac{ab}{(a+b)^2(a+b+1)}$

PDF graph:

Parameter a: 2 Parameter b: 4



What is your Belief in p After 9 Heads in 10 Flips?

Imagine we have a coin and we would like to know its true probability of coming up heads, p. We flip the coin 10 times and observe 9 heads and 1 tail. What is your belief in p based off this evidence? Using the definition of probability we could guess that $p \approx \frac{9}{10}$. That number is a very rough estimate, especially since it is only based off 10 coin flips. Moreover the "point-value" $\frac{9}{10}$ does not have the ability to articulate how uncertain it is.

Could we instead have a random variable for the true probability? Formally, let X represent the true probability of the coin coming up heads. We don't use the symbol P for random variables, so X will have to do. If $X = 0.7$ then the probability of heads is 0.7. X must be a continuous random variable with support $[0, 1]$ since probabilities are continuous values which must be between 0 and 1.

Before flipping the coin, we could say that our belief about the coin's heads probability is uniform: $X \sim \text{Uni}(0, 1)$. Let H be a random variable for the number of heads and let T be a random variable for the number of tails observed. What is $\text{P}(X = x | H = 9, T = 1)$?

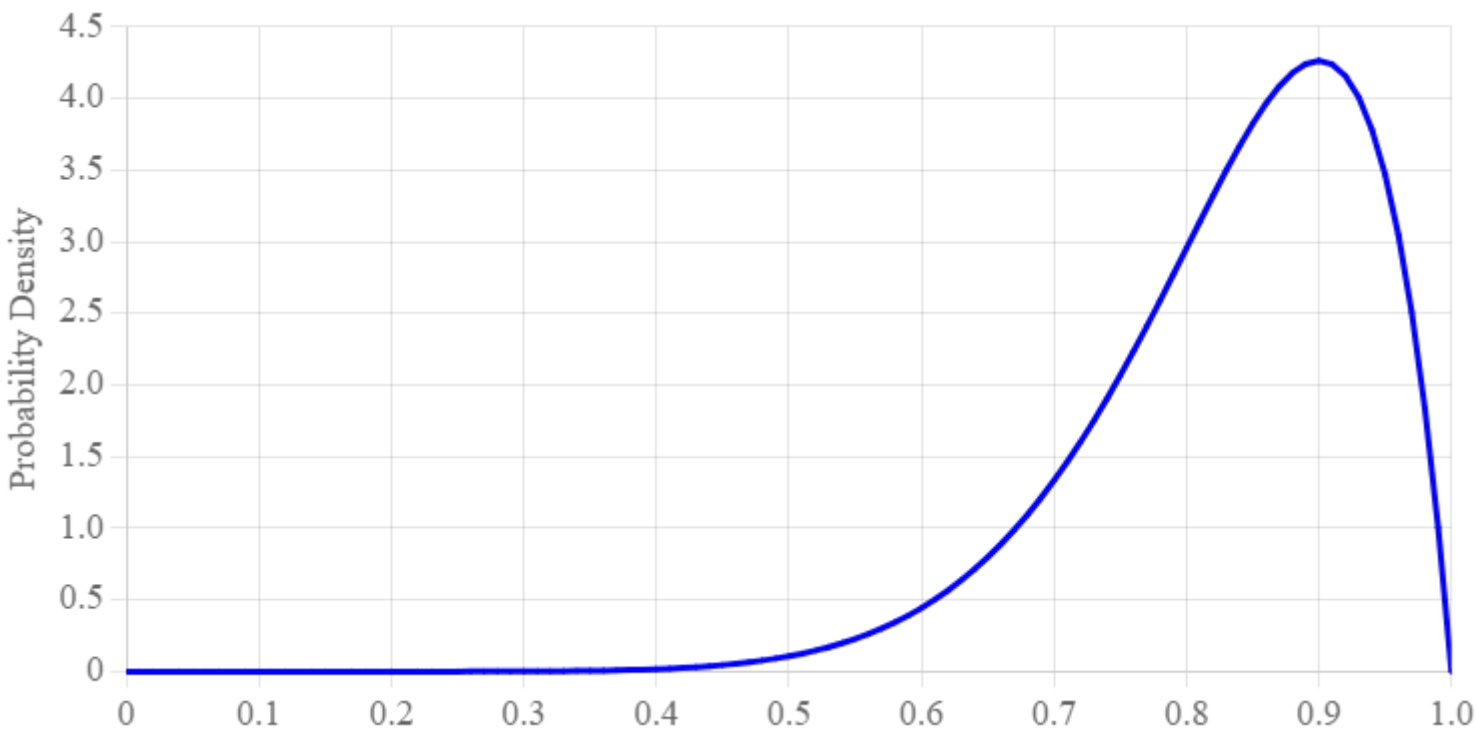
That probability is hard to think about! However it is much easier to reason about the probability with the condition reveresed: $P(H = 9, T = 1 | X = x)$. This term asks the question: what is the probability of seeing 9 heads and 1 tail in 10 coin flips, given that the true probability of a heads is x. Convince yourself that this probability is just a binomial probability mass function with $n = 10$ experiements, and $p = x$ evaluated at $k = 9$ heads:

$$P(H = 9, T = 1 | X = x) = \binom{10}{9} x^9 (1 - x)^1$$

We are presented with a perfect context for Bayes' theorem with random variables. We know a conditional probability in one direction and we would like to know it in the other:

$$\begin{aligned}
& f(X = x | H = 9, T = 1) \\
&= \frac{P(H = 9, T = 1 | X = x) \cdot f(X = x)}{P(H = 9, T = 1)} && \text{Bayes Theorem} \\
&= \frac{\binom{10}{9} x^9 (1 - x)^1 \cdot f(X = x)}{P(H = 9, T = 1)} && \text{Binomial PMF} \\
&= \frac{\binom{10}{9} x^9 (1 - x)^1 \cdot 1}{P(H = 9, T = 1)} && \text{Uniform PDF} \\
&= \frac{\binom{10}{9}}{P(H = 9, T = 1)} x^9 (1 - x)^1 && \text{Constants to front} \\
&= K \cdot x^9 (1 - x)^1 && \text{Rename constant}
\end{aligned}$$

Lets take a look at that function. For now we can let $K = \frac{1}{110}$. Regardless of K we will get the same shape, just scaled:



What a beautiful image. It tells us relatively likelihood over the probability that is governing our coinflips. Here are a few observations from this chart:

1. Even after only 10 coin flips we are very confident that the true probability is > 0.5
2. It is almost 10 times more likely that $X = 0.9$ as it is that $X = 0.6$.
3. $f(X = 1) = 0$, which makes sense. How could we have flipped that one tail if the probability of heads was 1?

Wait but why?
In the derivation above for $f(X = x | H = 9, T = 1)$ we made the claim that $P(H = 9, T = 1)$ is a constant. A lot of folks find that hard to believe. Why is that the case?

It may be helpful to juxtapose $P(H = 9, T = 1)$ with $P(H = 9, T = 1 | X = x)$. The later says "what is the probability of 9 heads, given the true probability is x". The former says "what is the probability of 9 heads, under all possible assignments of x". If you wanted to calculate $P(H = 9, T = 1)$ you could use

the law of total probability:

$$\begin{aligned} &P(H=9,T=1) \\ &\quad = \int_{y=0}^{1} P(H=9,T=1|X=y)f(X=y) \end{aligned}$$

That is a hard number to calculate, but it is in fact a constant with respect to x.

Beta Derivation

Let's generalize the derivation from the previous section, using h for the number of observed heads and t the number of observed tails.

If we let $H=h$ be the event that we saw h heads, and let $T=t$ be the event that we saw t tails in $h+t$ coinflips. We want to calculate the probability density function $f(X=x|H=h,T=t)$. We can use the exact same series of steps, starting with Bayes Theorem:

$$\begin{aligned} f(X=x|H=h,T=t) & \\ &= \frac{P(H=h,T=t|X=x)f(X=x)}{P(H=h,T=t)} && \text{Bayes Theorem} \\ &= \frac{\binom{h+t}{h}x^h(1-x)^t}{P(H=h,T=t)} && \text{Binomial PMF, Uniform PDF} \\ &= \frac{\binom{h+t}{h}}{P(H=h,T=t)}x^h(1-x)^t && \text{Moving terms around} \\ &= \frac{1}{c}\cdot x^h(1-x)^t && \text{where } c=\int_0^1 x^h(1-x)^t dx \end{aligned}$$

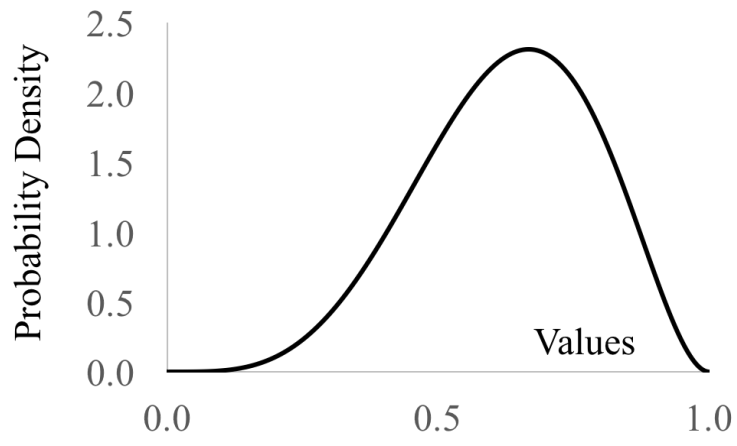
The equation that we arrived at when using a Bayesian approach to estimating our probability defines a probability density function and thus a random variable. The random variable is called a Beta distribution, and it is defined as follows:

The Probability Density Function (PDF) for $X \sim \text{Beta}(a,b)$ is:

$$f(X=x) = \begin{cases} \frac{1}{B(a,b)}x^{a-1}(1-x)^{b-1} & \text{if } 0<x<1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{where } B(a,b)=\int_0^1 x^{a-1}(1-x)^{b-1}dx$$

A Beta distribution has $E[X]=\frac{a}{a+b}$ and $Var(X)=\frac{ab}{(a+b)^2(a+b+1)}$. All modern programming languages have a package for calculating Beta CDFs. You will not be expected to compute the CDF by hand in CS109.

To model our estimate of the probability of a coin coming up heads: set $a=h+1$ and $b=t+1$. Beta is used as a random variable to represent a belief distribution of probabilities in contexts beyond estimating coin flips. For example perhaps a drug has been given to 6 patients, 4 of whom have been cured. We could express our belief in the probability that the drug can cure patients as $X \sim \text{Beta}(a=5,b=3)$:



Notice how the most likely belief for the probability of curing a patient, is $4/6$, the fraction of patients cured. This distribution shows that we hold a non-zero belief that the probability could be something other than $4/6$. It is unlikely that the probability is 0.01 or 0.09, but reasonably likely that it could be 0.5.

Beta as a Prior

You can set $X \sim \text{Beta}(a, b)$ as a prior to reflect how biased you think the coin is apriori to flipping it. This is a subjective judgment that represent $a + b - 2$ "imaginary" trials with $a - 1$ heads and $b - 1$ tails. If you then observe $h + t$ real trials with h heads you can update your belief. Your new belief would be, $X \sim \text{Beta}(a + h, b + t)$. Using the prior $\text{Beta}(1, 1) = \text{Uni}(0, 1)$ is the same as saying we haven't seen any "imaginary" trials, so apriori we know nothing about the coin. Here is the proof for the distribution of X when the prior was a Beta too:

If our prior belief is $X \sim \text{Beta}(a, b)$, then our posterior is $\text{Beta}(a + h, b + t)$:

$$
\begin{aligned}
& f(X = x | H = h, T = t) \\
&= \frac{P(H = h, T = t | X = x) f(X = x)}{P(H = h, T = t)} && \text{Bayes Theorem} \\
&= \frac{\binom{h+t}{h} x^h (1-x)^t \cdot \frac{1}{c} \cdot x^{a-1}(1-x)^{b-1}}{P(H = h, T = t)} && \text{Beta PMF, Uniform PDF} \\
&= K \cdot x^h (1-x)^t \cdot x^{a-1}(1-x)^{b-1} && \text{Combine Constants} \\
&= K \cdot x^{a+h-1}(1-x)^{b+t-1} && \text{Combine Like Bases}
\end{aligned}
$$

Which is the PDF of $\text{Beta}(a + h, b + t)$

It is pretty convenient that if we have a Beta prior belief, then our posterior belief is also Beta. This makes Betas especially convenient to work with, in code and in proof, if there are many updates that you will make to your belief over time. This property where the type of distribution is the same before and after an observation is called a conjugate prior.

Quick question: Are you allowed to just make up priors and imaginary trials? Some folks think that is fine (they are called Bayesians) and some folks think that you shouldn't make up prior beliefs (they are called frequentists). In general, for small data it can make you much better at making predictions if you are able to come up with a good prior belief.

Observation: There is a deep connection between the beta-prior and the uniform-prior (which we used initially). It turns out that $\text{Beta}(1, 1) = \text{Uni}(0, 1)$. Recall that $\text{Beta}(1, 1)$ means 0 imaginary heads and 0 imaginary tails.

Adding Random Variables

In this section on uncertainty theory we are going to explore some of the great results in probability theory. As a gentle introduction we are going to start with convolution. Convolution is a very fancy way of saying "adding" two different random variables together. The name comes from the fact that adding two random varaibles requires you to "convolve" their distribution functions. It is interesting to study in detail because (1) many natural processes can be modelled as the sum of random variables, and (2) because mathemeticians have made great progress on proving convolution theorems. For some particular random variables computing convolution has closed form equations. Importantly convolution is the sum of the random variables themselves, not the addition of the probability density functions (PDF)s that correspond to the random variables.

1. Adding Two Random Variables
2. Sum of Independent Poissons
3. Sum of Independent Binomials
4. Sum of Independent Normals
5. Sum of Independent Uniforms

Adding Two Random Variables

Deriving an expression for the likelihood for the sum of two random variables requires an interesting insight. If your random variables are discrete then the probability that $X + Y = n$ is the sum of mutually exclusive cases where X takes on a values in the range $[0, n]$ and Y takes on a value that allows the two to sum to n. Here are a few examples $X = 0 \text{ and } Y = n$, $X = 1 \text{ and } Y = n - 1$ etc. In fact all of the mutually exclusive cases can be enumerated in a sum:

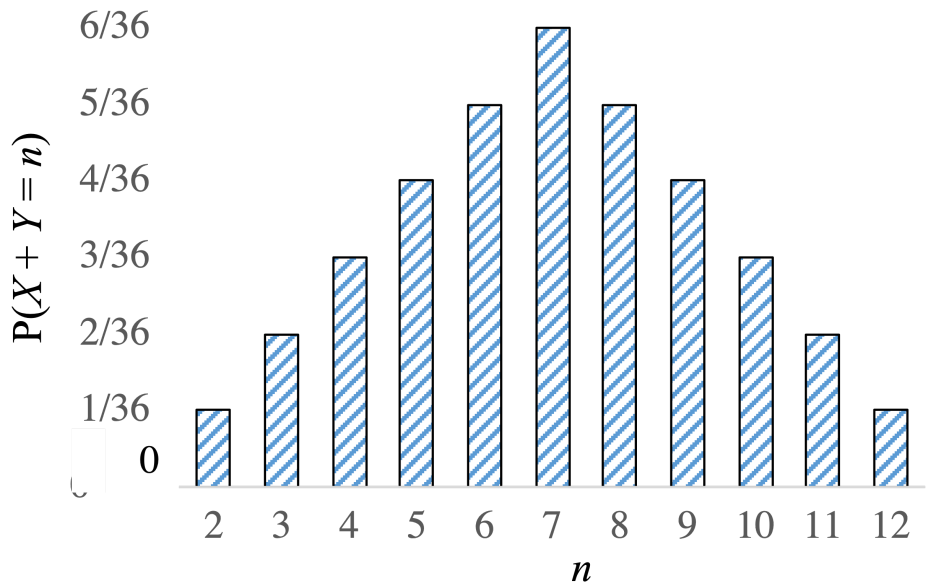
Def: General Rule for the Convolution of Discrete Variables

$$\mathrm{P}(X + Y = n) = \sum_{i=-\infty}^{\infty} \mathrm{P}(X = i, Y = n - i)$$

If the random variables are independent you can futher decompose the term $\mathrm{P}(X = i, Y = n - i)$. Let's expand on some of the mutually exclusive cases where $X + Y = n$:

i	X	Y	
0	0	n	$\mathrm{P}(X = 0, Y = n)$
1	1	$n - 1$	$\mathrm{P}(X = 1, Y = n - 1)$
2	2	$n - 2$	$\mathrm{P}(X = 2, Y = n - 2)$
	...		
n	n	0	$\mathrm{P}(X = n, Y = 0)$

Consider the sum of two independent dice. Let X and Y be the outcome of each dice. Here is the probability mass function for the sum $X + Y$:



Let's use this context to practice deriving the sum of two variables, in this case $\mathrm{P}(X+Y=n)$, starting with the General Rule for the Convolution of Discrete Random Variables. We start by considering values of n between 2 and 7. In this range $\mathrm{P}(X=i, Y=n-i) = \frac{1}{36}$ for all values of i between 1 and $n-1$. There is exactly one outcome of the two die where $X=i$ and $Y=n-i$. For values of i outside this range $n-i$ is not a valid dice outcome and $\mathrm{P}(X=i, Y=n-i) = 0$:

$$\begin{aligned}
&\mathrm{P}(X+Y=n) \\
&= \sum_{i=-\infty}^{\infty} \mathrm{P}(X=i, Y=n-i) \\
&= \sum_{i=1}^{n-1} \mathrm{P}(X=i, Y=n-i) \\
&= \sum_{i=1}^{n-1} \frac{1}{36} \\
&= \frac{n-1}{36}
\end{aligned}$$

For values of n greater than 7 we could use the same approach, though different values of i would make $\mathrm{P}(X=i, Y=n-i)$ non-zero.

This derivation for a general rule has a continuous equivalent:

$$f(X+Y=n) = \int_{i=-\infty}^{\infty} f(X=n-i, Y=i)\, di$$

Sum of Independent Poissons

For any two Poisson random variables: $X \sim \mathrm{Poi}(\lambda_1)$ and $Y \sim \mathrm{Poi}(\lambda_2)$ the sum of those two random variables is another Poisson: $X+Y \sim \mathrm{Poi}(\lambda_1+\lambda_2)$. This holds even when λ_1 is not the same as λ_2.

How could we prove a the above claim?

Example derivation:

Let's go about proving that the sum of two independent Poisson random variables is also Poisson. Let $X \sim \mathrm{Poi}(\lambda_1)$ and $Y \sim \mathrm{Poi}(\lambda_2)$ be two independent random variables, and $Z = X+Y$. What is $P(Z=n)$?

$$
\begin{aligned}
P(Z = n) &= P(X + Y = n) \\
&= \sum_{k=-\infty}^{\infty} \mathrm{P}(X = k, Y = n - k) && \text{(Convolution)} \\
&= \sum_{k=-\infty}^{\infty} P(X = k)P(Y = n - k) && \text{(Independence)} \\
&= \sum_{k=0}^{n} P(X = k)P(Y = n - k) && \text{(Range of } X \text{ and } Y\text{)} \\
&= \sum_{k=0}^{n} e^{-\lambda_1}\frac{\lambda_1^k}{k!}e^{-\lambda_2}\frac{\lambda_2^{n-k}}{(n-k)!} && \text{(Poisson PMF)} \\
&= e^{-(\lambda_1+\lambda_2)}\sum_{k=0}^{n}\frac{\lambda_1^k\lambda_2^{n-k}}{k!(n-k)!} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{n!}\sum_{k=0}^{n}\frac{n!}{k!(n-k)!}\lambda_1^k\lambda_2^{n-k} \\
&= \frac{e^{-(\lambda_1+\lambda_2)}}{n!}(\lambda_1+\lambda_2)^n && \text{(Binomial theorem)}
\end{aligned}
$$

Note that the Binomial Theorem (which we did not cover in this class, but is often used in contexts like expanding polynomials) says that for two numbers a and b and positive integer n, $(a+b)^n = \sum_{k=0}^{n}\binom{n}{k}a^k b^{n-k}$.

Sum of Independent Binomials with equal p

For any two independent Binomial random variables with the same "success" probability p:
$X \sim \mathrm{Bin}(n_1, p)$ and $Y \sim \mathrm{Bin}(n_2, p)$ the sum of those two random variables is another binomial:
$X + Y \sim \mathrm{Bin}(n_1 + n_2, p)$.

This result hopefully makes sense. The convolution is the number of sucesses across X and Y. Since each trial has the same probability of success, and there are now $n_1 + n_2$ trials, which are all independent, the convolution is simply a new Binomial. This rule does not hold when the two Binomial random variables have different parameters p.

Sum of Independent Normals

For any two independent normal random variables $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ the sum of those two random variables is another normal: $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.

Again this only holds when the two normals are independent.

Sum of Independent Uniforms

If X and Y are independent uniform random variables where $X \sim \mathrm{Uni}(0, 1)$ and $Y \sim \mathrm{Uni}(0, 1)$:

$$
f(X + Y = n) = \begin{cases} n & \text{if } 0 < n \leq 1 \\ 2 - n & \text{if } 1 < n \leq 2 \\ 0 & \text{else} \end{cases}
$$

Example derivation:
Calculate the PDF of $X + Y$ for independent uniform random variables $X \sim \mathrm{Uni}(0, 1)$ and $Y \sim \mathrm{Uni}(0, 1)$? First plug in the equation for general convolution of independent random variables:

$$
\begin{aligned}
f(X+Y=n) &= \int_{i=0}^{1} f(X=n-i, Y=i)di \\
&= \int_{i=0}^{1} f(X=n-i)f(Y=i)di && \text{Independence} \\
&= \int_{i=0}^{1} f(X=n-i)di && \text{Because } f(Y=y)=1
\end{aligned}
$$

It turns out that is not the easiest thing to integrate. By trying a few different values of n in the range $[0, 2]$ we can observe that the PDF we are trying to calculate is discontinuous at the point $n = 1$ and thus will be easier to think about as two cases: $n < 1$ and $n > 1$. If we calculate $f(X+Y=n)$ for both cases and correctly constrain the bounds of the integral we get simple closed forms for each case:

$$
f(X+Y=n) = \begin{cases} n & \text{if } 0 < n \leq 1 \\ 2-n & \text{if } 1 < n \leq 2 \\ 0 & \text{else} \end{cases}
$$

Central Limit Theorem

There are two ways that you could state the central limit theorem. Either that the sum of IID random variables is normally distributed, or that the average of IID random variables is normally distributed.

The Central Limit Thorem (Sum Version)
Let be independent and identically distributed random variables. The **sum** of these random variables approaches a normal as :

Where and . Note that since each is identically distributed they share the same expectation and variance.

At this point you probably think that the central limit theorem is awesome. But it gets even better. With some algebraic manipulation we can show that if the sample mean of IID random variables is normal, it follows that the sum of equally weighted IID random variables must also be normal:

The Central Limit Thorem (Average Version)
Let be independent and identically distributed random variables. The **average** of these random variables approaches a normal as :

Where and .

Central Limit Theorem Intuition

In the previous section we explored what happens when you add two random variables. What happens when you add more than two random variables? For example, what if I wanted to add up 100 different uniform random variables:

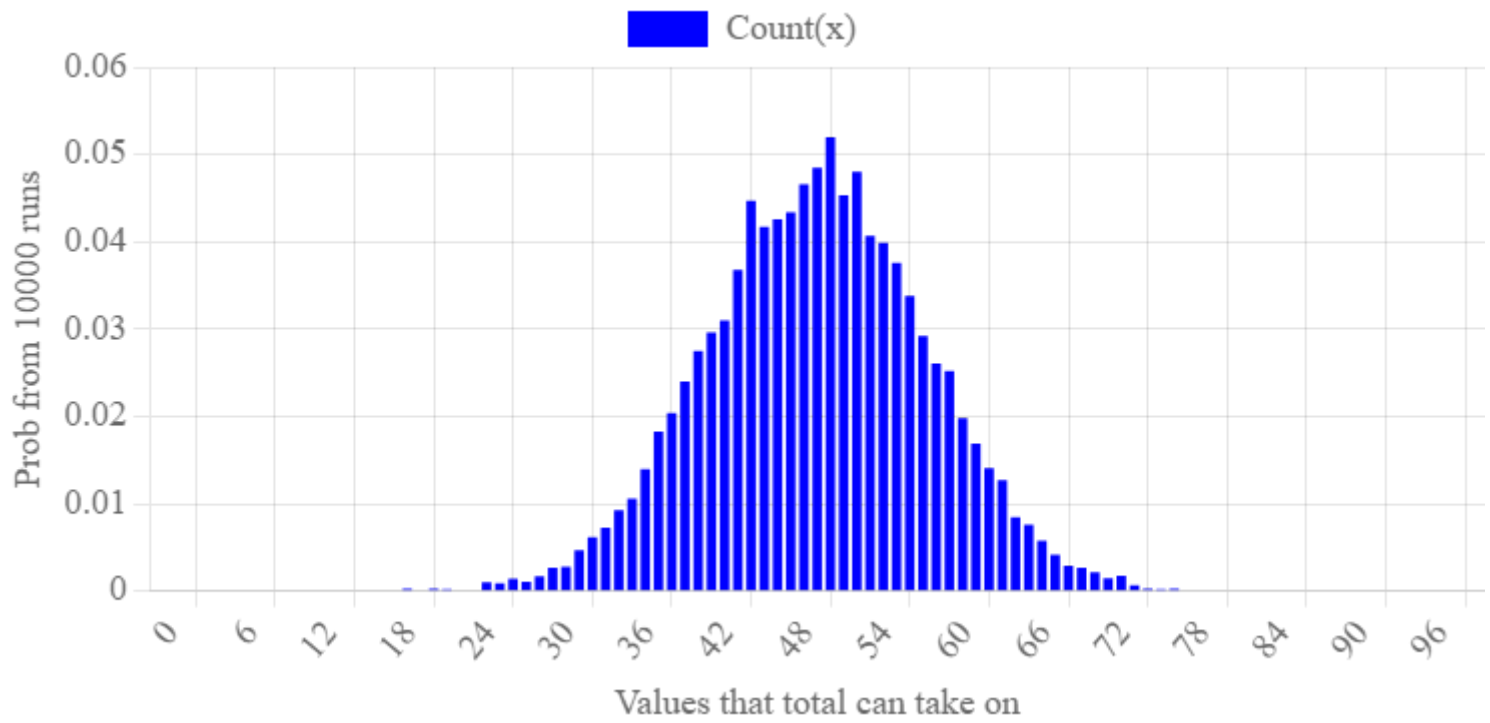
```
from random import random

def add_100_uniforms():
    total = 0
    for i in range(100):
        # returns a sample from uniform(0, 1)
        x_i = random()
        total += x_i
    return total
```

The value, `total` returned by this function will be a random variable. Hit the button below to run the function and observe the resulting value of total:

`add_100_uniforms()` `total: 45.70934`

What does total look like as a distribution? Let's calculate `total` many times and visualize the histogram of values it produces.



10,000 more runs

That is interesting! `total` which is the sum of 100 independent uniforms looks normal. Is that a special property of uniforms? No! It turns out to work for almost any type of distribution (as long as the thing you are adding has finite mean and finite variance, everything we have covered in this reader).

- Sum of 40 where ? Normal.
- Sum of 90 where ? Normal.
- Sum of 50 dice-rolls? Normal.
- Average of 10000 where ? Normal.

For any distribution the sum, or average, of independent equally-weighted samples from that distribution, will be normal.

Continuity Correction

Now we can see that the Binomial Approximation using a Normal actually derives from the central limit theorem. Recall that, when computing probabilities for a normal approximation, we had to to use a continuity correction. This was because we were approximating a discrete random variable (a binomial) with a continuous one (a normal). You should use a continuity correction any time your normal is approximating a discrete random variable. The rules for a general continuity correction are the same as the rules for the binomial-approximation continuity correction.

In the motivating example above, where we added 100 uniforms, a continuity correction isn't needed because the sum of uniforms is continuous. In the dice sum example below, a continuity correction is needed because die outcomes are discrete.

Examples

Example:

You will roll a 6 sided dice 10 times. Let be the total value of all 10 dice = . You win the game if or . Use the central limit theorem to calculate the probability that you win. Recall that and —.

Let be the approximating normal. By the Central Limit Theorem .
Substituting in the known values for expectation and variance:

Example:

Say you have a new algorithm and you want to test its running time. You have an idea of the variance of the algorithm's run time: but you want to estimate the mean: sec. You can run the algorithm repeatedly (IID trials). How many trials do you have to run so that your estimated runtime = with 95\% certainty? Let be the run time of the -th run (for).

By the central limit theorem, the standard normal must be equal to:

Now we rewrite our probability inequality so that the central term is :

And now we can find the value of that makes this equation hold.

Thus it takes 62 runs. If you are interested in how this extends to cases where the variance is unknown, look into variations of the students' t-test.

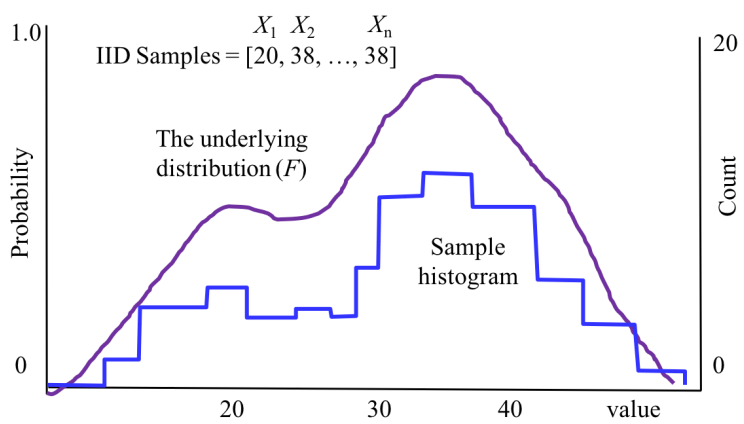
Sampling

In this section we are going to talk about statistics calculated on samples from a population. We are then going to talk about probability claims that we can make with respect to the original population -- a central requirement for most scientific disciplines.

Let's say you are the king of Bhutan and you want to know the average happiness of the people in your country. You can't ask every single person, but you could ask a random subsample. In this next section we will consider principled claims that you can make based on a subsample. Assume we randomly sample 200 Bhutanese and ask them about their happiness. Our data looks like this: $72, 85, \ldots, 71$. You can also think of it as a collection of $n = 200$ I.I.D. (independent, identically distributed) random variables $X_1, X_2, \ldots, X_n$.

Understanding Samples

The idea behind sampling is simple, but the details and the mathematical notation can be complicated. Here is a picture to show you all of the ideas involved:



The theory is that there is some large population (for example the 774,000 people who live in Bhutan). We collect a sample of n people at random, where each person in the population is equally likely to be in our sample. From each person we record one number (for example their reported happiness). We are going to call the number from the ith person we sampled X_i. One way to visualize your samples $X_1, X_2, \ldots, X_n$ is to make a histogram of their values.

We make the assumption that all of our X_is are identically distributed. That means that we are assuming there is a single underlying distribution F that we drew our samples from. Recall that a distribution for discrete random variables should define a probability mass function.

Estimating Mean and Variance from Samples

We assume that the data we look at are IID from the same underlying distribution (F) with a true mean (μ) and a true variance (σ^2). Since we can't talk to everyone in Bhutan we have to rely on our sample to estimate the mean and variance. From our sample we can calculate a sample mean ($\bar{X}$) and a sample variance (S^2). These are the best guesses that we can make about the true mean and true variance.

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n} \qquad S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$$

The first question to ask is, are those unbiased estimates? Yes. Unbiased, means that if we were to repeat this sampling process many times, the expected value of our estimates should be equal to the true values we are trying to estimate. We will prove that that is the case for $\bar{X}$. The proof for S^2 is in lecture slides.

$$
\begin{aligned}
E[\bar{X}] &= E[\sum_{i=1}^{n} \frac{X_i}{n}] = \frac{1}{n} E\left[\sum_{i=1}^{n} X_i\right] \\
&= \frac{1}{n} \sum_{i=1}^{n} E[X_i] = \frac{1}{n} \sum_{i=1}^{n} \mu = \frac{1}{n} n\mu = \mu
\end{aligned}
$$

The equation for sample mean seems related to our understanding of expectation. The same could be said about sample variance except for the surprising $(n-1)$ in the denominator of the equation. Why $(n-1)$? That denominator is necessary to make sure that the $E[S^2] = \sigma^2$.

The intuition behind the proof is that sample variance calculates the distance of each sample to the sample mean, \emph{not} the true mean. The sample mean itself varies, and we can show that its variance is also related to the true variance.

Standard Error

Ok, you convinced me that our estimates for mean and variance are not biased. But now I want to know how much my sample mean might vary relative to the true mean.

$$
\begin{aligned}
\text{Var}(\bar{X}) &= \text{Var}(\sum_{i=1}^{n} \frac{X_i}{n}) = \left(\frac{1}{n}\right)^2 \text{Var}\left(\sum_{i=1}^{n} X_i\right) \\
&= \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \text{Var}(X_i) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \sigma^2 = \left(\frac{1}{n}\right)^2 n\sigma^2 = \frac{\sigma^2}{n} \\
&\approx \frac{S^2}{n} \\
\text{Std}(\bar{X}) &\approx \sqrt{\frac{S^2}{n}}
\end{aligned}
$$

That term, Std($\bar{X}$), has a special name. It is called the standard error and its how you report uncertainty of estimates of means in scientific papers (and how you get error bars). Great! Now we can compute all these wonderful statistics for the Bhutanese people. But wait! You never told me how to calculate the Std(S^2). That is hard because the central limit theorem doesn't apply to the computation of S^2. Instead we will need a more general technique. See the next chapter: Bootstrapping

Let's say we calculate the our sample of happiness has n = 200 people. The sample mean is $\bar{X} = 83$ (what is the unit here? happiness score?) and the sample variance is $S^2 = 450$. We can now calculate the standard error of our estimate of the mean to be 1.5. When we report our results we will say that our estimate of the average happiness score in Bhutan is 83 $\pm$ 1.5. Our estimate of the variance of happiness is 450 $\pm$?.

Bootstrapping

The bootstrap is a newly invented statistical technique for both understanding distributions of statistics and for calculating p-values (a p-value is a the probability that a scientific claim is incorrect). It was invented here at Stanford in 1979 when mathematicians were just starting to understand how computers, and computer simulations, could be used to better understand probabilities.

The first key insight is that: if we had access to the underlying distribution (F) then answering almost any question we might have as to how accurate our statistics are becomes straightforward. For example, in the previous section we gave a formula for how you could calculate the sample variance from a sample of size n. We know that in expectation our sample variance is equal to the true variance. But what if we want to know the probability that the true variance is within a certain range of the number we calculated? That question might sound dry, but it is critical to evaluating scientific claims! If you knew the underlying distribution, F, you could simply repeat the experiment of drawing a sample of size n from F, calculate the sample variance from our new sample and test what portion fell within a certain range.

The next insight behind bootstrapping is that the best estimate that we can get for F is from our sample itself! The simplest way to estimate F (and the one we will use in this class) is to assume that the $P(X = k)$ is simply the fraction of times that k showed up in the sample. Note that this defines the probability mass function of our estimate $\hat{F}$ of F.

```
def bootstrap(sample):
   N = number of elements in sample
   pmf = estimate the underlying pmf from the sample
   stats = []
   repeat 10,000 times:
      resample = draw N new samples from the pmf
      stat = calculate your stat on the resample
      stats.append(stat)
   stats can now be used to estimate the distribution of the stat
```

Bootstrapping is a reasonable thing to do because the sample you have is the best and only information you have about what the underlying population distribution actually looks like. Moreover most samples will, if they're randomly chosen, look quite like the population they came from.

To calculate $\mathrm{Var}(S^2)$ we could calculate S_i^2 for each resample i and after 10,000 iterations, we could calculate the sample variance of all the S_i^2s. You might be wondering why the resample is the same size as the original sample (n). The answer is that the variation of the variation of stat that you are calculating could depend on the size of the sample (or the resample). To accurately estimate the distribution of the stat we must use resamples of the same size.

The bootstrap has strong theoretic grantees, and is accepted by the scientific community. It breaks down when the underlying distribution has a ``long tail'' or if the samples are not I.I.D.

Example of p-value calculation

We are trying to figure out if people are happier in Bhutan or in Nepal. We sample $n_1 = 200$ individuals in Bhutan and $n_2 = 300$ individuals in Nepal and ask them to rate their happiness on a scale from 1 to 10. We measure the sample means for the two samples and observe that people in Nepal are slightly happier--the difference between the Nepal sample mean and the Bhutan sample mean is 0.5 points on the happiness scale.

If you want to make this claim scientific you should calculate a p-value. A p-value is the probability that, when the null hypothesis is true, the statistic measured would be equal to, or more extreme than, than the value you are reporting. The null hypothesis is the hypothesis that there is no relationship between two

measured phenomena or no difference between two groups.

In the case of comparing Nepal to Bhutan, the null hypothesis is that there is no difference between the distribution of happiness in Bhutan and Nepal. The null hypothesis argument is: there is no difference in the distribution of happiness between Nepal and Bhutan. When you drew samples, Nepal had a mean that 0.5 points larger than Bhutan by chance.

We can use bootstrapping to calculate the p-value. First, we estimate the underlying distribution of the null hypothesis underlying distribution, by making a probability mass function from all of our samples from Nepal and all of our samples from Bhutan.

```
def pvalue_bootstrap(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   universal_sample = combine bhutan_samples and nepal_samples
   universal_pmf = estimate the underlying pmf of the universalSample
   count = 0
   observed_difference = mean(nepal_sample) - mean(bhutan_sample)
   repeat 10,000 times:
      bhutan_resample = draw N new samples from the universalPmf
      nepal_resample = draw M new samples from the universalPmf
      mu_bhutan = sample mean of the bhutanResample
      mu_nepal = sample mean of the nepalResample
      mean_difference = |muNepal - muBhutan|
      if mean_difference > observed_difference:
         count += 1
   pvalue = count / 10,000
```

This is particularly nice because nowhere did we have to make an assumption about a parametric distribution that our samples came from (ie we never had to claim that happiness is gaussian). You might have heard of a t-test. That is another way of calculating p-values, but it makes the assumption that both samples are gaussian and that they both have the same variance. In the modern context where we have reasonable computer power, bootstrapping is a more correct and versatile tool.

Algorithmic Analysis

In this section we are going to use probability to analyze code. Specifically we are going to be calculating expectations on code: expected run time, expected resulting values etc. The reason that we are going to focus on expectation is that it has several nice properties. One of the most useful properties that we have seen so far is that the expectation of a sum, is the sum of expectations, *regardless* of whether the random variables are independent of one another. In this section we will see a few more helpful properties, including the Law of Total Expectation, which is also helpful in analyzing code:

Law of Total Expectation
The law of total expectation gives you a way to calculate $\mathrm{E}[X]$ in the scenareo where it is easier to compute $\mathrm{E}[X|Y=y]$ where Y is some other random variable:

$$\begin{aligned}\mathrm{E}[X] &= \mathrm{E}\left[E[X|Y]\right] \\ &= \sum_{y} \mathrm{E}[X|Y=y]\,\mathrm{P}(Y=y)\end{aligned}$$

Distributed File System

Imagine the task of loading a large file from your computer at Stanford, over the interet. Your file is stored in a distributed file system. In a distributed file system, the closest instance of your file might be on one of several computers at different locations in the world. Imagine you know the probability that the file is in one of a few locations l: $\mathrm{P}(L=l)$, and for each location the expected time, T, to get the file, $\mathrm{E}[T|L=l]$, given it is in that location:

Location	$P(L=l)$	$\mathrm{E}[T\|L=l]$
SoCal	0.5	0.3 seconds
New York	0.2	20.7 seconds
Japan	0.3	96.3 seconds

The Law of Total Expectation gives a straightforward way to compute $\mathrm{E}[T]$:

$$\begin{aligned}\mathrm{E}[T] &= \sum_{l} \mathrm{E}[T|L=l]\,\mathrm{P}(L=l) \\ &= 0.5 \cdot 0.3 + 0.2 \cdot 20.7 + 0.3 \cdot 96.7 \\ &= 33.3 \text{ seconds}\end{aligned}$$

Toy Example of Recursive Code

In theoretical computer science, there are many times where you want to analyze the expected runtime of an algorithm. To practice this technique let's try to solve a simple recursive function. Let Y be the value returned by `recurse()`. What is $E[Y]$?

```python
def recurse():
    x = random.choice([1, 2, 3])  # Equally likely values
    if x == 1:
        return 3;
    else if (x == 2):
        return 5 + recurse()
    else:
        return 7 + recurse()
```

In order to solve this problem we are going to need to use the law of total expectation considering X as your background variable.

$$\begin{aligned}\mathrm{E}[Y] = &\sum_{i \in \{1,2,3\}} \mathrm{E}[Y|X=i]\,\mathrm{P}(X=i) \\ =&\mathrm{E}[Y|X=1]\,\mathrm{P}(X=1) \\ &+\mathrm{E}[Y|X=2]\,\mathrm{P}(X=2) \\ &+\mathrm{E}[Y|X=3]\,\mathrm{P}(X=3)\end{aligned}$$

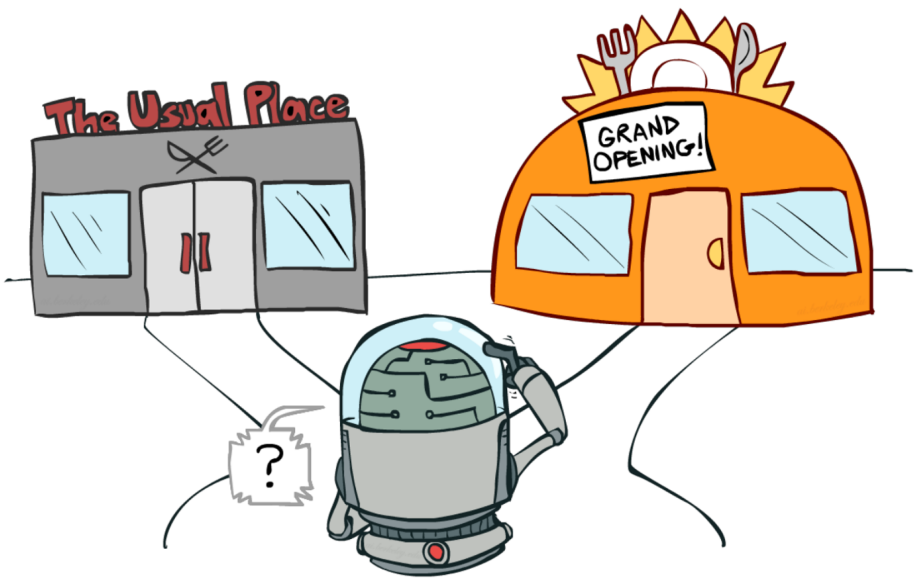
We know the probability $\mathrm{P}(X=x) = \frac{1}{3}$ for $x \in \{1,2,3\}$. How can we compute a value such as $\mathrm{E}[Y|X=2]$? Well that is the expectation of your return, in the world where $X=2$. In that case you will return 5 + recurse(). The expectation of that is $5 + \mathrm{E}[Y]$. Plugging a similar result for each case we can continue our solution:

$$\begin{aligned}E[Y|X=1] &= 3 \\ E[Y|X=2] &= 5 + \mathrm{E}[Y] \\ E[Y|X=3] &= 7 + \mathrm{E}[Y]\end{aligned}$$

Now we can just plug values into the law of total expectation:

$$\begin{aligned}\mathrm{E}[Y] = &\sum_{i \in \{1,2,3\}} \mathrm{E}[Y|X=i]\,\mathrm{P}(X=i) \\ =&\mathrm{E}[Y|X=1]\,\mathrm{P}(X=1) \\ &+\mathrm{E}[Y|X=2]\,\mathrm{P}(X=2) \\ &+\mathrm{E}[Y|X=3]\,\mathrm{P}(X=3) \\ =&3 \cdot 1/3 \\ &+(5+\mathrm{E}[Y]) \cdot 1/3 \\ &+(7+\mathrm{E}[Y]) \cdot 1/3 \\ =&(15+2\,\mathrm{E}[Y]) \cdot 1/3 \\ 1/3 \cdot \mathrm{E}[Y] =&5 \\ \mathrm{E}[Y] =&15\end{aligned}$$

Thompson Sampling



Imagine having to make the following series of decisions. You have two drugs you can administer, drug 1, or drug 2. Initially you have no idea which drug is better. You want to know which drug is the most effective, but at the same time, there are costs to *exploration* — the stakes are high.

Here is an example:

```
Welcome to the drug simulator.
There are two drugs: 1 and 2.

Next patient. Which drug? (1 or 2): 1
Failure. Yikes!

Next patient. Which drug? (1 or 2): 2
Success. Patient lives!

Next patient. Which drug? (1 or 2): 2
Failure. Yikes!

Next patient. Which drug? (1 or 2): 1
Failure. Yikes!

Next patient. Which drug? (1 or 2): 1
Success. Patient lives!

Next patient. Which drug? (1 or 2): 1
Failure. Yikes!

Next patient. Which drug? (1 or 2): 2
Success. Patient lives!

Next patient. Which drug? (1 or 2): 2
Failure. Yikes!

Next patient. Which drug? (1 or 2):
```

This problem is suprisingly complex. It sometimes goes by the name "the multi-armed bandit problem!" In fact, the perfect answer to this question can be exponentially hard to calculate. There are many approximate solutions and it is an active area of research.

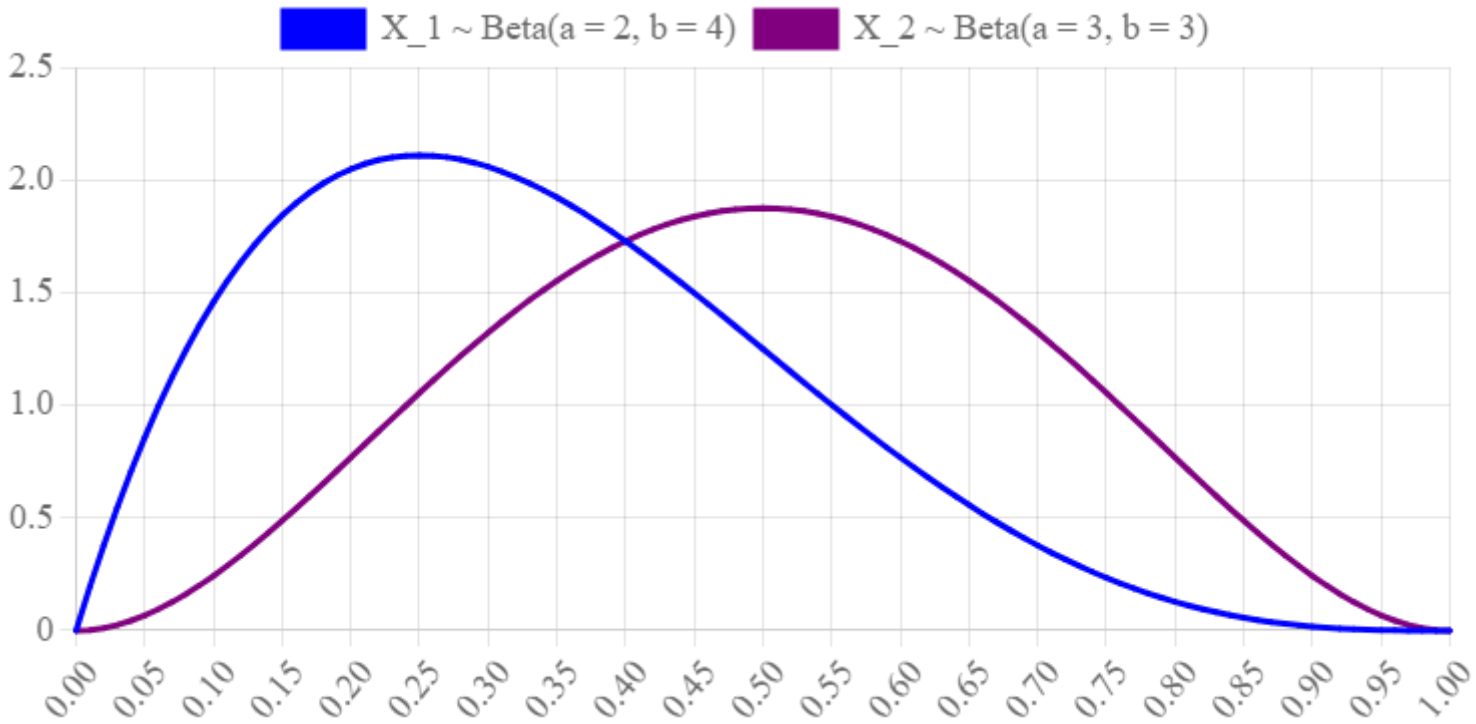
One solution has risen to be a rather popular option: Thompson Sampling. It is easy to implement, elegant to understand, has provable garuntees [1], and in practice does very well [2].

What You Know About The Choices

The first step in Thompson sampling is to express what you know (and what you do not know) about your choices. Let us revisit the example of the two drugs in the previous section. By the end we had tested drug 1 four times (with 1 success) and we had tested drug 2 four times (with 2 successes). A sophisticated way to represent our belief in the two hidden probabilites behind drug 1 and drug 2 is to use the Beta distribution. Let X_1 be the belief in the probability for drug 1 and let X_2 be the belief in the probability for drug 2.

$$X_1 \sim \text{Beta}(a = 2, b = 4)$$
$$X_2 \sim \text{Beta}(a = 3, b = 3)$$

Recall that in the Beta distribution with a uniform prior the first parameter, a, is number of observed successes + 1. The second parameter, b, is the number of observed fails + 1. It is helpful to look at these two distributions graphically:



If we had to guess, drug 2 is looking better, but there is still a lot of uncertainty, represented by the high variance in these beliefs. That is a helpful representation. But how can we use this information to make a good decision of the next drug.

Making a Choice

It is hard to know what is the right choice! If you only had one more patient, then it is clear what you should do. You should calculate the probability that $X_2 > X_1$ and if that probability is over 0.5 then you should chose a. However, if you need to continually administer the pills then it is less clear what is the right choice. If you chose 1, you miss out on the chance to learn more about 2. What should we do? We need to balance this need for "exploring" and the need to take advantage of what we already know.

The simple idea behind Thompson Sampling is to randomly make your choice according to its probability of being optimal. In this case we should chose drug 1 with the probability that 1 is > 2. How do people do this in practice? They have a very simple formula. Take a random sample from each Beta distribution. Chose the option which has a larger value for its sample.

```
sample_a = sample_beta(2, 4)
sample_b = sample_beta(3, 3)
if sample_a > sample_b:
    choose choice a
else:
    choose choice b
```

What does it mean to take a sample? It means to chose a value according to the probability density (or probability mass) function. So in our example above, we might sample 0.4 for drug 1, and sample 0.35 for drug 2. In which case we would go with drug 1.

At the start Thompson Sampling "explores" quite a lot of time. As it gets more confident that one drug is better than another, it will start to chose that drug most of the time. Eventually it will converge to knowing which drug is best, and it will always chose that drug.

Night Sight

In this problem we explore how to use probability theory to take photos in the dark. Digital cameras have a sensor that capture photons over the duration of a photo shot to produce pictures. However, these sensors are subject to “shot noise” which are random fluctuations in the amount of photons that hit the lens. In the scope of this problem, we only consider a single pixel. The arrival of shot noise photons on a surface is independent with constant rate.

Left: photo captured using a standard photo. Right: the same photo using a shot burst [1].

For shot noise, standard deviation is what matters! Why? Because if the camera can compute the expected amount of noise, it can simply subtract it out. But the fluctuations around the mean (measured as standard deviation) lead to changes in measurement that the camera can't simply subtract out.

Part 1: A Standard Photo

First lets calculate the amount of noise if we take a photo the standard way. If the time duration of a photo shot is 1000 μs, what is the standard deviation of the amount of photons captured by the pixel during a single photo? Note that shot noise photons land on a particular pixel at a rate of 10 photons per microsecond (μs).

Noise in a standard photo: As you may have guessed, because photos hit the camera at a constant rate, and independent of one another, the number of shot noise photos hitting any pixel is modelled as a Poisson! For the given rate of noise, let X be the amount of shot noise photos that hit the pixel:

$$X \sim \text{Poi}(\lambda = 10,000).$$

Note that 10,000 is the average number of photons that hit in 1000μs (duration in microseconds multiplied by photons per microsecond). The standard of a Poisson is simply equal to the square root of its parameter, $\sqrt{\lambda}$. Thus the standard deviation of the shot noise photons captured is 100 (quite high).

Part 2: A Shutter Shot

To mitigate shot noise, Stanford graduates realized that you can take a shutter shot (many camera shots in quick succession) and sum the number of photons captured. Because of limitations in cell phone cameras, the largest number of photos a camera can take in 1000μs is 15 photos, each with a duration of 66μs. What is the standard deviation of shot noise if we average the photons across a shutter shot of 15 photos?

Noise with a shutter shot:
Let Y be the average quantity of shot noise photons across the 15 photos, captured by the single pixel. We want to calculate the $\text{Var}(Y)$. Specifically, $Y = \frac{1}{15}\sum_{i=1}^{15} X_i$ where X_i is the amount of shot noise photons in the ith photo. Similar to the previous part:

$$X_i \sim \text{Poi}(\lambda = 66 \cdot 10)$$

and since X_i is a Poisson, $\mathrm{E}[X_i] = 660$ and $\mathrm{Var}(X_i) = 660$.

Since Y is the average of IID random variables, the Central Limit Theorem will kick in. Moreover, by the CLT rule Y will have variance equal to $1/n \cdot \mathrm{Var}(X_i)$.

$$\begin{aligned}\mathrm{Var}(Y) &= 1/n \cdot \mathrm{Var}(X_i) \\ &= 1/15 \cdot 660 = 44\end{aligned}$$

The standard deviation will then be the square root of this variance $\mathrm{Std}(Y) = \sqrt{44}$ which is approximately 6.6. That is a huge reduction in shot noise!

Problem by Will Song and Chris Piech. Night Sight by Google.

P-Hacking

It turns out that science has a bug! If you test many hypotheses but only report the one with the lowest p-value you are more likely to get a spurious result (one resulting from chance, not a real pattern).

Recall p-values: A p-value was meant to represent the probability of a spurious result. It is the chance of seeing a difference in means (or in whichever statistic you are measuring) at least as large as the one observed in the dataset if the two populations were actually identical. A p-value < 0.05 is considered "statistically significant". In class we compared sample means of two populations and calculated p-values. What if we had 5 populations and searched for pairs with a significant p-value? This is called p-hacking!

To explore this idea, we are going to look for patterns in a dataset which is totally random – every value is Uniform(0,1) and independent of every other value. There is clearly no significance in any difference in means in this toy dataset. However, we might find a result which looks statistically significant just by chance. Here is an example of a simulated dataset with 5 random populations, each of which has 20 samples:

	Pop 1	Pop 2	Pop 3	Pop 4	Pop 5
1	0.330	0.272	0.959	0.985	0.175
2	0.386	0.353	0.929	0.575	0.386
3	0.232	0.839	0.009	0.229	0.899
	0.836	0.003			
	...				
		0.833	0.333	0.133	
	0.649	0.723	0.565	0.061	0.479
20	0.726	0.158	0.678	0.498	0.645
Sample mean	0.534	0.579	0.474	0.437	0.545

The numbers in the table above are just for demonstration purposes. You should not base your answer off of them. We call each population a random population to emphasize that there is no pattern.

There are Many comparisons

How many ways can you choose a pair of two populations from a set of five to compare? The values of elements within the population do not matter nor does the order of the pair.

$$\binom{5}{2}$$

Understanding the mean of IID Uniforms

What is the variance of a Uniform(0, 1)?

Let $Z \sim \text{Uni}(0,1)$

$$\begin{aligned}\text{Var}(Z) &= \frac{1}{12}(\beta - \alpha) \\ &= \frac{1}{12}(1-0) \\ &= \frac{1}{12}\end{aligned}$$

What is an approximation for the distribution of the mean of 20 samples from Uniform(0,1)?

Let $Z_1 \ldots Z_n$ be i.i.d. $\text{Uni}(0,1)$. Let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} Z_i$.

$$\text{E}[X] = \frac{1}{n}\sum_{i=1}^{n} E[Z_i] = \frac{1}{n}\sum_{i=1}^{n} 0.5 = \frac{n}{n}0.5 = 0.5$$

$$\begin{aligned}\text{Var}(X) &= \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} Z_i\right) \\ &= \frac{1}{n^2}\text{Var}\left(\sum_{i=1}^{n} Z_i\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\text{Var}\left(Z_i\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} v \\ &= \frac{n}{n^2}v = \frac{v}{n} = \frac{v}{20} = \frac{1}{240}\end{aligned}$$

Using CLT, $\bar{X} \sim N\left(\mu = 0.5, \sigma^2 = \frac{1}{240}\right)$

What is an approximation for the distribution of the mean from one population minus the mean from another population? Note: this value may be negative if the first population has a smaller mean than the second.

Let X_1 and X_2 be the means of the populations.
$X_1 \sim N(\mu = 0.5, \sigma^2 = \frac{1}{240})$
$X_2 \sim N(\mu = 0.5, \sigma^2 = \frac{1}{240})$ The expectation is simple to calculate because

$$E[X_1 - X_2] = E[X_1] - E[X_2] = 0$$

$$\begin{aligned}\text{Var}(X_1 - X_2) &= \text{Var}(X_1) + \text{Var}(X_2) \\ &= \frac{1}{120}\end{aligned}$$

The sum (or difference) of independent normals is still normal: \fbox{$Y \sim N(\mu = 0, \sigma^2 = \frac{v}{10})$}

(8 points) What is the smallest difference in means, k, that would look statistically significant if there were only two populations? In other words, the probability of seeing a difference in means of k or greater is < 0.05.

One tricky part of this problem is to recognize the double sidedness to distance. We would consider it a significant distance if $P(Y < -k)$ or $P(Y > k)$.

$$\begin{aligned}P(Y < -k) + P(Y > k) &= 0.05 \\ F_Y(-k) + (1 - F_Y(k)) &= 0.05 \\ (1 - F_Y(k)) + (1 - F_Y(k)) &= 0.05 \\ 2 - 2F_Y(k) &= 0.05 \\ F_Y(k) &= 0.975\end{aligned}$$

Now we need the inverse Φ to get the value of k out.

$$
\begin{aligned}
0.975 &= \Phi\Big(\frac{k-0}{\sqrt{v/10}}\Big) \\
\Phi^{-1}(0.975) &= \frac{k}{\sqrt{v/10}} \\
k &= \Phi^{-1}(0.975)\sqrt{v/10}
\end{aligned}
$$

(5 points) Give an expression for the probability that the smallest sample mean among 5 random populations is less than 0.2.

Let X_i be the sample mean of population i.

$$
\begin{aligned}
P(min\{X_1 \dots X_n\} < 0.2) &= P\left(\bigcup_{i=1}^{5} X_i < 0.2\right) \\
&= 1 - P\left(\left(\bigcup_{i=1}^{5} X_i < 0.2\right)^{\complement}\right) \\
&= 1 - P\left(\bigcap_{i=1}^{5} X_i \geq 0.2\right) \\
&= 1 - \prod_{i=1}^{5} P(X_i \geq 0.2) \\
&= 1 - \prod_{i=1}^{5} 1 - \Phi\Big(\frac{0.2-0.5}{\sqrt{v/20}}\Big)
\end{aligned}
$$

(7 points) Use the following functions to write code that estimates the probability that among 5 populations you find a difference of means which would be considered significant (using the bootstrapping method designed to compare 2 populations). Run at least 10,000 simulations to estimate your answer. You may use the following helper functions.

```
# the smallest difference in means that would look statistically significant
k = calculate_k()

# create a matrix with n_rows by n_cols elements, each of which is Uni(0, 1)
matrix = random_matrix(n_rows, n_cols)

# from the matrix, return the column (as a list) which has the smallest mean
min_mean_col = get_min_mean_col(matrix)

# from the matrix, return the row (as a list) which has the largest mean
max_mean_col = get_max_mean_col(matrix)

# calculate the p-value between two lists using bootstrapping (like in pset5)
p_value = bootstrap(list1, list2)
```

Write pseudocode:

```
n_significant = 0
k = calculate_k()
for i in range(N_TRIALS):
    dataset = random_matrix(20, 5)
    col_max = get_max_mean_col(dataset)
    col_min = get_min_mean_col(dataset)}
    diff = np.mean(col_max) - np.mean(col_min)}
    if diff >= k:
        n_significant += 1}

print(n_significant / N_TRIALS)
```

Differential Privacy

Recently, many organizations have released machine learning models trained on massive datasets (GPT-3, YOLO, etc...). This is a great contribution to science and streamlines modern AI research. However, publicizing these models allows for the potential ``reverse engineering'' of models to uncover the training data for the model. Specifically, an attacker can download a model, look at the parameter values and then try to reconstruct the original training data. This is particularly bad for models trained on sensetive data like health information. In this section we are going to use randomness as a method to defend against algorithmic ``reverse engineering.''

Injecting Randomness

One way to combat algorithmic reverse engineering is to add some random element to an already existing dataset. Let

$$X_1, \ldots, X_i \overset{\text{i.i.d}}{\sim} \text{Bern}(p)$$

represent a set of real human data. Consider the following snippet of code:

```
def calculateXi(Xi):
    return Xi
```

Quite simply, an attacker can call the above for all 100 samples and uncover all 100 data points. Instead, we can inject an element of randomness:

```
def calculateYi(Xi):
    obfuscate = random() # Bern with parameter p=0.5
    if obfuscate:
        return indicator(random())
    else:
        return Xi
```

The attacker can in expectation call the new function 100 times and get the correct values for 50 of them (but they wont know which 50).

Recovering p

Now consider if we publish the function calculateYi, how could a researcher who is interested in the mean of the samples get useful data? They can look at:

$$Z = \sum_{n=1}^{100} Y_i.$$

Which has expectation:

$$E[Z] = E\left[\sum_{n=1}^{100} Y_i\right] = \sum_{n=1}^{100} E[Y_i] = \sum_{n=1}^{100} \left(\frac{p}{2} + \frac{1}{4}\right) = 50p + 25$$

Then to uncover an estimate, the scientist can do,

$$p \approx \frac{Z - 25}{50}$$

And proceed to conduct more research!

Parameter Estimation

We have learned many different distributions for random variables and all of those distributions had parameters: the numbers that you provide as input when you define a random variable. So far when we were working with random variables, we either were explicitly told the values of the parameters, or, we could divine the values by understanding the process that was generating the random variables.

What if we don't know the values of the parameters and we can't estimate them from our own expert knowledge? What if instead of knowing the random variables, we have a lot of examples of data generated with the same underlying distribution? In this chapter we are going to learn formal ways of estimating parameters from data.

These ideas are critical for artificial intelligence. Almost all modern machine learning algorithms work like this: (1) specify a probabilistic model that has parameters. (2) Learn the value of those parameters from data.

Parameters

Before we dive into parameter estimation, first let's revisit the concept of parameters. Given a model, the parameters are the numbers that yield the actual distribution. In the case of a Bernoulli random variable, the single parameter was the value p. In the case of a Uniform random variable, the parameters are the a and b values that define the min and max value. Here is a list of random variables and the corresponding parameters. From now on, we are going to use the notation θ to be a vector of all the parameters:

Distribution	**Parameters**
Bernoulli(p)	$\theta = p$
Poisson(λ)	$\theta = \lambda$
Uniform(a, b)	$\theta = [a, b]$
Normal(μ, σ^2)	$\theta = [\mu, \sigma^2]$

In the real world often you don't know the "true" parameters, but you get to observe data. Next up, we will explore how we can use data to estimate the model parameters.

It turns out there isn't just one way to estimate the value of parameters. There are two main schools of thought: Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP). Both of these schools of thought assume that your data are independent and identically distributed (IID) samples: $X_1, X_2, \ldots X_n$.

Maximum Likelihood Estimation

Our first algorithm for estimating parameters is called Maximum Likelihood Estimation (MLE). The central idea behind MLE is to select that parameters (θ) that make the observed data the most likely.

The data that we are going to use to estimate the parameters are going to be n independent and identically distributed (IID) samples: $X_1, X_2, \ldots X_n$.

Likelihood

We made the assumption that our data are identically distributed. This means that they must have either the same probability mass function (if the data are discrete) or the same probability density function (if the data are continuous). To simplify our conversation about parameter estimation we are going to use the notation $f(X|\theta)$ to refer to this shared PMF or PDF. Our new notation is interesting in two ways. First, we have now included a conditional on θ which is our way of indicating that the likelihood of different values of X depends on the values of our parameters. Second, we are going to use the same symbol f for both discrete and continuous distributions.

What does likelihood mean and how is "likelihood" different than "probability"? In the case of discrete distributions, likelihood is a synonym for the probability mass, or joint probability mass, of your data. In the case of continuous distribution, likelihood refers to the probability density of your data.

Since we assumed that each data point is independent, the likelihood of all of our data is the product of the likelihood of each data point. Mathematically, the likelihood of our data give parameters θ is:

$$L(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

For different values of parameters, the likelihood of our data will be different. If we have correct parameters our data will be much more probable than if we have incorrect parameters. For that reason we write likelihood as a function of our parameters (θ).

Maximization

In maximum likelihood estimation (MLE) our goal is to chose values of our parameters (θ) that maximizes the likelihood function from the previous section. We are going to use the notation $\hat{\theta}$ to represent the best choice of values for our parameters. Formally, MLE assumes that:

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\ L(\theta)$$

Argmax is short for Arguments of the Maxima. The argmax of a function is the value of the domain at which the function is maximized. It applies for domains of any dimension.

A cool property of argmax is that since log is a monotone function, the argmax of a function is the same as the argmax of the log of the function! That's nice because logs make the math simpler. If we find the argmax of the log of likelihood it will be equal to the armax of the likelihood. Thus for MLE we first write the Log Likelihood function (LL)

$$LL(\theta) = \log L(\theta) = \log \prod_{i=1}^{n} f(X_i|\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

To use a maximum likelihood estimator, first write the log likelihood of the data given your parameters. Then chose the value of parameters that maximize the log likelihood function. Argmax can be computed in many ways. All of the methods that we cover in this class require computing the first derivative of the

function.

Bernoulli MLE Estimation

For our first example, we are going to use MLE to estimate the p parameter of a Bernoulli distribution. We are going to make our estimate based on n data points which we will refer to as IID random variables $X_1, X_2, \dots X_n$. Every one of these random variables is assumed to be a sample from the same Bernoulli, with the same p, $X_i \sim \text{Ber}(p)$. We want to find out what that p is.

Step one of MLE is to write the likelihood of a Bernoulli as a function that we can maximize. Since a Bernoulli is a discrete distribution, the likelihood is the probability mass function.

The probability mass function of a Bernoulli X can be written as $f(x) = p^x(1-p)^{1-x}$. Wow! What's up with that? It's an equation that allows us to say that the probability that $X = 1$ is p and the probability that $X = 0$ is $1 - p$. Convince yourself that when $X_i = 0$ and $X_i = 1$ the PMF returns the right probabilities. We write the PMF this way because its derivable.

Now let's do some MLE estimation:

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} && \text{First write the likelihood function} \\
LL(\theta) &= \sum_{i=1}^{n} \log p^{x_i}(1-p)^{1-x_i} && \text{Then write the log likelihood function} \\
&= \sum_{i=1}^{n} x_i(\log p) + (1-x_i) log(1-p) \\
&= Y \log p + (n - Y) \log(1-p) && \text{where } Y = \sum_{i=1}^{n} x_i
\end{aligned}
$$

Great Scott! We have the log likelihood equation. Now we simply need to chose the value of p that maximizes our log-likelihood. As your calculus teacher probably taught you, one way to find the value which maximizes a function that is to find the first derivative of the function and set it equal to 0.

$$
\frac{\delta LL(p)}{\delta p} = Y\frac{1}{p} + (n - Y)\frac{-1}{1-p} = 0
$$

$$
\hat{p} = \frac{Y}{n} = \frac{\sum_{i=1}^{n} x_i}{n}
$$

All that work and find out that the MLE estimate is simply the sample mean...

Normal MLE Estimation

Practice is key. Next up we are going to try and estimate the best parameter values for a normal distribution. All we have access to are n samples from our normal which we refer to as IID random variables $X_1, X_2, \dots X_n$. We assume that for all i, $X_i \sim N(\mu = \theta_0, \sigma^2 = \theta_1)$. This example seems trickier since a normal has **two** parameters that we have to estimate. In this case θ is a vector with two values, the first is the mean (μ) parameter. The second is the variance(σ^2) parameter.

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(X_i|\theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\theta_1}} e^{-\frac{(x_i-\theta_0)^2}{2\theta_1}} && \text{Likelihood for a continuous variable is the PDF} \\
LL(\theta) &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\theta_1}} e^{-\frac{(x_i-\theta_0)^2}{2\theta_1}} && \text{We want to calculate log likelihood} \\
&= \sum_{i=1}^{n} \left[-\log(\sqrt{2\pi\theta_1}) - \frac{1}{2\theta_1}(x_i - \theta_0)^2 \right]
\end{aligned}
$$

Again, the last step of MLE is to chose values of θ that maximize the log likelihood function. In this case we can calculate the partial derivative of the LL function with respect to both θ_0 and θ_1, set both equations to equal 0 and than solve for the values of θ. Doing so results in the equations for the values

$\hat{\mu} = \hat{\theta_0}$ and $\hat{\sigma^2} = \hat{\theta_1}$ that maximize likelihood. The result is: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$.

Linear Transform Plus Noise

MLE is an algorithm that can be used for any probability model with a derivable likelihood function. As an example lets estimate the parameter θ in a model where there is a random variable Y such that $Y = \theta X + Z$, $Z \sim N(0, \sigma^2)$ and X is an unknown distribution.

In the case where you are told the value of X, θX is a number and $\theta X + Z$ is the sum of a gaussian and a number. This implies that $Y|X \sim N(\theta X, \sigma^2)$. Our goal is to chose a value of θ that maximizes the probability IID: $(X_1, Y_1), (X_2, Y_2), \ldots (X_n, Y_n)$.

We approach this problem by first finding a function for the log likelihood of the data given θ. Then we find the value of θ that maximizes the log likelihood function. To start, use the PDF of a Normal to express the probability of $Y|X, \theta$:

$$f(Y_i|X_i, \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}}$$

Now we are ready to write the likelihood function, then take its log to get the log likelihood function:

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(Y_i, X_i|\theta) && \text{Let's break up this joint} \\
&= \prod_{i=1}^{n} f(Y_i|X_i, \theta) f(X_i) && f(X_i) \text{ is independent of } \theta \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{Substitute in the definition of } f(Y_i|X_i)
\end{aligned}$$

$$\begin{aligned}
LL(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{Substitute in } L(\theta) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} + \sum_{i=1}^{n} \log f(X_i) && \text{Log of a product is the sum of logs} \\
&= n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \theta X_i)^2 + \sum_{i=1}^{n} \log f(X_i)
\end{aligned}$$

Remove constant multipliers and terms that don't include θ. We are left with trying to find a value of θ that maximizes:

$$\begin{aligned}
\hat{\theta} &= \underset{\theta}{\operatorname{argmax}} - \sum_{i=1}^{m} (Y_i - \theta X_i)^2 \\
&= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{m} (Y_i - \theta X_i)^2
\end{aligned}$$

This result says that the value of θ that makes the data most likely is one that minimizes the squared error of predictions of Y. We will see in a few days that this is the basis for linear regression.

Maximum A Posteriori

MLE is great, but it is not the only way to estimate parameters! This section introduces an alternate algorithm, Maximum A Posteriori (MAP).The paradigm of MAP is that we should chose the value for our parameters that is the most likely given the data. At first blush this might seem the same as MLE, however notice that MLE chooses the value of parameters that makes the \emph{data} most likely. Formally, for IID random variables $X_1, \ldots, X_n$:

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n)$$

In the equation above we trying to calculate the conditional probability of unobserved random variables given observed random variables. When that is the case, think Bayes Theorem! Expand the function f using the continuous version of Bayes Theorem.

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n) && \text{Now apply Bayes Theorem} \\
&= \underset{\theta}{\text{argmax}}\ \frac{f(X_1, X_2, \ldots, X_n | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Ahh much better}
\end{aligned}$$

Note that f, g and h are all probability densities. I used different symbols to make it explicit that they may have different functions. Now we are going to leverage two observations. First, the data is assumed to be IID so we can decompose the density of the data given θ. Second, the denominator is a constant with respect to θ. As such its value does not affect the argmax and we can drop that term. Mathematically:

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ \frac{\prod_{i=1}^{n} f(X_i | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{Since the samples are IID} \\
&= \underset{\theta}{\text{argmax}}\ \prod_{i=1}^{n} f(X_i | \theta) g(\theta) && \text{Since } h \text{ is constant with respect to } \theta
\end{aligned}$$

As before, it will be more convenient to find the argmax of the log of the MAP function, which gives us the final form for MAP estimation of parameters.

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i | \theta)) \right)$$

Using Bayesian terminology, the MAP estimate is the mode of the "posterior" distribution for θ. If you look at this equation side by side with the MLE equation you will notice that MAP is the argmax of the exact same function \emph{plus} a term for the log of the prior.

Parameter Priors

In order to get ready for the world of MAP estimation, we are going to need to brush up on our distributions. We will need reasonable distributions for each of our different parameters. For example, if you are predicting a Poisson distribution, what is the right random variable type for the prior of λ?

A desiderata for prior distributions is that the resulting posterior distribution has the same functional form. We call these "conjugate" priors. In the case where you are updating your belief many times, conjugate priors makes programming in the math equations much easier.

Here is a list of different parameters and the distributions most often used for their priors:

Parameter	Distribution
Bernoulli p	Beta
Binomial p	Beta
Poisson λ	Gamma
Exponential λ	Gamma
Multinomial p_i	Dirichlet
Normal μ	Normal
Normal σ^2	Inverse Gamma

You are only expected to know the new distributions on a high level. You do not need to know Inverse Gamma. I included it for completeness.

The distributions used to represent your "prior" belief about a random variable will often have their own parameters. For example, a Beta distribution is defined using two parameters (a, b). Do we have to use parameter estimation to evaluate a and b too? No. Those parameters are called "hyperparameters". That is a term we reserve for parameters in our model that we fix before running parameter estimate. Before you run MAP you decide on the values of (a, b).

Dirichlet

The Dirichlet distribution generalizes Beta in same way Multinomial generalizes Bernoulli. A random variable X that is Dirichlet is parametrized as $X \sim \text{Dirichlet}(a_1, a_2, \ldots, a_m)$. The PDF of the distribution is:

$$f(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = K \prod_{i=1}^{m} x_i^{a_i - 1}$$

Where K is a normalizing constant.

You can intuitively understand the hyperparameters of a Dirichlet distribution: imagine you have seen $\sum_{i=1}^{m} a_i - m$ imaginary trials. In those trials you had $(a_i - 1)$ outcomes of value i. As an example consider estimating the probability of getting different numbers on a six-sided Skewed Dice (where each side is a different shape). We will estimate the probabilities of rolling each side of this dice by repeatedly rolling the dice n times. This will produce n IID samples. For the MAP paradigm, we are going to need a prior on our belief of each of the parameters $p_1 \ldots p_6$. We want to express that we lightly believe that each roll is equally likely.

Before you roll, let's imagine you had rolled the dice six times and had gotten one of each possible values. Thus the "prior" distribution would be Dirichlet$(2, 2, 2, 2, 2, 2)$. After observing $n_1 + n_2 + \cdots + n_6$ new trials with n_i results of outcome i, the "posterior" distribution is Dirichlet($2 + n_1, \ldots 2 + n_6$). Using a prior which represents one imagined observation of each outcome is called "Laplace smoothing" and it guarantees that none of your probabilities are 0 or 1.

Gamma

The Gamma(k, θ) distribution is the conjugate prior for the λ parameter of the Poisson distribution (It is also the conjugate for Exponential, but we won't delve into that).

The hyperparameters can be interpreted as: you saw k total imaginary events during θ imaginary time periods. After observing n events during the next t time periods the posterior distribution is Gamma($k + n, \theta + t$).

For example Gamma(10, 5) would represent having seen 10 imaginary events in 5 time periods. It is like imagining a rate of 2 with some degree of confidence. If we start with that Gamma as a prior and then see 11 events in the next 2 time periods our posterior is Gamma(21,7) which is equivalent to an updated rate of 3.

Machine Learning

Machine Learning is the subfield of computer science that gives computers the ability to perform tasks without being explicitly programmed. There are several different tasks that fall under the domain of machine learning and several different algorithms for "learning". In this chapter, we are going to focus on Classification and two classic Classification algorithms: Naive Bayes and Logistic Regression.

Classification

In classification tasks, your job is to use training data with feature/label pairs ($\mathbf{x}$, y) in order to estimate a function $\hat{y} = g(\mathbf{x})$. This function can then be used to make a prediction. In classification the value of y takes on one of a \textbf{discrete} number of values. As such we often chose $g(\mathbf{x}) = \underset{y}{\text{argmax}}\ \hat{P}(Y = y|\mathbf{X})$.

In the classification task you are given N training pairs: $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$ Where $\mathbf{x}^{(i)}$ is a vector of m discrete features for the ith training example and $y^{(i)}$ is the discrete label for the ith training example.

In our introduction to machine learning, we are going to assume that all values in our training data-set are binary. While this is not a necessary assumption (both naive Bayes and logistic regression can work for non-binary data), it makes it much easier to learn the core concepts. Specifically we assume that all labels are binary $y^{(i)} \in \{0, 1\}\ \forall i$ and all features are binary $x_j^{(i)} \in \{0, 1\}\ \forall i, j$.

Naïve Bayes

Naive Bayes is a Machine Learning algorithm for the ``classification task". It make the substantial assumption (called the Naive Bayes assumption) that all features are independent of one another, given the classification label. This assumption is wrong, but allows for a fast and quick algorithm that is often useful. In order to implement Naive Bayes you will need to learn how to train your model and how to use it to make predictions, once trained.

Training (aka Parameter Estimation)

The objective in training is to estimate the probabilities $P(Y)$ and $P(X_i|Y)$ for all $0 < i \leq m$ features. We use the symbol $\hat{p}$ to make it clear that the probability is an estimate.

Using an MLE estimate:

$$\hat{p}(X_i = x_i|Y = y) = \frac{(\#\text{ training examples where } X_i = x_i \text{ and } Y = y)}{(\#\text{ training examples where } Y = y)}$$

Using a Laplace MAP estimate:

$$\hat{p}(X_i = x_i|Y = y) = \frac{(\#\text{ training examples where } X_i = x_i \text{ and } Y = y) + 1}{(\#\text{ training examples where } Y = y) + 2}$$

The prior probability of Y trained using an MLE estimate:

$$\hat{p}(Y = y) = \frac{(\#\text{ training examples where } Y = y)}{(\#\text{ training examples})}$$

Prediction

For an example with $\mathbf{x} = [x_1, x_2, \ldots, x_m]$, estimate the value of y as:

$$\hat{y} = \underset{y=\{0,1\}}{\arg\max}\ \log \hat{p}(Y = y) + \sum_{i=1}^{m} \log \hat{p}(X_i = x_i|Y = y)$$

Note that for small enough datasets you may not need to use the log version of the argmax.

Theory

In the world of classification when we make a prediction we want to chose the value of y that maximizes $P(Y = y|\mathbf{X})$.

$$\begin{aligned}
\hat{y} &= \underset{y=\{0,1\}}{\arg\max} P(Y = y|\mathbf{X} = \mathbf{X}) && \text{Our objective} \\
&= \underset{y=\{0,1\}}{\arg\max} \frac{P(Y = y)P(\mathbf{X} = \mathbf{x}|Y = y)}{P(\mathbf{X} = \mathbf{x})} && \text{By bayes theorem} \\
&= \underset{y=\{0,1\}}{\arg\max} P(Y = y)P(\mathbf{X} = \mathbf{x}|Y = y)) && \text{Since } P(\mathbf{X} = \mathbf{x}) \text{ is constant with respect to } Y
\end{aligned}$$

Using our training data we could interpret the joint distribution of $\mathbf{X}$ and Y as one giant multinomial with a different parameter for every combination of $\mathbf{X} = \mathbf{x}$ and $Y = y$. If for example, the input vectors are only length one. In other words $|\mathbf{x}| = 1$ and the number of values that x and y can take on are small, say binary, this is a totally reasonable approach. We could estimate the multinomial using MLE or MAP estimators and then calculate argmax over a few lookups in our table.

The bad times hit when the number of features becomes large. Recall that our multinomial needs to estimate a parameter for every unique combination of assignments to the vector $\mathbf{x}$ and the value y. If there are $|\mathbf{x}| = n$ binary features then this strategy is going to take order $\mathcal{O}(2^n)$ space and there will likely be many parameters that are estimated without any training data that matches the corresponding assignment.

Naive Bayes Assumption
The Naïve Bayes Assumption is that each feature of $\mathbf{x}$ is independent of one another given y.

The Naïve Bayes Assumption is wrong, but useful. This assumption allows us to make predictions using space and data which is linear with respect to the size of the features: $\mathcal{O}(n)$ if $|\mathbf{x}| = n$. That allows us to train and make predictions for huge feature spaces such as one which has an indicator for every word on the internet. Using this assumption the prediction algorithm can be simplified.

$$
\begin{aligned}
\hat{y} &= \arg\max_{y=\{0,1\}} P(Y=y)P(\mathbf{X}=\mathbf{x}|Y=y) && \text{As we last left off} \\
&= \arg\max_{y=\{0,1\}} P(Y=y)\prod_i P(X_i=x_i|Y=y) && \text{Naïve bayes assumption} \\
&= \arg\max_{y=\{0,1\}} \log P(Y=y) + \sum_i \log P(X_i=x_i|Y=y) && \text{For numerical stability}
\end{aligned}
$$

In the last step we leverage the fact that the argmax of a function is equal to the argmax of the log of a function. This algorithm is both fast and stable both when training and making predictions.

Logistic Regression

Logistic Regression is a classification algorithm (I know, terrible name. Perhaps Logistic Classification would have been better) that works by trying to learn a function that approximates $\mathrm{P}(y|x)$. It makes the central assumption that $\mathrm{P}(y|x)$ can be approximated as a sigmoid function applied to a linear combination of input features. It is particularly important to learn because logistic regression is the basic building block of artificial neural networks.

Mathematically, for a single training datapoint $(\mathbf{x}, y)$ Logistic Regression assumes:

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(z) \text{ where } z = \theta_0 + \sum_{i=1}^{m} \theta_i x_i$$

This assumption is often written in the equivalent forms:

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(\theta^T \mathbf{x}) \qquad \text{where we always set } x_0 \text{ to be 1}$$
$$P(Y = 0|\mathbf{X} = \mathbf{x}) = 1 - \sigma(\theta^T \mathbf{x}) \qquad \text{by total law of probability}$$

Using these equations for probability of $Y|X$ we can create an algorithm that selects values of theta that maximize that probability for all data. I am first going to state the log probability function and partial derivatives with respect to theta. Then later we will (a) show an algorithm that can chose optimal values of theta and (b) show how the equations were derived.

An important thing to realize is that: given the best values for the parameters (θ), logistic regression often can do a great job of estimating the probability of different class labels. However, given bad , or even random, values of θ it does a poor job. The amount of ``intelligence" that you logistic regression machine learning algorithm has is dependent on having good values of θ.

Notation

Before we get started I want to make sure that we are all on the same page with respect to notation. In logistic regression, θ is a vector of parameters of length m and we are going to learn the values of those parameters based off of n training examples. The number of parameters should be equal to the number of features of each datapoint.

Two pieces of notation that we use often in logistic regression that you may not be familiar with are:

$$\theta^T \mathbf{x} = \sum_{i=1}^{m} \theta_i x_i = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m \qquad \text{dot product, aka weighted sum}$$
$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad \text{sigmoid function}$$

Log Likelihood

In order to chose values for the parameters of logistic regression we use Maximum Likelihood Estimation (MLE). As such we are going to have two steps: (1) write the log-likelihood function and (2) find the values of θ that maximize the log-likelihood function.

The labels that we are predicting are binary, and the output of our logistic regression function is supposed to be the probability that the label is one. This means that we can (and should) interpret the each label as a Bernoulli random variable: $Y \sim \text{Bern}(p)$ where $p = \sigma(\theta^T \mathbf{x})$.

To start, here is a super slick way of writing the probability of one datapoint (recall this is the equation form of the probability mass function of a Bernoulli):

$$P(Y = y|X = \mathbf{x}) = \sigma(\theta^T \mathbf{x})^y \cdot \left[1 - \sigma(\theta^T \mathbf{x})\right]^{(1-y)}$$

Now that we know the probability mass function, we can write the likelihood of all the data:

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)} | X = \mathbf{x}^{(i)}) && \text{The likelihood of independent training labels} \\
&= \prod_{i=1}^{n} \sigma(\theta^T \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^T \mathbf{x}^{(i)})\right]^{(1-y^{(i)})} && \text{Substituting the likelihood of a Bernoulli}
\end{aligned}
$$

And if you take the log of this function, you get the reported Log Likelihood for Logistic Regression. The log likelihood equation is:

$$
LL(\theta) = \sum_{i=1}^{n} y^{(i)} \log \sigma(\theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \log[1 - \sigma(\theta^T \mathbf{x}^{(i)})]
$$

Recall that in MLE the only remaining step is to chose parameters (θ) that maximize log likelihood.

Gradient of Log Likelihood

Now that we have a function for log-likelihood, we simply need to chose the values of theta that maximize it. We can find the best values of theta by using an optimization algorithm. However, in order to use an optimization algorithm, we first need to know the partial derivative of log likelihood with respect to each parameter. First I am going to give you the partial derivative (so you can see how it is used). Then I am going to show you how to derive it:

$$
\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}
$$

Gradient Descent Optimization

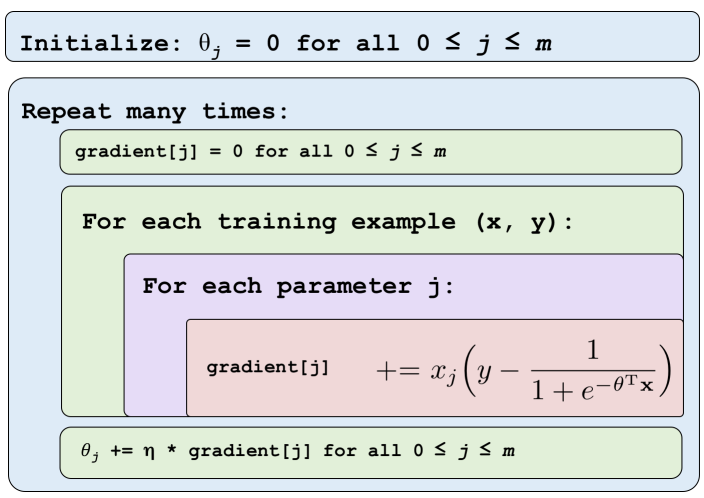
Our goal is to choosing parameters (θ) that maximize likelihood, and we know the partial derivative of log likelihood with respect to each parameter. We are ready for our optimization algorithm.

In the case of logistic regression we can't solve for θ mathematically. Instead we use a computer to chose θ. To do so we employ an algorithm called gradient descent (a classic in optimization theory). The idea behind gradient descent is that if you continuously take small steps downhill (in the direction of your negative gradient), you will eventually make it to a local minima. In our case we want to maximize our likelihood. As you can imagine, minimizing a negative of our likelihood will be equivalent to maximizing our likelihood.

The update to our parameters that results in each small step can be calculated as:

$$
\begin{aligned}
\theta_j^{\text{new}} &= \theta_j^{\text{old}} + \eta \cdot \frac{\partial LL(\theta^{\text{old}})}{\partial \theta_j^{\text{old}}} \\
&= \theta_j^{\text{old}} + \eta \cdot \sum_{i=1}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}
\end{aligned}
$$

Where η is the magnitude of the step size that we take. If you keep updating θ using the equation above you will converge on the best values of θ. You now have an intelligent model. Here is the gradient ascent algorithm for logistic regression in pseudo-code:



Pro-tip: Don't forget that in order to learn the value of θ_0 you can simply define $\mathbf{x}_0$ to always be 1.

Derivations

In this section we provide the mathematical derivations for the gradient of log-likelihood. The derivations are worth knowing because these ideas are heavily used in Artificial Neural Networks.

Our goal is to calculate the derivative of the log likelihood with respect to each theta. To start, here is the definition for the derivative of a sigmoid function with respect to its inputs:

$$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)[1-\sigma(z)] \qquad \text{to get the derivative with respect to } \theta\text{, use the chain rule}$$

Take a moment and appreciate the beauty of the derivative of the sigmoid function. The reason that sigmoid has such a simple derivative stems from the natural exponent in the sigmoid denominator.

Since the likelihood function is a sum over all of the data, and in calculus the derivative of a sum is the sum of derivatives, we can focus on computing the derivative of one example. The gradient of theta is simply the sum of this term for each training datapoint.

First I am going to show you how to compute the derivative the hard way. Then we are going to look at an easier method. The derivative of gradient for one datapoint $(\mathbf{x}, y)$:

$$\begin{aligned}
\frac{\partial LL(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} y \log \sigma(\theta^T\mathbf{x}) + \frac{\partial}{\partial \theta_j}(1-y)\log[1-\sigma(\theta^T\mathbf{x}] && \text{derivative of sum of terms} \\
&= \left[\frac{y}{\sigma(\theta^T\mathbf{x})} - \frac{1-y}{1-\sigma(\theta^T\mathbf{x})}\right]\frac{\partial}{\partial \theta_j}\sigma(\theta^T\mathbf{x}) && \text{derivative of } \log f(x) \\
&= \left[\frac{y}{\sigma(\theta^T\mathbf{x})} - \frac{1-y}{1-\sigma(\theta^T\mathbf{x})}\right]\sigma(\theta^T\mathbf{x})[1-\sigma(\theta^T\mathbf{x})]\mathbf{x}_j && \text{chain rule + derivative of sigma} \\
&= \left[\frac{y-\sigma(\theta^T\mathbf{x})}{\sigma(\theta^T\mathbf{x})[1-\sigma(\theta^T\mathbf{x})]}\right]\sigma(\theta^T\mathbf{x})[1-\sigma(\theta^T\mathbf{x})]\mathbf{x}_j && \text{algebraic manipulation} \\
&= \big[y-\sigma(\theta^T\mathbf{x})\big]\mathbf{x}_j && \text{cancelling terms}
\end{aligned}$$

Derivatives Without Tears

That was the hard way. Logistic regression is the building block of Artificial Neural Networks. If we want to scale up, we are going to have to get used to an easier way of calculating derivatives. For that we are going to have to welcome back our old friend the chain rule. By the chain rule:

$$\begin{aligned}
\frac{\partial LL(\theta)}{\partial \theta_j} &= \frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial \theta_j} && \text{Where } p = \sigma(\theta^T\mathbf{x}) \\
&= \frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} && \text{Where } z = \theta^T\mathbf{x}
\end{aligned}$$

Chain rule is the decomposition mechanism of calculus. It allows us to calculate a complicated partial derivative ($\frac{\partial LL(\theta)}{\partial \theta_j}$) by breaking it down into smaller pieces.

$$\begin{aligned}
LL(\theta) &= y \log p + (1-y)\log(1-p) && \text{Where } p = \sigma(\theta^T\mathbf{x}) \\
\frac{\partial LL(\theta)}{\partial p} &= \frac{y}{p} - \frac{1-y}{1-p} && \text{By taking the derivative}
\end{aligned}$$

$$\begin{aligned}
p &= \sigma(z) && \text{Where } z = \theta^T\mathbf{x} \\
\frac{\partial p}{\partial z} &= \sigma(z)[1-\sigma(z)] && \text{By taking the derivative of the sigmoid}
\end{aligned}$$

$$\begin{aligned}
z &= \theta^T\mathbf{x} && \text{As previously defined} \\
\frac{\partial z}{\partial \theta_j} &= \mathbf{x}_j && \text{Only } \mathbf{x}_j \text{ interacts with } \theta_j
\end{aligned}$$

Each of those derivatives was much easier to calculate. Now we simply multiply them together.

$$
\begin{aligned}
\frac{\partial LL(\theta)}{\partial \theta_j} &= \frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} \\
&= \left[\frac{y}{p} - \frac{1-y}{1-p}\right] \cdot \sigma(z)[1-\sigma(z)] \cdot \mathbf{x}_j && \text{By substituting in for each term} \\
&= \left[\frac{y}{p} - \frac{1-y}{1-p}\right] \cdot p[1-p] \cdot \mathbf{x}_j && \text{Since } p = \sigma(z) \\
&= [y(1-p) - p(1-y)] \cdot \mathbf{x}_j && \text{Multiplying in} \\
&= [y - p]\mathbf{x}_j && \text{Expanding} \\
&= [y - \sigma(\theta^T \mathbf{x})]\mathbf{x}_j && \text{Since } p = \sigma(\theta^T \mathbf{x})
\end{aligned}
$$

MLE Normal Demo

Lets manually perform maximum likelihood estimation. Your job is to chose parameter values that make the data look as likely as possible. Here are the 20 data points, which we assume come from a Normal distribution

Data = [6.3 , 5.5 , 5.4, 7.1, 4.6, 6.7, 5.3 , 4.8, 5.6, 3.4, 5.4, 3.4, 4.8, 7.9, 4.6, 7.0, 2.9, 6.4, 6.0 , 4.3]

Chose your parameter estimates

Parameter μ: 5 Parameter σ: 3

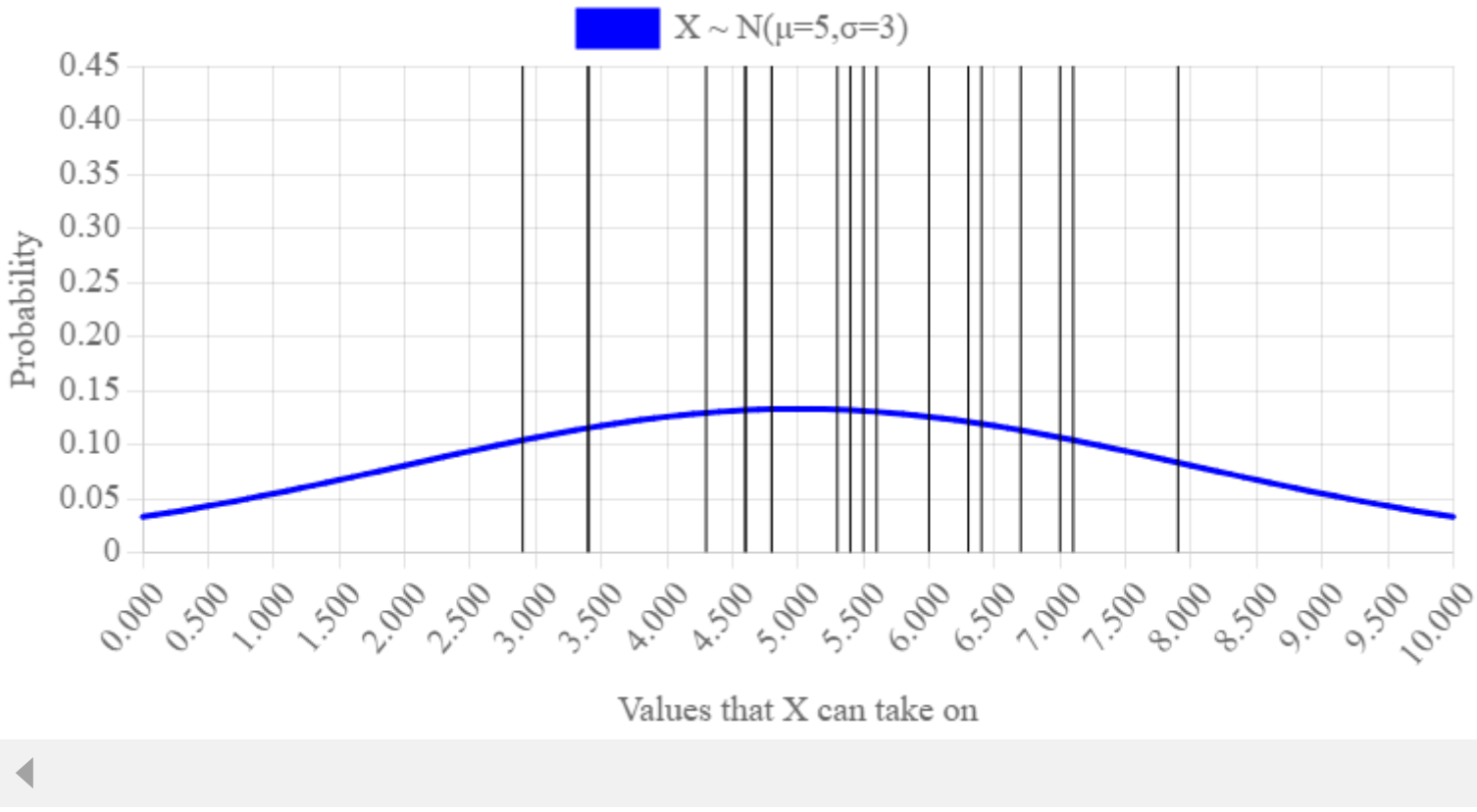
Likelihood of the data given your params

Likelihood: 4.0307253523200347e-19

Log Likelihood: -399.7

Best Seen: -399.7

Your Gaussian



MLE of a Pareto Distribution

You are creating artwork with different sized circles which follow a Pareto distribution:

$X \sim \text{Pareto}(\alpha)$

A Pareto distribution is defined by a single parameter α and has PDF

$$f(x) = \frac{\alpha}{x^{\alpha+1}}$$

You would like the alpha in your artwork to match that of sand in your local beach. You go to the beachand collect 100 particles of sand and measure their size. Call the measured radii $x_1, \ldots, x_{100}$:

```
observations = [1.677, 3.812, 1.463, 2.641, 1.256, 1.678, 1.157,
1.146, 1.323, 1.029, 1.238, 1.018, 1.171, 1.123, 1.074, 1.652,
1.873, 1.314, 1.309, 3.325, 1.045, 2.271, 1.305, 1.277, 1.114,
1.391, 3.728, 1.405, 1.054, 2.789, 1.019, 1.218, 1.033, 1.362,
1.058, 2.037, 1.171, 1.457, 1.518, 1.117, 1.153, 2.257, 1.022,
1.839, 1.706, 1.139, 1.501, 1.238, 2.53 , 1.414, 1.064, 1.097,
1.261, 1.784, 1.196, 1.169, 2.101, 1.132, 1.193, 1.239, 1.518,
2.764, 1.053, 1.267, 1.015, 1.789, 1.099, 1.25 , 1.253, 1.418,
1.494, 1.015, 1.459, 2.175, 2.044, 1.551, 4.095, 1.396, 1.262,
1.351, 1.121, 1.196, 1.391, 1.305, 1.141, 1.157, 1.155, 1.103,
1.048, 1.918, 1.889, 1.068, 1.811, 1.198, 1.361, 1.261, 4.093,
2.925, 1.133, 1.573]
```

Derive a formula for the MLE estimate of α based on the data you have collected.

Writing the Log Likihood Function

The first major objective in MLE is to come up with a log likelihood expression for our data. To do so we start by writing how likely our dataset looks, if we are told the value of α:

$$L(\alpha) = f(x_1 \ldots x_n) = \prod_{i=1}^{n} \frac{\alpha}{x_i^{\alpha+1}}$$

Optimization will be much easier if we instead try to optimize the log liklihood:

$$\begin{aligned} LL(\alpha) &= \log L(\alpha) = \log \prod_{i=1}^{n} \frac{\alpha}{x_i^{\alpha+1}} \\ &= \sum_{i=1}^{n} \log \frac{\alpha}{x_i^{\alpha+1}} \\ &= \sum_{i=1}^{n} \log \alpha - (\alpha+1) \log x_i \\ &= n \log \alpha - (\alpha+1) \sum_{i=1}^{n} \log x_i \end{aligned}$$

Selecting α

We are going to select α to be the value which maximizes the log likelihood. To do so we are going to need the derivative of LL w.r.t. α

$$\begin{aligned} \frac{\partial LL(\alpha)}{\partial \alpha} &= \frac{\partial LL(\alpha)}{\partial \alpha} \left(n \log \alpha - (\alpha+1) \sum_{i=1}^{n} \log x_i \right) \\ &= \frac{n}{\alpha} - \sum_{i=1}^{n} \log x_i \end{aligned}$$

One way to optimize is to take the derivative and set it equal to zero:

$$
\begin{aligned}
0 &= \frac{n}{\alpha} - \sum_{i=1}^{n} \log x_i \\
\sum_{i=1}^{n} \log x_i &= \frac{n}{\alpha} \\
\alpha &= \frac{n}{\sum_{i=1}^{n} \log x_i}
\end{aligned}
$$

At this point we have a formula that we can use to calculate α! Wahoo

Putting it into code

```python
import math

def estimate_alpha(observations):
    # This code computes the MLE estimate of alpha
    log_sum = 0
    for x_i in observations:
        log_sum += math.log(x_i)
    n = len(observations)
    return n / log_sum

def main():
    observations = [1.677, 3.812, 1.463, 2.641, 1.256, 1.678, 1.157, 1.146,
    1.323, 1.029, 1.238, 1.018, 1.171, 1.123, 1.074, 1.652, 1.873, 1.314,
    1.309, 3.325, 1.045, 2.271, 1.305, 1.277, 1.114, 1.391, 3.728, 1.405,
    1.054, 2.789, 1.019, 1.218, 1.033, 1.362, 1.058, 2.037, 1.171, 1.457,
    1.518, 1.117, 1.153, 2.257, 1.022, 1.839, 1.706, 1.139, 1.501, 1.238,
    2.53 , 1.414, 1.064, 1.097, 1.261, 1.784, 1.196, 1.169, 2.101, 1.132,
    1.193, 1.239, 1.518, 2.764, 1.053, 1.267, 1.015, 1.789, 1.099, 1.25 ,
    1.253, 1.418, 1.494, 1.015, 1.459, 2.175, 2.044, 1.551, 4.095, 1.396,
    1.262, 1.351, 1.121, 1.196, 1.391, 1.305, 1.141, 1.157, 1.155, 1.103,
    1.048, 1.918, 1.889, 1.068, 1.811, 1.198, 1.361, 1.261, 4.093, 2.925,
    1.133, 1.573]
    alpha = estimate_alpha(observations)
    print(alpha)

if __name__ == '__main__':
    main()
```

Gaussian Mixtures

Data = [6.47, 5.82, 8.7, 4.76, 7.62, 6.95, 7.44, 6.73, 3.38, 5.89, 7.81, 6.93, 7.23, 6.25, 5.31, 7.71, 7.42, 5.81, 4.03, 7.09, 7.1, 7.62, 7.74, 6.19, 7.3, 7.37, 6.99, 2.97, 3.3, 7.08, 6.23, 3.67, 3.05, 6.67, 6.5, 6.08, 3.7, 6.76, 6.56, 3.61, 7.25, 7.34, 6.27, 6.54, 5.83, 6.44, 5.34, 7.7, 4.19, 7.34]

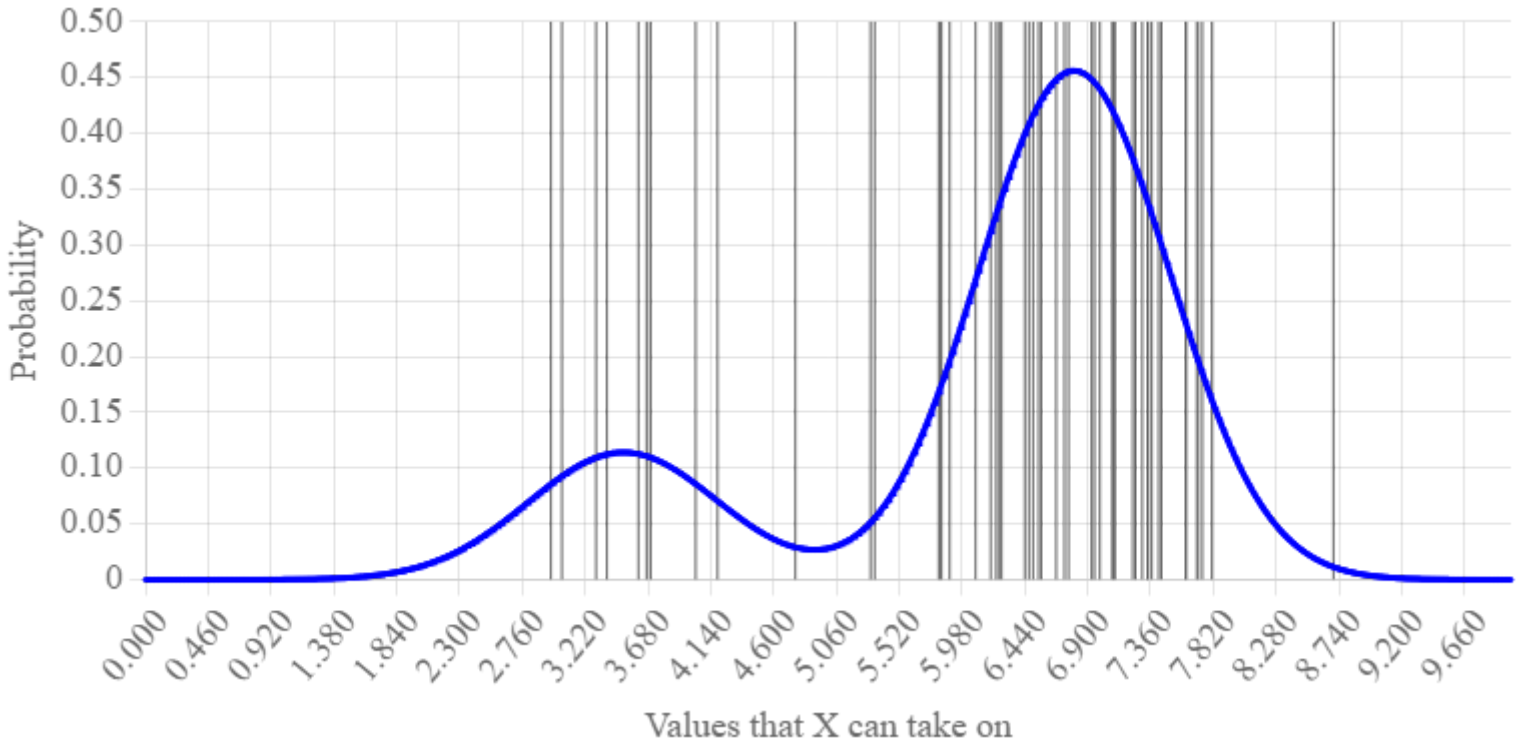
Parameter t: 0.2

Parameter μ_a: 3.5

Parameter σ_a: 0.7

Parameter μ_b: 6.8

Parameter σ_b: 0.7



Likelihood: 1.847658621579746e-34

Log Likelihood: -77.7

Best Seen: -77.7

What is a Gaussian Mixture?

A Gaussian Mixture describes a random variable whose PDF could come from one of two Gaussians (or more, but we will just use two in this demo). There is a certain probability the sample will come from the first gaussian, otherwise it comes from the second. It has five parameters: 4 to describe the two gaussians and one to describe the relative weighting of the two gaussians.

Generative Code

```
from scipy import stats
def sample():
   # choose group membership
   membership = stats.bernoulli.rvs(0.2)
   if membership == 1:
      # sample from gaussian 1
      return stats.norm.rvs(3.5,0.7)
   else:
      # sample from gaussian 2
      return stats.norm.rvs(6.8,0.7)
```

Probability Density Function

$$f(X = x) = t \cdot f(A = x) + (1 - t) \cdot f(B = x)$$

$$\text{st}$$

$$A \sim N(\mu_a, \sigma_a^2)$$

$$B \sim N(\mu_b, \sigma_b^2)$$

Putting it all together, the PDF of a Gaussian Mixture is:

$$f(x) = t \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{1}{2}(\frac{x-\mu_a}{\sigma_a})^2}\right) + (1-t) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}(\frac{x-\mu_b}{\sigma_b})^2}\right)$$

MLE for Gaussian Mixture

Special note: even though the generative story has a bernoulli (group membership) it is never observed. MLE maximizes the likelihood of the observed data.

Let $\vec{\theta} = [t, \mu_a, \mu_b, \sigma_a, \sigma_b]$ be the parameters. Because the math will get long I will use θ as notation in place of $\vec{\theta}$. Just keep in mind that it is a vector.

The MLE idea is to chose values of θ which maximize log likelihood. All optimization methods require us to calculate the partial derivatives of the thing we want to optimize (log likelihood) with respect to the values we can change (our parameters).

Likelihood function

$$L(\theta) = \prod_i^n f(x_i|\theta)$$
$$= \prod_i^n \left[t \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\right) + (1-t) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}(\frac{x_i-\mu_b}{\sigma_b})^2}\right)\right]$$

Log Likelihood function

$$\begin{aligned} LL(\theta) &= \log L(\theta) \\ &= \log \prod_i^n f(x_i|\theta) \\ &= \sum_i^n \log f(x_i|\theta) \end{aligned}$$

That is sufficient for now, but if you wanted to expand out the term you would get:

$$LL(\theta) = \sum_i^n \log \left[t \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\right) + (1-t) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}(\frac{x_i-\mu_b}{\sigma_b})^2}\right)\right]$$

Derivative of LL with respect to θ

Here is an example of calculating a partial derivative with respect to one of the parameters, μ_a. You would need a derivative like this for all parameters.

Caution: When I first wrote this demo I thought it would be a simple derivative . It is not so simple because the log has a sum in it. As such the log term doesn't reduce. The log still serves to make the outer $\prod$ into a $\sum$. As such the LL partial derivatives are solvable, but the proof uses quite a lot of chain rule.

Takeaway: The main takeaway from this section (in case you want to skip the derivative proof) is that the resulting derivative is complex enough that we will want a way to compute argmax without having to set that derivative equal to zero and solving for μ_a. Enter gradient descent!

A good first step when doing a huge derivative of a log likelihood function is to think of the derivative for the log of likelihood of a single datapoint. This is the inner sum in the log likelihood expression:

$$\frac{d}{d\mu_a} \log f(x_i|\theta)$$

Before we start: notice that μ_a does not show up in this term from $f(x_i|\theta)$:

$$(1-t) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}(\frac{x_i-\mu_b}{\sigma_b})^2}\right) = K$$

In the proof, when we encounter this term, we are going to think of it as a constant which we call K. Ok, lets go for it!

$$
\begin{aligned}
&\frac{d}{d\mu_a}\log f(x_i|\theta) \\
&\quad = \frac{1}{f(x_i|\theta)}\frac{d}{d\mu_a}f(x_i|\theta) && \text{chain rule on } \log \\
&\quad = \frac{1}{f(x_i|\theta)}\frac{d}{d\mu_a}\left[t\cdot\left(\frac{1}{\sqrt{2\pi}\sigma_a}e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\right)+K\right] && \text{substitute in } f(x_i|\theta) \\
&\quad = \frac{1}{f(x_i|\theta)}\frac{d}{d\mu_a}\left[t\cdot\left(\frac{1}{\sqrt{2\pi}\sigma_a}e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\right)\right] && \frac{d}{d\mu_a}K=0 \\
&\quad = \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a}\cdot\frac{d}{d\mu_a}e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2} && \text{pull out const} \\
&\quad = \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a}\cdot e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\cdot\frac{d}{d\mu_a}-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2 && \text{chain on } e^x \\
&\quad = \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a}\cdot e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\cdot\left[-(\frac{x_i-\mu_a}{\sigma_a})\frac{d}{d\mu_a}(\frac{x_i-\mu_a}{\sigma_a})\right] && \text{chain on } x^2 \\
&\quad = \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a}\cdot e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\cdot\left[-(\frac{x_i-\mu_a}{\sigma_a})\cdot\frac{-1}{\sigma_a}\right] && \text{final derivative} \\
&\quad = \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a^3}\cdot e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\cdot(x_i-\mu_a) && \text{simplify}
\end{aligned}
$$

That was for a single data-point. For the full dataset:

$$
\begin{aligned}
\frac{dLL(\theta)}{d\mu_a} &= \sum_i^n \frac{d}{d\mu_a}\log f(x_i|\theta) \\
&= \sum_i^n \frac{t}{f(x_i|\theta)\sqrt{2\pi}\sigma_a^3}\cdot e^{-\frac{1}{2}(\frac{x_i-\mu_a}{\sigma_a})^2}\cdot(x_i-\mu_a)
\end{aligned}
$$

This process should be repeated for all five parameters! Now, how should we find a value of μ_a, which, in the presence of the other settings to parameters, and the data, makes this derivative zero? Setting the derivative = 0 and solving for μ_a is not going to work.

Use an Optimizer to Estimate Params

Once we have a LL function and the derivative of LL with respect to each parameter we are ready to compute argmax using an optimizer. In this case the best choice would probably be gradient ascent (or gradient descent with negative log likelihood).

$$
\nabla_\theta LL(\theta) = \begin{bmatrix}
\frac{dLL(\theta)}{dt} \\
\frac{dLL(\theta)}{d\mu_a} \\
\frac{dLL(\theta)}{d\mu_b} \\
\frac{dLL(\theta)}{d\sigma_a} \\
\frac{dLL(\theta)}{d\sigma_b}
\end{bmatrix}
$$