Statistics and Applications {ISSN 2454-7395 (online)}
Special Issue on Recent Advances in Bayesian Statistics
and Machine Learning
Volume 24, No. 2, 2026 (New Series), pp 211–245
http://www.ssca.org.in/journal

From Classical Probability to Quantum Amplitudes: A Pedagogical Introduction for Statisticians

Shyam Dhamapurkar[1] **and Sourish Das**[2]

[1]*Computer Science, Chennai Mathematical Institute*

[2]*Mathematics, Chennai Mathematical Institute*

Received: 9 May 2026; Revised: 10 July 2026; Accepted: 13 August 2026

Abstract

Quantum computing offers provable speedups for fundamental statistical tasks: estimating means, solving linear systems, and sampling from posterior distributions. This article provides a self-contained, pedagogical bridge from classical probability and statistics to their quantum analogues. We begin with Kolmogorov's axioms and build up through conditional probability, Bayes' rule, random variables, expectation, sampling distributions, the central limit theorem, maximum likelihood estimation, and Bayesian inference. For each concept, we present a quantum counterpart using only linear algebra. A single running example, *estimating the bias of a coin*, illustrates both classical and quantum approaches throughout. We give detailed, step-by-step explanations of Grover's search algorithm Grover (1996), Quantum Amplitude Estimation Brassard *et al.* (2002), the HHL algorithm Harrow *et al.* (2009), and Quantum Markov Chain Monte Carlo Szegedy (2004). Throughout, we state the precise regime of validity of each complexity bound and distinguish query complexity from total gate complexity; edge cases and technical caveats are kept brief and clearly marked so they inform without interrupting the main narrative. The article is self-contained and requires no prior knowledge of quantum mechanics. For deeper algorithmic detail, readers may consult the comprehensive review by Lopatnikova *et al.* (2024) or the standard textbook by Nielsen and Chuang (2010).

Key words: Quantum statistics; Quantum states; Quantum coin toss; Exponential efficiency; Binomial distribution as quantum state; Quantum expectation and variance.

AMS Subject Classifications: 62K05, 05B05

1. Introduction

Classical statistics and classical computing grew up together, like twins who learned to walk side by side. Markov chain Monte Carlo Geyer (1992), the bootstrap, the EM algo-

Corresponding Author: Sourish Das
Email: sourish@cmi.ac.in

rithm, and gradient descent were all designed for classical computers; machines that operate according to the rules of classical physics. These machines can copy data arbitrarily (try copying a file on your computer), reset bits to zero (delete a file), and execute deterministic logic (if this, then that). We have become so accustomed to these capabilities that we rarely think about them.

Why should a statistician care about quantum computing?

Quantum computers operate under a different set of physical laws. They leverage three properties that have no classical analogues:

- **Superposition**: A quantum system can exist in multiple states simultaneously, like a coin spinning in the air before it lands.
- **Entanglement**: Quantum systems can be correlated in ways that cannot be reproduced by any classical system, even with hidden variables.
- **Interference**: Quantum amplitudes (the "square roots" of probabilities) can add constructively or destructively, allowing us to amplify desired outcomes and suppress undesired ones.

These properties are not just mathematical curiosities. They enable quantum computers to solve certain problems fundamentally faster than any classical computer ever could. For statisticians, this is exciting news because many of our core tasks, estimating means, solving linear systems, sampling from posterior distributions, are exactly the kinds of problems where quantum computers offer provable speedups. But there is a catch. Quantum computers do not think like classical computers. We cannot simply take existing `R` or `Python` code and run it on a quantum computer. We must rethink how we represent data, how we compute expectations, and how we perform inference. This article is designed to help build that new mental model, one classical concept at a time.

What this article is, and what it is not

This article is a self-contained, pedagogical introduction to quantum computing for people who already know probability and statistics. It provides a conceptual bridge that maps classical statistical concepts, probability distributions, random variables, expectation, Bayes' rule,to their quantum analogues, and presents a step-by-step guide to the most important quantum algorithms for statisticians: Grover's search Grover (1996), Quantum Amplitude Estimation Brassard *et al.* (2002), the HHL algorithm Harrow *et al.* (2009), and Quantum Markov Chain Monte Carlo Szegedy (2004). It is, throughout, a practical discussion of when quantum advantage is plausible and what bottlenecks remain, we are as interested in where these algorithms fall short as in where they succeed.

Comprehensive surveys of quantum algorithms for statistics already exist, most notably the recent review by Lopatnikova *et al.* (2024), which catalogues a wide range of algorithms and provides detailed complexity analyses. However, such surveys typically assume a degree of comfort with quantum notation, Dirac bra-ket formalism, or computational

complexity theory that a classically trained statistician may not yet have. This article is designed as a *translation guide*: every quantum concept is introduced *only after* its classical statistical counterpart has been established, and the two are compared side by side. The goal is not to replace Lopatnikova *et al.* (2024) or Nielsen and Chuang (2010), but to equip the reader with the vocabulary and intuition needed to tackle those heavier texts with confidence.

This article is not a comprehensive survey of all quantum algorithms (Lopatnikova *et al.* (2024) serves that purpose). It is not a guide to programming quantum computers, though we provide references to Qiskit and other SDKs at the end. And it is not a physics textbook: we use only linear algebra, no physics beyond the basic postulates of measurement and unitary evolution.

The running example: a biased coin

> **The problem:** We have a coin that may be unfair. It comes up Heads with probability θ and Tails with probability $1 - \theta$. We do not know θ. We flip the coin n times and observe k heads. What can we say about θ?

Throughout this article, we will return again and again to this single, simple problem: estimating the bias of a coin. This problem is deceptively simple. It is the "Hello, World" of statistics (Casella and Berger, 2002). It appears in every introductory textbook. And yet, it illustrates every major concept we will need: probability axioms, conditional probability and Bayes' rule, random variables, expectation and variance, sampling distributions and the CLT, MLE, and Bayesian inference. By the end of this article we will know the **quantum** analogue of each of these concepts. We will understand how a quantum computer can estimate θ with quadratically fewer oracle queries than a classical computer needs samples. We will see how Grover's search (Grover, 1996) can find the maximum of a posterior distribution faster than classical search. And we will appreciate the practical challenges, data loading and readout, that currently limit how much of this advantage can be realised on actual hardware.

If you understand how the classical and quantum approaches differ for this one simple problem, you will understand how they differ for any problem. Now, let us build the classical foundation.

Prerequisites and notation

The only mathematical prerequisites for this article are threefold. First, **basic probability**: the reader should know what a probability space is, what conditional probability means, and what expectation does; Chapters 1–3 of Casella and Berger (2002) are more than sufficient background. Second, **linear algebra**: the reader should be comfortable with vectors, matrices, inner products, eigenvalues, and eigenvectors. We do not need advanced topics like singular value decomposition, though we will mention it in passing; for quantum computing specifically, the standard reference is Nielsen and Chuang (2010). Third, **basic calculus**: one should know how to take derivatives and integrals, which we will need for maximum likelihood estimation and for locating the posterior mode.

Equally important is what we do *not* need. We do not assume any physics knowledge, no Schrödinger equation, no Hamiltonian mechanics, no quantum field theory. We do not assume any prior exposure to quantum computing whatsoever. And we do not assume any programming experience, though we do mention programming platforms briefly near the end of the article for readers who want to experiment hands-on.

We use standard statistical notation throughout ($\mathbb{E}[X]$, $\mathrm{Var}(X)$, $\mathrm{Cov}(X, Y)$, $N(\mu, \sigma^2)$, $\mathrm{Binomial}(n, \theta)$, $\mathrm{Beta}(\alpha, \beta)$), alongside Dirac's bra-ket notation, which we introduce here as a quick reference before we begin using it in earnest:

$\lvert\psi\rangle$	A column vector (pronounced "ket psi")
$\langle\psi\rvert$	The conjugate transpose of $\lvert\psi\rangle$ ("bra psi")
$\langle\phi\rvert\psi\rangle$	The inner product of $\lvert\phi\rangle$ and $\lvert\psi\rangle$
$\langle\psi\rvert O\lvert\psi\rangle$	The expectation value $\langle\psi\rvert O\lvert\psi\rangle$, *i.e.*, $\langle\psi\rvert O\lvert\psi\rangle$
$\lvert\psi\rangle\langle\psi\rvert$	The outer product $\lvert\psi\rangle\langle\psi\rvert$ (a projection operator)
$\lvert 0\rangle, \lvert 1\rangle$	Standard basis vectors: $\begin{pmatrix}1\\0\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix}$

The key thing to remember is that $\lvert\psi\rangle$ is just a vector; the notation is a convention, not new mathematics. Note that $\langle\phi\rvert\psi\rangle$ always has exactly two slots (an inner product), while $\langle\psi\rvert O\lvert\psi\rangle$ is a different shape, an operator sandwiched between a bra and a ket; we use the two notations consistently throughout so they are never conflated.

A word of encouragement. Richard Feynman famously said, "If you think you understand quantum mechanics, you don't understand quantum mechanics." Do not be discouraged if some concepts feel counterintuitive at first; the mathematics is linear algebra you already know, and the intuition comes with practice.

How to read this article

This article is designed to be read sequentially: each section builds on the previous ones, and the running coin example is meant to be tracked continuously rather than treated as a series of disconnected illustrations. That said, readers already comfortable with classical probability and statistics may skim Sections 2–4, which establish the classical foundations most statisticians already know well, and focus their attention on the quantum analogues developed in Sections 5 through 9, which is where the genuinely new material lives. Readers who want the algorithmic payoff first and the careful justification afterward might also choose to read the four algorithm sections, Sections 6, 7, 8, and 9, somewhat out of order, since each is largely self-contained once Sections 2 through 5 have been absorbed; we have tried to make the cross-references explicit enough that this is a safe way to read the article.

2. Classical probability axioms and quantum states

Why do we need axioms?

Before we write down the axioms, let us ask a more fundamental question: why do we need axioms for probability at all? The answer is that probability is a mathematical theory,

and like all mathematical theories (Euclidean geometry, group theory, real analysis), it needs a set of foundational assumptions from which all other results can be derived. The axioms ensure that probability is self-consistent and that different people using the same rules will reach the same conclusions. The modern axiomatic foundation of probability was laid by the Russian mathematician Andrey Kolmogorov in 1933. Before Kolmogorov, probability was a collection of useful but loosely connected techniques. After Kolmogorov, probability became a rigorous branch of measure theory, on par with real analysis or topology (Billingsley, 1995).

We begin with the definition of a probability space. This is, as we will see shortly, the classical analogue of what will later become a Hilbert space in the quantum setting, so it is worth paying close attention to the structure.

Definition 1 (Probability space)**:** A probability space is a triple $(\Omega, \mathcal{F}, P)$ where Ω is the sample space (all possible outcomes), $\mathcal{F}$ is a σ-algebra of measurable events, and $P : \mathcal{F} \to [0, 1]$ is a probability measure.

The sample space Ω

The sample space is simply the set of all possible outcomes of an experiment. It can be finite, countably infinite, or uncountably infinite, and the three cases behave quite differently.

Example 1 (Finite sample space)**:** For a single coin flip, $\Omega = \{H, T\}$. There are exactly two possible outcomes.

Example 2 (Countably infinite sample space)**:** For flipping a coin until the first head appears, $\Omega = \{H, TH, TTH, TTTH, \ldots\}$. There are infinitely many outcomes, but we can list them in a sequence.

Example 3 (Uncountable sample space)**:** For measuring the exact waiting time until a radioactive decay, $\Omega = [0, \infty)$. There are uncountably many possible outcomes.

The σ-algebra $\mathcal{F}$

Not every subset of Ω can be assigned a probability. This is a technical subtlety that arises specifically for uncountable sample spaces; it is related to the Banach–Tarski paradox and the existence of non-measurable sets. The σ-algebra $\mathcal{F}$ is the collection of subsets to which we *can* assign probabilities, and it must satisfy three properties: (1) $\Omega \in \mathcal{F}$ (the whole space is an event); (2) if $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$ (complements are events); (3) if $A_1, A_2, \ldots \in \mathcal{F}$ then $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ (countable unions are events). For discrete sample spaces we usually take $\mathcal{F}$ to be the power set of Ω (all subsets); for continuous sample spaces we take the Borel σ-algebra, generated by open intervals.

Definition 2 (Kolmogorov's axioms)**:** The probability measure P satisfies: **(1) Non-negativity**: $P(A) \geq 0$. **(2) Normalization**: $P(\Omega) = 1$. **(3) Countable additivity**: for disjoint $A_1, A_2, \ldots \in \mathcal{F}$, $P(\bigcup_i A_i) = \sum_i P(A_i)$.

From these three axioms alone we can derive finite additivity, the complement rule

$P(A^c) = 1 - P(A)$, monotonicity, and inclusion-exclusion.

Example 4 (Coin flip)**:** $\Omega = \{H, T\}$, $\mathcal{F}$ = all subsets, $P(\{H\}) = \theta$, $P(\{T\}) = 1 - \theta$. Non-negativity holds since $\theta, 1 - \theta \geq 0$; normalization holds since $\theta + (1 - \theta) = 1$; countable additivity reduces to finite additivity here.

From classical probability to quantum mechanics: a parallel structure

Now that we have the classical foundation, let us observe a striking parallel. In classical probability we have: Sample space $\Omega \xrightarrow{\text{events}}$ Probability measure $P \xrightarrow{\text{outcomes}}$ Random variable X. In quantum mechanics, the structure is similar but with complex numbers replacing real numbers and vectors replacing sets: Hilbert space $\mathcal{H} \xrightarrow{\text{subspaces}}$ State $|\psi\rangle \xrightarrow{\text{measurement}}$ Observable O.

This parallel is not accidental. Quantum mechanics is, in a deep sense, a non-commutative generalisation of probability theory. Where classical probabilities are real numbers between 0 and 1 that add, quantum "probabilities" are squared magnitudes of complex amplitudes that, for orthogonal events, also add; the distinctive new feature is that for *non-orthogonal* decompositions the amplitudes can cancel before squaring. For a thorough introduction, see Nielsen and Chuang (2010).

The quantum analogue of a probability space is a Hilbert space; the definition below lists three properties an inner product must satisfy, each of which is worth pausing on since they recur constantly in everything that follows.

Definition 3 (Hilbert space)**:** A Hilbert space $\mathcal{H}$ is a vector space over the complex numbers $\mathbb{C}$ equipped with an inner product $\langle \cdot, \cdot \rangle$ that is: (1) *linear in the second argument*, $\langle \phi, a\psi_1 + b\psi_2 \rangle = a\langle \phi, \psi_1 \rangle + b\langle \phi, \psi_2 \rangle$; (2) *conjugate symmetric*, $\langle \phi, \psi \rangle = \overline{\langle \psi, \phi \rangle}$; and (3) *positive definite*, $\langle \psi, \psi \rangle \geq 0$ with equality only if $\psi = 0$. The space is complete with respect to the induced norm $\|\psi\| = \sqrt{\langle \psi, \psi \rangle}$. For our purposes, finite-dimensional Hilbert spaces $\mathcal{H} \cong \mathbb{C}^d$ are entirely sufficient, we never need the infinite-dimensional machinery that a full treatment of quantum mechanics eventually requires.

Remark 1 (Why complex numbers?)**:** This is a natural question to ask. Why do quantum states use complex amplitudes rather than real numbers? The answer is that complex numbers are necessary to capture interference phenomena. With only real amplitudes, the only possible interference is constructive (same sign) or destructive (opposite sign), but complex phases allow for far more subtle interference patterns than a real-valued theory could produce. This is not merely a mathematical convenience adopted for elegance; experiments confirm that quantum mechanics genuinely requires complex amplitudes to match observed reality (Nielsen and Chuang, 2010).

The fundamental object in quantum mechanics is the quantum state itself.

Definition 4 (Pure quantum state)**:** A pure quantum state is a unit vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$. In an orthonormal basis $\{|i\rangle\}_{i=0}^{N-1}$, we can write $|\psi\rangle = \sum_i \psi_i |i\rangle$ with $\sum_i |\psi_i|^2 = 1$; the complex numbers ψ_i are called **amplitudes**.

The notation $|\psi\rangle$ (pronounced "ket psi") is, as a column vector, $|\psi\rangle = (\psi_0, \ldots, \psi_{N-1})^\top$; the corresponding row vector is $\langle\psi| = (\psi_0^*, \ldots, \psi_{N-1}^*)$, where $*$ denotes complex conjugation, and the inner product of two states is $\langle\phi|\psi\rangle = \sum_i \phi_i^* \psi_i$.

The key difference: amplitudes vs. probabilities

Classical probability distributions are vectors of non-negative real numbers that sum to 1: $p = (p_0, \ldots, p_{N-1})$, $p_i \geq 0$, $\sum_i p_i = 1$. Quantum states are vectors of complex numbers whose squared magnitudes sum to 1: $\psi = (\psi_0, \ldots, \psi_{N-1})$, $\psi_i \in \mathbb{C}$, $\sum_i |\psi_i|^2 = 1$. The difference looks small on the page but is profound in its consequences. Because amplitudes can be negative or complex, they can **interfere**: two paths to the same outcome can have amplitudes with opposite signs and cancel, reducing, or even eliminating, the probability of that outcome. This is flatly impossible in classical probability, where probabilities are non-negative numbers that can only add constructively; there is no classical mechanism by which two positive probabilities can combine to give something smaller than either one.

Example 5 (Interference)**:** Consider the quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. The probability of measuring 0 is $|\frac{1}{\sqrt{2}}|^2 = 1/2$. Now consider a transformation that sends $|1\rangle$ to $-|1\rangle$; the new state is $|\psi'\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. If we measure immediately in the original basis, $P(0)$ is still $1/2$, the individual probabilities have not changed at all, since the sign flip on $|1\rangle$ is invisible to a measurement that only looks at $|0\rangle$ versus $|1\rangle$. But if we first apply this transformation and *then* measure in a different basis, the interference becomes visible, and the resulting probabilities can differ substantially from what either branch alone would predict. This is the principle behind Grover's search algorithm (Section 6) and, in one form or another, behind every quantum speedup discussed in this article.

The Born rule: from amplitudes to probabilities

The Born rule connects the quantum state (amplitudes) to observable outcomes (probabilities). It is named after Max Born, who proposed it in 1926 and later received the Nobel Prize for this work.

Definition 5 (Born rule)**:** A measurement of $|\psi\rangle$ in basis $\{|i\rangle\}$ yields outcome i with probability $P(i) = |\langle i|\psi\rangle|^2 = |\psi_i|^2$; the state then **collapses** to $|i\rangle$.

The collapse is a distinctive feature of quantum mechanics. Before measurement, the system is in a superposition of many possibilities. At the moment of measurement, it "chooses" one outcome, and the state becomes that outcome. This is not an approximation or a lack of knowledge, it is a fundamental feature of the theory. (The mechanism of collapse is still debated philosophically, but the mathematical rule itself is not.) In classical probability, if you have a random variable X with distribution $p(x)$, and you observe $X = x_0$, you update your beliefs to "X is definitely x_0", this is similar to collapse. However, there is no classical analogue of superposition: a classical system cannot be simultaneously in two distinct states with coherent amplitudes the way a quantum system can.

It is important to be precise about what the Born rule does and does not preserve

from Kolmogorov's axioms. For a *fixed* basis $\{|i\rangle\}$, $P(i) = |\psi_i|^2$ satisfies countable additivity exactly as required: for disjoint $A, B \subseteq \{i\}$,

$$P(A \cup B) = \sum_{i \in A \cup B} |\psi_i|^2 = \sum_{i \in A} |\psi_i|^2 + \sum_{i \in B} |\psi_i|^2 = P(A) + P(B),$$

with *no* interference term, because A and B correspond to orthogonal projectors in the same basis. Interference enters only when amplitudes from a *non-orthogonal* decomposition are summed before squaring: if $|\phi\rangle = \alpha|u\rangle + \beta|v\rangle$ for non-orthogonal $|u\rangle, |v\rangle$, the cross terms $\alpha^*\beta\langle u|v\rangle + \alpha\beta^*\langle v|u\rangle$ generally do not vanish. This is the genuine source of interference and of every speedup in this article, a statement about non-orthogonal decompositions, not a blanket failure of additivity for the Born rule itself.

Mixed states and density matrices

Not all quantum states we will need to work with are pure. Sometimes we have classical uncertainty about which pure state a system is actually in, for example, we might prepare a qubit in state $|0\rangle$ with probability 1/2 and in state $|1\rangle$ with probability 1/2, perhaps because of an imperfect preparation procedure. This kind of situation, where there is a genuine classical probability distribution over which pure state we have, is called a **mixed state**, and it requires a more general mathematical object than a single ket to describe.

Definition 6 (Mixed state)**:** A mixed state is represented by a density matrix $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where $p_i \geq 0$ and $\sum_i p_i = 1$. The density matrix satisfies three properties: it is Hermitian ($\rho = \rho^\dagger$), positive semidefinite (all eigenvalues non-negative), and has unit trace ($\mathrm{tr}(\rho) = 1$). For a pure state $|\psi\rangle$, the density matrix is simply $\rho = |\psi\rangle\langle\psi|$, the special case where the mixture collapses onto a single term.

The Born rule generalizes naturally to mixed states, but doing so requires introducing a slightly more general notion of "measurement" than the simple basis measurements we used above. Measurements are described most generally by a collection of **Kraus operators** $\{M_m\}$ satisfying $\sum_m M_m^\dagger M_m = I$; the associated **Positive Operator-Valued Measure (POVM)** is the set of operators $E_m := M_m^\dagger M_m$, and the probability of outcome m is $P(m) = \mathrm{tr}(E_m \rho) = \mathrm{tr}(M_m^\dagger M_m \rho)$. We will use the Kraus operators M_m themselves when we need to track the post-measurement state, as in Section 3, and the POVM elements E_m when we only care about outcome probabilities; the two notations describe exactly the same physical measurement and are related by $E_m = M_m^\dagger M_m$. For the special case of **projective measurements**, where $M_m = |m\rangle\langle m|$ for some orthonormal basis $\{|m\rangle\}$ (so that $E_m = M_m$ as well), this entire apparatus reduces to the familiar $P(m) = \langle m|\rho|m\rangle$ we would expect from the Born rule.

Example 6 (The quantum coin)**:** A biased coin can be represented two ways. As a *mixed* state,

$$\rho_{\mathrm{mix}} = \theta|0\rangle\langle 0| + (1-\theta)|1\rangle\langle 1| = \begin{pmatrix} \theta & 0 \\ 0 & 1-\theta \end{pmatrix},$$

or as a *pure* state $|\psi_\theta\rangle = \sqrt{\theta}|0\rangle + \sqrt{1-\theta}|1\rangle$, with density matrix

$$\rho_{\text{pure}} = |\psi_\theta\rangle\langle\psi_\theta| = \begin{pmatrix} \theta & \sqrt{\theta(1-\theta)} \\ \sqrt{\theta(1-\theta)} & 1-\theta \end{pmatrix}.$$

Both are valid density matrices (Hermitian, positive semidefinite, unit trace); what distinguishes them is the off-diagonal *coherence* term $\sqrt{\theta(1-\theta)}$, present only in ρ_{pure}. The mixed state represents *epistemic* uncertainty, classical ignorance about which of $|0\rangle, |1\rangle$ we are in, exactly like a classical probability mixture, producing no interference. The pure state represents *ontological* uncertainty: a genuine coherent superposition, not secretly one definite state, and it is this coherence that gives rise to interference effects.[1]

We use $|\psi_\theta\rangle$ as our standard quantum coin throughout the rest of this article. Applying the Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$:

$$H|\psi_\theta\rangle = \frac{\sqrt{\theta} + \sqrt{1-\theta}}{\sqrt{2}}|0\rangle + \frac{\sqrt{\theta} - \sqrt{1-\theta}}{\sqrt{2}}|1\rangle,$$

so $P(0) = \frac{1}{2}(\theta + (1-\theta) + 2\sqrt{\theta(1-\theta)}) = \frac{1}{2}(1 + 2\sqrt{\theta(1-\theta)})$ and $P(1) = \frac{1}{2}(1 - 2\sqrt{\theta(1-\theta)})$. These still sum to 1, but each now contains the cross-term $2\sqrt{\theta(1-\theta)}$: a genuine quantum interference effect, arising because H mixes the amplitudes of $|0\rangle$ and $|1\rangle$ *before* the Born rule squares them, with no analogue if we could only manipulate the classical probabilities $\theta, 1-\theta$ directly. The effect is visible only because we measure *after* applying H; measured immediately in the original basis, no such cross-term appears.

Summary: classical vs. quantum probability

Before moving on to conditional probability and Bayes' rule, it is worth pausing to consolidate everything we have built in this section into a single reference table, since we will return to and extend this dictionary repeatedly throughout the rest of the article.

The key takeaway: quantum mechanics is a **non-commutative probability theory**. Within any fixed measurement basis, the Born rule is additive exactly as Kolmogorov's axioms require; the new ingredient is that amplitudes belonging to a coherent superposition can be combined, via a unitary transformation, *before* the Born rule is applied, and the resulting cross-terms are the engine of every quantum speedup discussed below.

3. Conditional probability and Bayes' rule

Conditional probability is the mathematical machinery of updating beliefs in light of new information, and Bayes' rule is its most important consequence, arguably the single

[1]Classical mixture models (*e.g.* Gaussian mixtures) are the statistical analogue of the *mixed* state; the *pure* state has no classical statistical analogue. Measuring either ρ_{mix} or ρ_{pure} directly in the $\{|0\rangle, |1\rangle\}$ basis gives $P(0) = \theta, P(1) = 1-\theta$ with no interference, since $\{0\}$ and $\{1\}$ are orthogonal events; the coherence only becomes visible after a unitary mixes $|0\rangle$ and $|1\rangle$ before measurement, as below.

Table 1: Classical probability and its quantum analogue

Concept	**Classical**	**Quantum**
Space	Sample space Ω	Hilbert space $\mathcal{H}$
Basic object	Distribution $p(\omega)$	State vector $\|\psi\rangle$
Normalization	$\sum_\omega p(\omega) = 1$	$\|\|\psi\rangle\|^2 = 1$
Combination rule	$p(\omega) \geq 0$, add	Add within a basis; may cancel across non-orthogonal states
Measurement outcome	$\omega \sim p(\omega)$	$i \sim \|\psi_i\|^2$
Update after measurement	Condition on observed ω	Collapse to $\|i\rangle$
Uncertainty about state	Mixture of distributions	Density matrix ρ

most useful formula in all of applied statistics. We state both here in their familiar classical form before turning to their quantum counterparts.

Definition 7 (Conditional probability)**:** For $P(B) > 0$, $P(A \mid B) = P(A \cap B)/P(B)$ (Casella and Berger, 2002).

Theorem 1 (Bayes' rule)**:**

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{\sum_i P(B \mid A_i)P(A_i)}.$$

Theorem 2 (Law of total probability)**:** For a partition $\{A_i\}$, $P(B) = \sum_i P(B \mid A_i)P(A_i)$.

Example 7 (Coin inference)**:** Suppose our prior places equal weight on two candidate biases: $P(\theta = 0.5) = 0.5$ and $P(\theta = 0.7) = 0.5$. After observing $k = 7$ heads in $n = 10$ flips, Bayes' rule gives the posterior weight on $\theta = 0.7$ as proportional to the likelihood times the prior: $P(K{=}7 \mid \theta{=}0.7) \cdot 0.5 = \binom{10}{7}(0.7)^7(0.3)^3 \cdot 0.5 \approx 0.2668 \times 0.5 = 0.1334$, while the competing hypothesis contributes $P(K{=}7 \mid \theta{=}0.5) \cdot 0.5 = \binom{10}{7}(0.5)^{10} \cdot 0.5 \approx 0.1172 \times 0.5 = 0.0586$. Normalising, the posterior probability that $\theta = 0.7$ is $0.1334/(0.1334 + 0.0586) \approx 0.695$ (Gelman *et al.*, 2013), a substantial update from the prior 50% toward the hypothesis better supported by the data, exactly as intuition would suggest.

Quantum conditional probability

In quantum mechanics, conditioning on a measurement outcome updates the state in a manner directly analogous to classical conditioning, using the Kraus-operator notation M_m we introduced in Section 2.

Definition 8 (Quantum conditional state)**:** For Kraus operators $\{M_m\}$ with $\sum_m M_m^\dagger M_m = I$, if outcome m occurs with probability $p_m = \mathrm{tr}(M_m^\dagger M_m \rho)$, the post-measurement state is $\rho_m = M_m \rho M_m^\dagger / p_m$. For projective measurements ($M_m = |m\rangle\langle m|$), this is $\rho_m = |m\rangle\langle m|$, collapse onto the measured basis vector.

A natural question is whether there is a quantum analogue of Bayes' rule updating a prior *state* into a posterior *state* given quantum "data." Unlike the classical case, there is no single canonical quantum Bayes' rule; several inequivalent formalizations exist depending on what plays the role of the likelihood (Wiebe *et al.*, 2012). We present one widely used construction, flagged explicitly as one choice among several.

Definition 9 (A quantum Bayes' update)**:** Given prior ρ and likelihood (Kraus) operators $\{L_x\}$ with $\sum_x L_x^\dagger L_x = I$, one natural posterior, obtained by applying the corresponding quantum instrument and renormalising, is

$$\rho_{\text{post}} = \frac{\sum_x L_x \rho L_x^\dagger}{\text{tr}(\sum_x L_x \rho L_x^\dagger)}.$$

This aggregates over all possible outcomes x rather than conditioning on one observed x; conditioning on a single realised x_0 instead uses the single-Kraus-operator rule $\rho_{x_0} = L_{x_0}\rho L_{x_0}^\dagger/\text{tr}(L_{x_0}\rho L_{x_0}^\dagger)$, a special case of the quantum conditional state above. In the classical-like (diagonal, commuting) case, the amplitude version of this update reduces, heuristically, to $\sqrt{p(\theta \mid y)} \propto \sqrt{p(y \mid \theta)}\sqrt{p(\theta)}$, a useful mnemonic, not a theorem, since making it precise requires specifying exactly how θ and y are encoded as quantum registers. For a detailed treatment of the several competing notions, see Wiebe *et al.* (2012).

Theorem 3 (Quantum law of total probability)**:** For a POVM $\{E_m\}$ and $\rho = \sum_i p_i \rho_i$, $P(m) = \text{tr}(E_m \rho) = \sum_i p_i \text{tr}(E_m \rho_i)$.

4. Random variables and probability distributions

Now that we have established the foundations of classical probability and their quantum analogues, we turn to one of the most important concepts in statistics: the random variable. A random variable is a way of turning the abstract outcomes of an experiment into numbers that we can analyse, instead of talking about "Heads" and "Tails," we assign the number 1 to Heads and 0 to Tails, which lets us compute averages, variances, and other summaries. In quantum mechanics, the analogue is an **observable**: just as a random variable assigns a number to each outcome in the sample space, an observable assigns a number, an eigenvalue, to each possible measurement outcome.

Classical random variables

Intuitively, a random variable is simply a numerical summary of an experiment. Before we flip a coin, we do not know whether it will land Heads or Tails, but we know in advance that if it is Heads we will record 1, and if it is Tails we will record 0; the random variable is precisely this rule for converting outcomes into numbers.

Definition 10 (Random variable)**:** A random variable $X : \Omega \to \mathbb{R}$ is a measurable function from the sample space Ω to the real numbers. The cumulative distribution function (CDF) of X is $F_X(x) = P(X \leq x) = P(\{\omega \in \Omega : X(\omega) \leq x\})$.

The term "measurable" here is a technical condition ensuring that the set $\{\omega : X(\omega) \leq$

$x\}$ is itself an event in $\mathcal{F}$, so that we are actually allowed to assign it a probability. For all practical purposes in statistics, every function one would ever write down is measurable; the condition only becomes relevant for the same pathological, non-measurable sets that motivated the σ-algebra machinery in Section 2, and essentially never arises in applied work.

Example 8 (Coin flip indicator)**:** Define $X(\text{Heads}) = 1$ and $X(\text{Tails}) = 0$. Then $F_X(x) = 0$ for $x < 0$; $F_X(x) = 1 - \theta$ for $0 \leq x < 1$; and $F_X(x) = 1$ for $x \geq 1$.

Definition 11 (PMF and PDF)**:** For discrete X, the PMF is $p_X(x) = P(X = x) \geq 0, \sum_x p_X(x) = 1$. For continuous X, the PDF f_X satisfies $P(a \leq X \leq b) = \int_a^b f_X(x)\,dx$, $f_X(x) \geq 0$, $\int_{-\infty}^{\infty} f_X(x)\,dx = 1$ (Casella and Berger, 2002).

Definition 12 (Binomial distribution)**:** For independent Bernoulli(θ) trials $X_1, \ldots, X_n$, $K = \sum_i X_i$ has $P(K = k) = \binom{n}{k}\theta^k(1-\theta)^{n-k}$, $k = 0, \ldots, n$. This is the natural distribution governing exactly our running example: the total number of heads in n independent flips of a coin with bias θ.

Example 9 (Coin flips)**:** For $n = 10, \theta = 0.7$: $P(K = 7) = \binom{10}{7}(0.7)^7(0.3)^3 = 120 \times 0.0823543 \times 0.027 \approx 0.267$, the same likelihood we used numerically in the Bayes' rule example of Section 3.

Quantum observables

Now we come to the quantum analogue. Just as a classical random variable assigns a real number to each outcome in the sample space, a quantum observable assigns a real number, an eigenvalue, to each possible measurement outcome. Recall from linear algebra that a Hermitian matrix $A = A^\dagger$, where $\dagger$ denotes conjugate transpose, has three important properties: all of its eigenvalues are real; eigenvectors corresponding to distinct eigenvalues are automatically orthogonal; and the matrix can always be diagonalised by a unitary transformation. The spectral theorem generalises these familiar linear-algebra facts to operators on general Hilbert spaces (Nielsen and Chuang, 2010).

Definition 13 (Observable)**:** A Hermitian operator $O = O^\dagger$ on a Hilbert space $\mathcal{H}$ is called an **observable**. Its spectral decomposition is $O = \sum_o o|o\rangle\langle o|$, where $\{o\}$ are the distinct eigenvalues, the possible measurement outcomes, and $\{|o\rangle\}$ form an orthonormal basis of eigenvectors. This is the precise quantum analogue of writing a classical random variable as $X(\omega) = \sum_x x \cdot \mathbf{1}_{\{X=x\}}(\omega)$, where $\mathbf{1}_A$ is the indicator function of event A: both expressions decompose a numerical summary into a weighted sum over its possible values, weighted by an indicator (classically) or a projector (quantumly) onto the event that summary takes that value.

When we measure the observable O on a quantum state ρ, the Born rule tells us the probability of obtaining eigenvalue o is $P(O = o) = \text{tr}(|o\rangle\langle o|\rho) = \langle o|\rho|o\rangle$; for a pure state $|\psi\rangle$, this simplifies to the familiar $P(O = o) = |\langle o|\psi\rangle|^2$.

Table 2: Random variables and their quantum analogue, observables

Classical	Quantum
Sample space Ω	Hilbert space $\mathcal{H}$
Outcome $\omega \in \Omega$	Basis state $\|i\rangle \in \mathcal{H}$
Random variable $X : \Omega \to \mathbb{R}$	Observable $O = \sum_o o\|o\rangle\langle o\|$
$P(X = x) = \sum_{\omega:X(\omega)=x} P(\omega)$	$P(O = o) = \mathrm{tr}(\|o\rangle\langle o\|\rho)$
$\mathbb{E}[X] = \sum_x xP(X = x)$	$\langle O\rangle = \sum_o oP(O = o) = \mathrm{tr}(O\rho)$

Quantum sample states

One of the most powerful ideas in quantum computing for statistics is the **quantum sample state**: a single quantum state that encodes an entire probability distribution in its amplitudes, rather than in a classical array of numbers.

Definition 14 (Quantum sample state)**:** For a distribution $p(x)$ on a finite set $\mathcal{X} = \{x_1, \ldots, x_N\}$, the quantum sample state is $|P\rangle = \sum_{x\in\mathcal{X}} \sqrt{p(x)}|x\rangle$, where $\{|x\rangle\}$ is an orthonormal basis labeled by the elements of $\mathcal{X}$.

Why is this called a "sample state"? Because measuring $|P\rangle$ in the $\{|x\rangle\}$ basis gives outcome x with probability $|\langle x|P\rangle|^2 = |\sqrt{p(x)}|^2 = p(x)$, exactly the original distribution. The state therefore functions as a "factory" for producing i.i.d. samples from $p(x)$: each measurement draws one sample, and repeated measurements (on fresh copies of the state, since measurement is destructive) draw i.i.d. samples exactly as classical Monte Carlo would.

Exponential efficiency in storage. To store a distribution over $N = 2^n$ values classically requires N numbers, about 8 MB for $N = 10^6$ double-precision floats. As a quantum sample state, only $n = \log_2 N$ qubits are needed: 20 qubits for $N = 1{,}000{,}000$, an exponential reduction. But this is a storage statement, not a preparation-time statement: writing an arbitrary N-dimensional distribution into amplitudes generically requires $O(N)$ gates, so the storage saving does not automatically yield a preparation-time saving unless the distribution has special structure. Grover and Rudolph (2002) showed distributions with an efficiently integrable CDF (*e.g.* the normal, or our binomial example below) can be prepared in $O(\mathrm{polylog}\, N)$ operations; but Herbert (2021) showed this construction is essentially limited to distributions simple enough to not need Monte Carlo in the first place, raising real doubt about whether efficient preparation is available for the complex empirical distributions statisticians actually meet. Reading out an already-prepared state (tomography) is a separate cost, discussed in Section 10.

Definition 15 (Amplitude encoding)**:** For $x = (x_0, \ldots, x_{N-1})^\top \in \mathbb{C}^N$ with $\|x\| = 1$, amplitude encoding represents it as $|x\rangle = \sum_i x_i|i\rangle$. Unlike a quantum sample state, the amplitudes here are arbitrary normalized complex numbers, not square roots of a target distribution; amplitude encoding underlies linear-algebra algorithms such as HHL (Harrow *et al.*, 2009), and loading an arbitrary vector this way costs $O(N)$ operations in general, the same generic cost as above.

Example 10 (Binomial quantum state)**:** $|\mathrm{Bin}_{n,\theta}\rangle = \sum_{k=0}^{n} \sqrt{\binom{n}{k}\theta^k(1-\theta)^{n-k}}|k\rangle$ needs only $\lceil \log_2(n+1) \rceil$ qubits to *store*, 7 qubits for $n = 100$, versus 101 classical probabilities. (Because the binomial amplitudes follow an efficiently integrable CDF, Grover–Rudolph can in fact prepare this particular state efficiently; a generic distribution over the same 101 outcomes would not enjoy this guarantee.) Given many copies of $|\mathrm{Bin}_{n,\theta}\rangle$ for unknown θ, estimating θ is the quantum version of our coin problem, the setting for Quantum Amplitude Estimation, Section 6 (Brassard *et al.*, 2002; Montanaro, 2015).

5. Expectation and variance

Now that we understand random variables (classically) and observables (quantum mechanically), we turn to two of the most important summaries of any distribution: the expectation, which tells us the average value we should expect, and the variance, which tells us how spread out the distribution is around that average.

Definition 16 (Expectation)**:** For a discrete random variable X with probability mass function $p(x)$, $\mathbb{E}[X] = \sum_x x\, p(x)$. For a continuous random variable X with probability density function $f(x)$, $\mathbb{E}[X] = \int_{-\infty}^{\infty} x f(x)\, dx$ (Casella and Berger, 2002). The expectation is linear: for any constants a, b and random variables X, Y, $\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$.

Definition 17 (Variance)**:** The variance of a random variable X is $\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$; the standard deviation is $\sigma = \sqrt{\mathrm{Var}(X)}$.

Example 11 (Coin flip)**:** For a single coin flip, define $X = 1$ for Heads and $X = 0$ for Tails. Then $\mathbb{E}[X] = 1 \cdot \theta + 0 \cdot (1-\theta) = \theta$ and $\mathbb{E}[X^2] = 1^2 \cdot \theta + 0^2 \cdot (1-\theta) = \theta$, so $\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \theta - \theta^2 = \theta(1-\theta)$. For a fair coin ($\theta = 0.5$), the variance is 0.25; for a completely biased coin ($\theta = 0$ or $\theta = 1$), the variance is 0, exactly as we would expect since there is then no uncertainty left in the outcome at all.

The quantum expectation generalizes the classical expectation to observables in a direct and natural way.

Definition 18 (Quantum expectation and variance)**:** For a quantum state ρ (density matrix) and an observable $O = \sum_o o|o\rangle\langle o|$, the expectation value is $\langle O \rangle = \mathrm{tr}(O\rho) = \sum_o o\, P(O{=}o)$, where $P(O = o) = \mathrm{tr}(|o\rangle\langle o|\rho)$ is the probability of obtaining outcome o. For a pure state $|\psi\rangle$, we have $\rho = |\psi\rangle\langle\psi|$ and $\mathrm{tr}(O|\psi\rangle\langle\psi|) = \langle\psi|O|\psi\rangle = \langle\psi|O|\psi\rangle$, so the two notations for the expectation value used throughout this article, trace form and bra-ket form, coincide exactly whenever ρ is pure, and we use whichever is more convenient in context. The quantum expectation satisfies the same linearity property as the classical expectation: $\langle aO_1 + bO_2 \rangle = a\langle O_1 \rangle + b\langle O_2 \rangle$. The variance of an observable O in state ρ is $\mathrm{Var}(O) = \langle O^2 \rangle - \langle O \rangle^2 = \mathrm{tr}(O^2\rho) - (\mathrm{tr}(O\rho))^2$.

Example 12 (Quantum coin expectation)**:** For the quantum coin state $|\psi_\theta\rangle = \sqrt{\theta}|0\rangle + \sqrt{1-\theta}|1\rangle$ and the observable $Z = |0\rangle\langle 0| - |1\rangle\langle 1| = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ (the Pauli Z matrix), we have $\langle Z \rangle = \langle\psi_\theta|Z|\psi_\theta\rangle = |\sqrt{\theta}|^2 \cdot 1 + |\sqrt{1-\theta}|^2 \cdot (-1) = \theta - (1-\theta) = 2\theta - 1$, a linear transform of θ. More directly, if we define the observable $O = |0\rangle\langle 0|$, which measures simply whether the outcome is Heads, then $\langle O \rangle = \langle\psi_\theta|O|\psi_\theta\rangle = \theta$, matching the classical expectation exactly, as

it should: the quantum and classical machinery must agree when restricted to this simple case.

Estimating expectations is perhaps the single most fundamental computational task in statistics. Almost everything we do as practitioners, computing sample means, running regression, performing hypothesis tests, evaluating risk in finance, calculating treatment effects in medicine, reduces, at bottom, to estimating an expectation. Classically, the workhorse method for this task is **Monte Carlo** (Metropolis and Ulam, 1949): draw N i.i.d. samples from a distribution and average the function of interest. The standard error of this estimate scales as $O(1/\sqrt{N})$, which means halving the error requires four times as many samples, a frustratingly slow rate of improvement that statisticians have lived with for decades. **Quantum Amplitude Estimation** (Brassard *et al.*, 2002; Montanaro, 2015) achieves a quadratic speedup *in the number of oracle queries* over this classical rate, under the assumption of coherent access to a state-preparation oracle (we make the precise caveats around this assumption explicit in Section 10). To understand how QAE achieves this, we first need to understand Grover's search algorithm and its generalisation, Quantum Amplitude Amplification.

6. Quantum search and amplitude estimation

Before we can understand the quadratic speedup for estimating expectations promised at the end of the last section, we need to understand Grover's search algorithm, since it is the conceptual foundation that both Quantum Amplitude Amplification and Quantum Amplitude Estimation are built on top of. We will work through Grover's algorithm in full geometric detail, then show how it generalises, and finally apply the generalised version back to our running coin-bias problem.

The problem: unstructured search

Imagine you have a database with N items. One of these items is marked, it satisfies some condition, and you want to find it. You have a "black box" oracle that can tell you whether a given item is marked, but the oracle gives you no other information; this is called **unstructured search**. Classically, the best you can do is check items one by one. In the worst case you might check all N items before finding the marked one; on average you will check about $N/2$. So the classical query complexity is $O(N)$. Grover's algorithm (Grover, 1996) solves this problem using only $O(\sqrt{N})$ queries, a quadratic speedup, and one that has been proven optimal: no quantum algorithm can do better for fully unstructured search.

Grover's algorithm, step by step

- **Step 1: Represent the database as a quantum state.** For $N = 2^n$, label each item by an n-bit string $x \in \{0,1\}^n$. We represent the entire database as a quantum superposition $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, the **uniform superposition**, in which every basis state $|x\rangle$ carries amplitude $1/\sqrt{N}$. Intuitively, this single quantum state "contains" every item in the database simultaneously, with no item yet favoured over any other.

- **Step 2: The oracle.** We are given access to an oracle U_f that recognises the marked item x_0 and acts as $U_f|x\rangle = -|x\rangle$ if $x = x_0$, and $U_f|x\rangle = |x\rangle$ otherwise; equivalently $U_f = I - 2|x_0\rangle\langle x_0|$. This operator flips the sign of the amplitude on the marked state and leaves every other amplitude untouched, a purely local action that, on its own, does nothing observable, since the Born rule only sees squared magnitudes.
- **Step 3: Reflection about the mean.** Grover's algorithm pairs the oracle with a second operator, the **diffusion operator** $U_s = 2|s\rangle\langle s| - I$ for $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$. To see what this does, consider any state $|\phi\rangle = \sum_x \alpha_x |x\rangle$ with mean amplitude $\bar{\alpha} = \frac{1}{N} \sum_x \alpha_x$; applying U_s sends $|\phi\rangle$ to $\sum_x (2\bar{\alpha} - \alpha_x)|x\rangle$, so amplitudes above the mean shrink and amplitudes below the mean grow, literally a reflection of each amplitude about the average amplitude.
- **Step 4: The Grover iteration.** Combining these two operators, $G = U_s U_f$, turns out to act as a rotation in the two-dimensional plane spanned by the marked state $|x_0\rangle$ and the uniform superposition of all the unmarked states, $|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle$. This geometric picture, a single rotation in a two-dimensional subspace, repeated many times, is the key to understanding why the algorithm works at all.
- **Step 5: Geometric interpretation.** Writing $|s\rangle = \sin\theta_G |x_0\rangle + \cos\theta_G |s'\rangle$ with $\sin\theta_G = 1/\sqrt{N}$ (we use θ_G, not θ, to avoid clashing with the coin bias used throughout this article), each application of G rotates by a further angle $2\theta_G$: $G|s\rangle = \sin(3\theta_G)|x_0\rangle + \cos(3\theta_G)|s'\rangle$, and in general after k iterations, $G^k|s\rangle = \sin((2k+1)\theta_G)|x_0\rangle + \cos((2k+1)\theta_G)|s'\rangle$. Each iteration moves the state a little closer to $|x_0\rangle$, but, crucially, if we keep iterating past the optimal point, the rotation overshoots and the state begins moving *away* from $|x_0\rangle$ again.
- **Step 6: Measurement.** Measuring after k iterations succeeds with probability $P(\text{success}) = \sin^2((2k+1)\theta_G)$, which is maximised exactly when $(2k+1)\theta_G$ is as close as possible to $\pi/2$. The exact optimal iteration count is therefore $k^\star = \lfloor \pi/(4\theta_G) \rfloor = \lfloor \pi/(4\arcsin(1/\sqrt{N})) \rfloor$. Since $\arcsin(1/\sqrt{N}) = 1/\sqrt{N} + O(N^{-3/2})$ for large N, this gives the familiar asymptotic estimate $k^\star \approx (\pi/4)\sqrt{N}$, an approximation, not an exact equality, since $\theta_G \neq 1/\sqrt{N}$ exactly for any finite N. This distinction matters in practice: stopping a few iterations early or late, especially for small N, can noticeably hurt the success probability.

Example 13 (Searching coin biases)**:** Suppose we discretise the coin bias into $N = 1{,}000{,}000$ candidate values and an oracle can recognise the "correct" one. Classical search checks up to 10^6 possibilities in the worst case. Grover's algorithm needs only $\sqrt{10^6} = 1000$ queries asymptotically; more precisely, the exact optimal count is $k^\star = \lfloor \pi/(4\arcsin(10^{-3})) \rfloor \approx 785$, already matching the asymptotic order $(\pi/4)\sqrt{N} \approx 785$ closely even at this only moderately large N.

Quantum Amplitude Amplification (QAA)

Grover's algorithm is the special case of QAA (Brassard *et al.*, 2002) where the initial state is the uniform superposition; in general we start from $|\psi\rangle = \mathcal{A}|0\rangle = a_1|\psi_1\rangle + a_0|\psi_0\rangle$ prepared by a unitary $\mathcal{A}$, with $|\psi_1\rangle$ the "good" subspace, $a_1 = \langle\psi_1|\psi\rangle$ its amplitude, and $|a_1|^2$ the probability of a good outcome.

Definition 19 (QAA)**:** Since $\mathcal{A}$ is unitary, $\mathcal{A}^{-1} = \mathcal{A}^\dagger$ (both notations appear in the literature; we use $\mathcal{A}^{-1}$ to emphasise that it "undoes" state preparation). Define $Q = -\mathcal{A}U_0\mathcal{A}^{-1}U_f$, where U_f flips the sign of the good subspace and U_0 flips the sign of $|0\rangle$. After $t = O(1/|a_1|)$ applications of Q, the good-subspace amplitude becomes $O(1)$ (Brassard *et al.*, 2002).

The structure of QAA mirrors Grover step for step, which is worth making explicit since it is what lets the same geometric intuition carry over: $\mathcal{A}$ prepares the initial state, replacing the Hadamard layer $H^{\otimes n}$; U_f marks good states, replacing the oracle; and $\mathcal{A}U_0\mathcal{A}^{-1} = \mathcal{A}U_0\mathcal{A}^\dagger$ reflects about the initial state, replacing reflection about the mean. Everything we learned about Grover's rotation in a two-dimensional subspace applies again here, just with the uniform superposition $|s\rangle$ replaced by the more general prepared state $|\psi\rangle$.

Definition 20 (Quantum Amplitude Estimation)**:** Q has eigenvalues $e^{\pm 2i\phi}$ on the 2D subspace spanned by $|\psi_1\rangle, |\psi_0\rangle$, where $\sin^2\phi = |a_1|^2$. Quantum Phase Estimation (Section 8) on Q estimates ϕ to precision $O(\epsilon)$ using $O(1/\epsilon)$ queries; plugging $\hat{\phi}$ into $\widehat{|a_1|^2} = \sin^2\hat{\phi}$ then estimates $|a_1|^2$ to precision $O(\epsilon)$ as well, *provided* $|a_1|^2$ is bounded away from 0 and 1.[2]

We now return to the general problem of estimating an expectation $\mu = \mathbb{E}[f(X)] = \sum_x p(x)f(x)$ for some function f with $0 \le f(x) \le 1$, and state the classical and quantum bounds precisely before working through the coin example step by step.

Theorem 4 (Classical Monte Carlo error bound)**:** Let $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} f(X_i)$ be the sample mean from N i.i.d. draws of $X \sim p(x)$. Then $\mathbb{E}[\hat{\mu}] = \mu$ and $\mathrm{Var}(\hat{\mu}) = \sigma^2/N$ where $\sigma^2 = \mathrm{Var}(f(X))$, so $N = O(\sigma^2/\epsilon^2)$ draws suffice to achieve error ϵ with constant probability (Metropolis and Ulam, 1949).

Theorem 5 (Quantum Monte Carlo via QAE (Brassard *et al.*, 2002; Montanaro, 2015))**:** QAE estimates μ to precision ϵ with high probability using $O(\sigma/\epsilon)$ queries to f and the state-preparation oracle, away from the boundary regime $\mu \approx 0$ or $\mu \approx 1$, a **quadratic speedup** over classical Monte Carlo in query complexity.

Step-by-step QAE for the coin example

To estimate θ itself, rather than the sample proportion of some fixed batch of flips, the oracle below must encode the Bernoulli probability θ directly, rather than n individually realised 0/1 outcomes. This distinction is easy to gloss over but is actually the crux of why QAE behaves so differently from classical Monte Carlo, so we spell out the construction carefully.

- **Step 1: State preparation.** We build the state $\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle_n|0\rangle_m|0\rangle_1$. The first register, of n qubits, is an *index* register, it is not recording flip outcomes the way a

[2]Near these endpoints, $\sin^2$ is locally flat (derivative vanishes), so the same angular precision ϵ translates into a smaller error in $|a_1|^2$; near $\phi = \pi/4$ the map is best-conditioned. The headline "$O(1/\epsilon)$ queries for precision ϵ" should be read as holding uniformly away from the endpoints; see Brassard *et al.* (2002) for the precise non-asymptotic bound and Montanaro (2015) for the resulting Monte Carlo speedup.

classical batch of n flips would, but is instead used to drive N independent coherent "virtual flips" of the same underlying coin. The second register will encode θ; the third is a single auxiliary qubit that will eventually carry the information we want to measure.

- **Step 2: Function encoding.** An oracle $\mathcal{A}_\theta$ prepares, completely independently of the index i, the same encoding of the bias θ in the second register: $\mathcal{A}_\theta|i\rangle_n|0\rangle_m = |i\rangle_n|\theta\rangle_m$. Every branch of the superposition therefore shares the *same* parameter θ, rather than each branch carrying its own independently realised 0/1 outcome as a classical batch of flips would. This is the key conceptual difference from classical sampling, and it is exactly what lets QAE target the parameter θ itself rather than merely a noisy sample proportion k/n.
- **Step 3: Controlled rotation.** A controlled rotation $C\text{-}R_y(2\arcsin\sqrt{\theta})$ acts on the auxiliary qubit: $|\theta\rangle_m|0\rangle_1 \mapsto |\theta\rangle_m(\sqrt{1-\theta}|0\rangle + \sqrt{\theta}|1\rangle)$, which exactly reproduces the pure quantum coin state $|\psi_\theta\rangle$ from Section 2 inside the auxiliary qubit.
- **Step 4: Uncomputation.** We apply $\mathcal{A}_\theta^{-1}$ to clear the second register back to $|0\rangle_m$, so that it no longer carries any entanglement with the rest of the system.
- **Step 5: Resulting state.** The state is now $|\psi\rangle = \left(\frac{1}{\sqrt{N}}\sum_i |i\rangle_n\right) \otimes (\sqrt{1-\theta}|0\rangle + \sqrt{\theta}|1\rangle)$, a product state, with the index register and the auxiliary qubit completely unentangled. By the Born rule applied to this unentangled auxiliary factor, $P(\text{aux}{=}1) = \theta$ exactly, with no dependence whatsoever on N or on any single realised classical sample. The role of N here is entirely different from its classical role: it only controls how many times the same rotation is coherently repeated across the index register within a single QAE run, which is precisely the resource that QAE's underlying amplitude amplification machinery uses to sharpen the eventual estimate.
- **Step 6: Amplitude estimation.** Applying QAE to this construction estimates $|a_1|^2 = \theta$ to additive precision ϵ using $O(1/\epsilon)$ queries, subject to the boundary caveat discussed above when θ is close to 0 or 1.

Example 14 (Numerical comparison)**:** For $\epsilon = 0.001$, away from the boundary: classical Monte Carlo needs $\approx 1/\epsilon^2 = 10^6$ flips; QAE needs $\approx 1/\epsilon = 1000$ oracle queries, requiring coherent execution, but provably a quadratic speedup in query complexity. The gap widens dramatically as the target precision tightens, which is worth seeing concretely rather than only asymptotically:

Table 3: The query-complexity gap between classical Monte Carlo and QAE grows linearly in $1/\epsilon$; for high-precision estimation the practical difference becomes enormous, though, as emphasised throughout, this gap is in oracle queries, not necessarily wall-clock time on real hardware

Target precision ϵ	Classical samples $O(1/\epsilon^2)$	QAE queries $O(1/\epsilon)$	Ratio
10^{-1}	100	10	10×
10^{-2}	10,000	100	100×
10^{-3}	1,000,000	1,000	1,000×
10^{-4}	100,000,000	10,000	10,000×

7. Central limit theorem, Markov chains, and quantum walks

Definition 21 (Sample mean)**:** For i.i.d. $X_1, \ldots, X_n$ with mean μ, variance σ^2: $\bar{X}_n = \frac{1}{n}\sum_i X_i$, $\mathbb{E}[\bar{X}_n] = \mu$, $\mathrm{Var}(\bar{X}_n) = \sigma^2/n$ (Casella and Berger, 2002).

As n grows, $\mathrm{Var}(\bar{X}_n) \to 0$ and the distribution becomes approximately normal regardless of the underlying distribution, the Central Limit Theorem.

Theorem 6 (CLT)**:** $\sqrt{n}(\bar{X}_n - \mu)/\sigma \xrightarrow{d} N(0,1)$, equivalently $\sqrt{n}(\bar{X}_n - \mu) \xrightarrow{d} N(0, \sigma^2)$.

Example 15 (Coin flips and CLT)**:** $\mu = \theta, \sigma^2 = \theta(1-\theta)$: $\sqrt{n}\left(\frac{k}{n} - \theta\right) \approx N(0, \theta(1-\theta))$, giving the 95% CI $\hat{\theta} \pm 1.96\sqrt{\hat{\theta}(1-\hat{\theta})/n}$, where $\hat{\theta} = k/n$ and 1.96 is the 97.5% standard-normal quantile.

Now that we understand expectation and variance, we turn to a deeper question: if we take a sample and compute a statistic, like the sample mean, how does that statistic behave across different samples? This is the study of **sampling distributions**, and it is where the Central Limit Theorem enters.

When we measure identical quantum sample states $|P\rangle = \sum_x \sqrt{p(x)}|x\rangle$ (Section 4) repeatedly, we obtain i.i.d. samples from $p(x)$, and the classical CLT applies directly to these measurement outcomes: quantum sampling is statistically indistinguishable from classical sampling *once the state is prepared*. This is worth dwelling on for a moment, since it is easy to assume quantum mechanics changes everything, it does not change the statistics of measurement outcomes once you have them. The quantum advantage instead lies entirely in *how efficiently we can prepare* $|P\rangle$ in the first place, particularly when p is the stationary distribution of a Markov chain. This is the role of quantum walks, to which we now turn. We first review the classical theory of Markov chains, since the quantum walk construction is built directly on top of it.

Classical Markov chains

A Markov chain is a random process where the next state depends only on the current state, the so-called "Markov property", and not on the full history of how the process arrived there. This memorylessness is what makes Markov chains tractable to analyse and is the foundation on which MCMC methods are built.

Definition 22 (Markov chain)**:** A (discrete-time, finite-state) Markov chain on a state space $\mathcal{X}$ of size N is defined by a transition matrix $P \in \mathbb{R}^{N \times N}$ where $P(x,y) = P(X_{t+1}{=}y \mid X_t{=}x)$, satisfying $P(x,y) \geq 0$ for all x, y and $\sum_y P(x,y) = 1$ for each x (each row sums to 1, since the chain must go *somewhere*).

A Markov chain is called **reversible** with respect to a distribution π if it satisfies the *detailed balance* condition $\pi(x)P(x,y) = \pi(y)P(y,x)$ for all x, y; in this case π is automatically a **stationary distribution**, satisfying $\pi^\top P = \pi^\top$, meaning that once the chain reaches π it stays there in distribution forever after. The **mixing time** of a Markov chain is the

number of steps required for the distribution of X_t to get close to π, regardless of the chain's starting point; this mixing time is governed by the **spectral gap** $\delta = 1 - \lambda_2$, where λ_2 is the second-largest eigenvalue of P in absolute value (the largest eigenvalue is always exactly 1, corresponding to the stationary distribution itself). The classical mixing time scales as $O(1/\delta)$: chains with a large spectral gap mix quickly, while chains with a small spectral gap, meaning the second eigenvalue is close to 1, can take a very long time to forget their starting point.

Example 16 (Two-state chain)**:** Consider a two-state chain on $\{0, 1\}$ with transition matrix $P = \left(\begin{smallmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{smallmatrix}\right)$. Direct computation shows the eigenvalues of P are 1 and $1 - \alpha - \beta$, so the spectral gap is $\delta = 1 - (1 - \alpha - \beta) = \alpha + \beta$. Solving $\pi^\top P = \pi^\top$ gives the stationary distribution $\pi = \left(\frac{\beta}{\alpha+\beta}, \frac{\alpha}{\alpha+\beta}\right)$, and the mixing time is $O(1/(\alpha + \beta))$: the chain mixes quickly when either transition probability is large, and slowly when both α and β are small, since the chain then has little tendency to leave whichever state it starts in.

Szegedy quantum walks

In 2004, Mario Szegedy showed how to quantise any reversible Markov chain (Szegedy, 2004). On the doubled register $\mathcal{H} = \mathbb{C}^{\mathcal{X}} \otimes \mathbb{C}^{\mathcal{X}}$, define $U_P|x\rangle|0\rangle = |x\rangle \sum_y \sqrt{P(x, y)}|y\rangle$ (extended to a full unitary in the standard way). Let $\Pi = \sum_x |x\rangle\langle x| \otimes U_P|0\rangle\langle 0|U_P^\dagger$ be the projector onto the span of $\{U_P|x\rangle|0\rangle\}_x$, S the swap operator, and $\Pi' = S\Pi S$. The reflections are $\mathcal{R}_1 = 2\Pi - I$, $\mathcal{R}_2 = 2\Pi' - I$, and the walk operator is $W(P) = \mathcal{R}_2\mathcal{R}_1$.[3]

Theorem 7 (Quantum walk speedup (Szegedy, 2004))**:** The state $|\pi\rangle = \sum_x \sqrt{\pi(x)}|x\rangle \otimes \left(\sum_y \sqrt{P(x, y)}|y\rangle\right)$ is a $+1$ eigenstate of $W(P)$; measuring its first register recovers exactly the quantum sample state $\sum_x \sqrt{\pi(x)}|x\rangle$ of Section 4. If the classical chain has spectral gap δ, the eigenvalues of $W(P)$ are $e^{\pm i\Theta}$ with *phase gap* $\Theta = \Omega(\sqrt{\delta})$, quadratically larger than δ. Quantum phase estimation on $W(P)$ can prepare a state ϵ-close to $|\pi\rangle$ using $O(1/\sqrt{\delta})$ applications of $W(P)$, versus $O(1/\delta)$ classical steps.

Why does the quantum walk achieve a quadratic speedup at all? The classical Markov chain moves by random, memoryless steps, and each step reduces the distance to stationarity by a factor governed by δ; convergence is a diffusive, gradual process. The quantum walk instead uses interference to explore the state space coherently. The walk operator rotates the state in a two-dimensional invariant subspace, and the relevant rotation angle scales as $\Theta = \Omega(\sqrt{\delta})$ rather than δ itself, so the number of applications of $W(P)$ needed to complete an order-one rotation, and hence to localise on $|\pi\rangle$ via phase estimation, is $O(1/\sqrt{\delta})$ rather than $O(1/\delta)$. This is exactly the same square-root relationship between rotation angle and iteration count that drove the Grover speedup in Section 6; the Szegedy walk is, in this sense, Grover's algorithm generalised from searching a fixed marked item to sampling from an entire stationary distribution.

[3]We use the projector notation Π, Π' here, rather than $U_P U_P^\dagger$ directly, because $U_P U_P^\dagger = I$ for a full unitary completion and would trivialise the reflection; what matters is the projector onto the specific subspace prepared by U_P conditioned on the first register.

Example 17 (Cycle)**:** A classical random walk on an N-cycle has *hitting time* $O(N^2)$ (diffusive); a continuous-time quantum walk reaches the antipodal node in $O(N)$ steps, *ballistic* propagation (Childs *et al.*, 2003; Kempe, 2003).[4]

Quantum Markov chain Monte Carlo

Classical MCMC is among the most widely used computational tools in all of Bayesian statistics: we construct a Markov chain whose stationary distribution is our target distribution π, typically a posterior distribution that we cannot sample from directly, and then simulate the chain for many steps to obtain approximate samples from π. Quantum MCMC (QMCMC) aims to skip the simulation entirely and prepare the quantum sample state $|\pi\rangle = \sum_x \sqrt{\pi(x)}|x\rangle$ directly, using the Szegedy walk together with quantum phase estimation to project onto (a state close to) the eigenstate encoding $|\pi\rangle$. The number of applications of $W(P)$ needed to resolve the eigenvalue-1 eigenspace from its neighbours scales as $O(1/\sqrt{\delta})$ (Szegedy, 2004), mirroring the spectral-gap speedup from the theorem above.

Remark 2 (The $\pi_{\min}$ caveat)**:** The full runtime is more subtle than the spectral-gap factor alone: the success probability of projecting onto $|\pi\rangle$, starting from a generic initial state such as the uniform superposition, depends on the squared overlap of that state with $|\pi\rangle$, which can be as small as $\Omega(\pi_{\min})$ for $\pi_{\min} = \min_x \pi(x)$. In high-dimensional Bayesian posteriors, $\pi_{\min}$ can be exponentially small in the parameter dimension, so the direct (non-annealed) algorithm can require $1/\sqrt{\pi_{\min}}$ extra resources and underperform classical MCMC. **Annealed QMCMC** mitigates this with a slowly varying sequence $P_0, \ldots, P_r$ with π_0 easy to prepare and $\langle \pi_i | \pi_{i+1} \rangle \geq p > 0$, achieving total runtime $O(r\sqrt{\tau}\log^2(r/\epsilon)(1/p)\log(1/p))$ for relaxation time τ (Lopatnikova *et al.*, 2024). Practitioners should treat the direct algorithm with caution until this overlap is well controlled.

Table 4: Quadratic speedup of quantum walks over classical Markov chains, measured in number of walk steps.

Chain type	Classical	Quantum	Why it differs
Two-state chain	$O(1/(\alpha+\beta))$	$O(1/\sqrt{\alpha+\beta})$	Coherent rotation toward $\|\pi\rangle$ vs. random diffusion.
Random walk on cycle (hitting time)	$O(N^2)$	$O(N)$	Ballistic vs. diffusive propagation (Childs *et al.*, 2003; Kempe, 2003).
General reversible chain (mixing, Szegedy walk)	$O(1/\delta)$	$O(1/\sqrt{\delta})$	Phase gap $\Omega(\sqrt{\delta})$ (Szegedy, 2004); possible $\pi_{\min}$ overhead, see remark above.

[4]This is a hitting-time statement, not mixing-time: a discrete-time quantum walk on a cycle does not converge to a stationary distribution as a classical ergodic chain does, since evolution under a fixed unitary is reversible. The Szegedy-walk *mixing* speedup above uses phase estimation, not repeated unitary evolution, to localise on $|\pi\rangle$.

8. Maximum likelihood estimation

Maximum likelihood estimation is one of the most widely used methods for parameter estimation in all of statistics. Given data, we find the parameter value that makes the observed data most probable under the assumed model, a deceptively simple idea that underlies an enormous fraction of applied statistical practice, from simple coin-flip inference to the most complex regression models.

Definition 23 (Likelihood and MLE)**:** For i.i.d. $X_1, \ldots, X_n$ with density $f(x;\theta)$, the likelihood is $L(\theta) = \prod_i f(x_i;\theta)$, log-likelihood $\ell(\theta) = \sum_i \log f(x_i;\theta)$, and $\hat{\theta}_{\mathrm{MLE}} = \arg\max_\theta \ell(\theta)$.

Under standard regularity conditions, the MLE enjoys three important asymptotic properties that explain why it is the default choice across so much of statistics. First, **consistency**: $\hat{\theta}_{\mathrm{MLE}} \xrightarrow{p} \theta$ as $n \to \infty$, so with enough data the estimator converges to the true parameter value. Second, **asymptotic normality**: $\sqrt{n}(\hat{\theta}_{\mathrm{MLE}} - \theta) \xrightarrow{d} N(0, I(\theta)^{-1})$, where $I(\theta)$ is the Fisher information, so the estimator is approximately normally distributed around the truth for large n, which is what licenses the usual confidence-interval constructions. Third, **asymptotic efficiency**: the MLE achieves the Cramér-Rao lower bound asymptotically, meaning no other consistent, asymptotically unbiased estimator can do better in terms of asymptotic variance.

Example 18 (MLE of coin bias)**:** $\ell(\theta) = k\log\theta + (n-k)\log(1-\theta)$; $d\ell/d\theta = k/\theta - (n-k)/(1-\theta) = 0 \Rightarrow \hat{\theta}_{\mathrm{MLE}} = k/n$, confirmed a maximum since $d^2\ell/d\theta^2 = -k/\theta^2 - (n-k)/(1-\theta)^2 < 0$.

A particularly important special case of MLE, and one that will recur in Section 8's discussion of HHL, is linear regression. The model is $y = X\beta + \varepsilon$, where $y \in \mathbb{R}^n$ is the vector of responses, $X \in \mathbb{R}^{n\times d}$ is the design matrix, $\beta \in \mathbb{R}^d$ is the vector of regression coefficients we wish to estimate, and $\varepsilon \sim N(0, \sigma^2 I_n)$ is Gaussian noise. The log-likelihood under this model is $\ell(\beta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|y - X\beta\|^2$; since the first term does not depend on β, maximizing $\ell(\beta)$ is exactly equivalent to minimizing the residual sum of squares $\|y - X\beta\|^2$, which is the familiar ordinary least squares criterion. Differentiating and setting the gradient to zero gives the normal equations $X^\top X\hat{\beta} = X^\top y$, so whenever $X^\top X$ is invertible, $\hat{\beta} = (X^\top X)^{-1}X^\top y$, the closed-form solution every statistician encounters in their first regression course, here derived as a special case of the same maximum-likelihood machinery we used for the coin.

Quantum Phase Estimation (QPE)

Before we can understand the HHL algorithm, we need to understand Quantum Phase Estimation, a fundamental quantum subroutine that recurs throughout quantum algorithms wherever we need to extract information encoded in the phase of a unitary's eigenvalue. QPE estimates the phase θ_U in an eigenvalue $e^{2\pi i\theta_U}$ of a unitary U (Kitaev, 1995; Cleve *et al.*, 1998) (we write θ_U to distinguish from the coin bias θ). Given a unitary U and an eigenstate $|u\rangle$ such that $U|u\rangle = e^{2\pi i\theta_U}|u\rangle$, QPE produces an n-bit approximation of θ_U in an auxiliary register, using n ancilla qubits initialised to $|0\rangle$ and proceeding in four stages:

- **Step 1: Superposition.** Apply a Hadamard gate to each of the n ancilla qubits to create a uniform superposition: $\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}|k\rangle \otimes |u\rangle$. At this point the ancilla register

carries no information about θ_U at all; it is simply an equal-weight superposition over every possible n-bit string.

- **Step 2: Controlled unitaries.** Apply controlled-U^{2^j}, with the j-th ancilla qubit as control, for each j. Because $|u\rangle$ is an eigenstate of U, each controlled application multiplies the corresponding branch of the superposition by the appropriate power of the eigenvalue phase, leaving the state $\frac{1}{\sqrt{2^n}} \sum_k e^{2\pi i \theta_U k} |k\rangle \otimes |u\rangle$: the phase θ_U is now encoded in the relative phases between the $|k\rangle$ branches of the ancilla register, though it is not yet directly observable.
- **Step 3: Inverse QFT.** Applying the inverse Quantum Fourier Transform to the ancilla register converts these relative phases into an ordinary binary amplitude pattern, concentrating the amplitude on (or near) the n-bit binary expansion of θ_U.
- **Step 4: Measurement.** Measuring the ancilla register in the computational basis now yields the best n-bit estimate of θ_U with probability $\geq 4/\pi^2 \approx 0.4$ (exactly, if $2^n \theta_U$ happens to be an integer); this success probability can be boosted arbitrarily close to 1 using standard repetition or median tricks (Cleve *et al.*, 1998).

The HHL algorithm

The HHL algorithm, named after its inventors Harrow, Hassidim, and Lloyd (Harrow *et al.*, 2009), solves linear systems of equations exponentially faster than classical algorithms under certain conditions. This makes it a powerful tool for quantum maximum likelihood estimation, particularly for the linear regression problem we just worked through classically. For an improved version of the algorithm with better dependence on the target precision, see Childs *et al.* (2017); for applications of HHL-style techniques to quantum machine learning more broadly, see Rebentrost *et al.* (2014); Schuld *et al.* (2016); Kerenidis and Prakash (2017).

Problem statement. We are given an $N \times N$ Hermitian matrix A (non-Hermitian matrices can always be embedded into a larger Hermitian matrix via $\left(\begin{smallmatrix} 0 & A \\ A^\dagger & 0 \end{smallmatrix}\right)$, so this restriction costs no generality), assumed to be s-sparse, meaning at most s non-zero entries per row or column, and well-conditioned, with condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ reasonably small. We are also given a quantum state $|b\rangle = \frac{1}{\|b\|} \sum_i b_i |i\rangle$ encoding the right-hand side of the linear system. The goal is to produce the quantum state $|x\rangle = A^{-1}|b\rangle / \|A^{-1}|b\rangle\|$, encoding the (normalised) solution to $Ax = b$.

Theorem 8 (HHL runtime (Harrow *et al.*, 2009))**:** The original HHL algorithm solves the quantum linear systems problem in time $O(s^2 \kappa^2 \log N/\epsilon)$, where ϵ is the allowed error. For sparse matrices ($s = O(\text{polylog}\, N)$) and well-conditioned matrices ($\kappa = O(1)$), this is exponentially faster in N than the best classical algorithms, which require $O(Ns\kappa)$ time using sparse conjugate-gradient-type methods, or $O(N^\omega)$ with $\omega \approx 2.37$ for dense matrices via fast matrix multiplication. Subsequent work has substantially improved the dependence on ϵ and κ: in particular, Childs *et al.* (2017) achieve $O(s\kappa\, \text{polylog}(s\kappa/\epsilon))$ query complexity, exponentially better in $1/\epsilon$ than the original bound. We state the original $O(s^2\kappa^2 \log N/\epsilon)$ bound here because it corresponds directly to the elementary, easy-to-follow construction we walk through step by step below; readers implementing HHL in practice should use the

improved algorithm of Childs *et al.* (2017) instead.

Step-by-step HHL

- **Step 1: Quantum Phase Estimation.** The first step decomposes the input state $|b\rangle$ into the eigenbasis of A. With eigenvectors $|u_j\rangle$ and eigenvalues λ_j of A, write $|b\rangle = \sum_j \beta_j |u_j\rangle$; applying QPE to the unitary e^{iA} (constructed via Hamiltonian simulation (Lloyd, 1996; Berry *et al.*, 2007), since e^{iA} is not given to us directly) yields $\mathrm{QPE}(|b\rangle|0\rangle) = \sum_j \beta_j |u_j\rangle|\tilde{\lambda}_j\rangle$, entangling each eigenvector branch with an estimate of its own eigenvalue.
- **Step 2: Controlled rotation.** Conditional on the estimated eigenvalue $\tilde{\lambda}_j$ sitting in an auxiliary register, apply a small rotation to a fresh ancilla qubit: $|u_j\rangle|\tilde{\lambda}_j\rangle|0\rangle \mapsto |u_j\rangle|\tilde{\lambda}_j\rangle\Big(\sqrt{1-(C/\tilde{\lambda}_j)^2}|0\rangle + (C/\tilde{\lambda}_j)|1\rangle\Big)$, where the constant $C = O(1/\kappa)$ is chosen so that $C/|\tilde{\lambda}_j| \le 1$ for every j, this is always possible because every eigenvalue satisfies $\lambda_{\min} \le |\lambda_j| \le \lambda_{\max}$, so $C/|\lambda_j| \le C\kappa/\lambda_{\max} \le 1$ once $C = O(\lambda_{\max}/\kappa)$ after rescaling so that $\lambda_{\max} = O(1)$. The point of this rotation is that it secretly encodes the factor $1/\tilde{\lambda}_j$, exactly the eigenvalue-inversion we need to compute A^{-1}, into the amplitude of the auxiliary qubit being $|1\rangle$.
- **Step 3: Postselection.** We measure the auxiliary qubit and keep only the runs where it comes out $|1\rangle$; the probability of this happening is $p_{\mathrm{succ}} = \sum_j |\beta_j C/\tilde{\lambda}_j|^2$, which can be uncomfortably small when κ is large, this is precisely what drives the κ^2 dependence (or, in the improved algorithm, the milder κ dependence) in the runtime bound above, since naively we would need $O(1/p_{\mathrm{succ}})$ repetitions to see a successful run. In practice this naive overhead is avoided by applying amplitude amplification to the postselection step itself, boosting the expected number of repetitions down to $O(1)$. Conditional on success, the (unnormalised) state is $\sum_j \beta_j (C/\tilde{\lambda}_j)|u_j\rangle|\tilde{\lambda}_j\rangle|1\rangle$, which we then renormalise by $1/\sqrt{p_{\mathrm{succ}}}$.
- **Step 4: Uncomputation.** Finally, we apply the inverse of QPE to clear the eigenvalue register back to $|0\rangle$, leaving $|x\rangle = \frac{1}{\|A^{-1}|b\rangle\|}\sum_j \frac{\beta_j}{\lambda_j}|u_j\rangle = A^{-1}|b\rangle/\|A^{-1}|b\rangle\|$, exactly the normalised solution vector we wanted, encoded entirely in the amplitudes of a quantum state.

Example 19 (2×2 system)**:** Consider the simple 2×2 system $A = \mathrm{diag}(2,1)$, $b = (1,1)$. Classically, the solution is immediate: $A^{-1}b = (1/2, 1)$. Let us verify that the quantum construction above reproduces this answer. In the quantum setting, the eigenvalues of A are $\lambda_1 = 2, \lambda_2 = 1$, with corresponding eigenvectors $|u_1\rangle = |0\rangle$, $|u_2\rangle = |1\rangle$; the normalised right-hand side is $|b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, so $\beta_1 = \beta_2 = 1/\sqrt{2}$. QPE records the eigenvalue estimates $\tilde{\lambda}_1 \approx 2$, $\tilde{\lambda}_2 \approx 1$. The controlled rotation in Step 2 then multiplies the amplitude on $|0\rangle$ by $C/2$ and the amplitude on $|1\rangle$ by $C/1 = C$, giving (before normalisation) amplitudes $(1/\sqrt{2})(C/2) = C/(2\sqrt{2})$ on $|0\rangle$ and $(1/\sqrt{2})(C) = C/\sqrt{2}$ on $|1\rangle$. After postselection and uncomputation, renormalising while preserving this ratio gives:

$$|x\rangle = \frac{C/(2\sqrt{2})}{\sqrt{(C/(2\sqrt{2}))^2 + (C/\sqrt{2})^2}}|0\rangle + \frac{C/\sqrt{2}}{\sqrt{(C/(2\sqrt{2}))^2 + (C/\sqrt{2})^2}}|1\rangle = \frac{1}{\sqrt{5}}|0\rangle + \frac{2}{\sqrt{5}}|1\rangle,$$

which is exactly proportional to $A^{-1}b = (1/2, 1)$, matching the classical answer exactly up to the overall normalisation that any quantum state must carry. This small example, worked through in full, is meant to make the four abstract steps above concrete enough to trust before we discuss the algorithm's broader implications.

Why is HHL exponentially faster than classical algorithms, at least in N and under the sparsity and conditioning assumptions stated above? The key conceptual shift is that HHL works with quantum states that encode the solution, rather than with classical vectors that must be written down entry by entry. The matrix A is never fully stored in memory; instead, it is accessed only through a sparse oracle that can be queried in superposition. Quantum phase estimation then allows HHL to "invert" all eigenvalues simultaneously, exploiting quantum parallelism in a way that has no classical counterpart, while Hamiltonian simulation (Lloyd, 1996; Berry *et al.*, 2007) provides the time evolution e^{-iAt} that QPE needs as its input unitary.

Remark 3 (Limitations)**:** Two important limitations temper this exponential promise. First, the speedup is exponential in N only for sparse, well-conditioned matrices; for dense or ill-conditioned matrices the theoretical advantage disappears entirely, since both s and κ then scale with N themselves. Second, and perhaps more importantly for practitioners, the output of HHL is a *quantum state* $|x\rangle$, not a classical vector. Extracting the full classical solution vector requires quantum tomography, which as we will see in Section 10 takes $O(N/\epsilon^2)$ measurements, potentially negating the speedup entirely if the full vector is what is actually needed (O'Donnell and Wright, 2016). Data loading, meaning the preparation of $|b\rangle$ itself in the first place, can also be a serious bottleneck (Giovannetti *et al.*, 2008; Kerenidis and Prakash, 2017).

Example 20 (Quantum linear regression)**:** Suppose we have n data points in d dimensions, with $n \ll d$, a high-dimensional regression setting that is increasingly common in modern applications. Classical ridge regression, solving $(X^\top X + \lambda I)\hat{\beta} = X^\top y$, costs $O(d^3)$ time in general, or $O(nd^2)$ using the Woodbury identity when $n \ll d$ lets us avoid inverting the full $d \times d$ matrix directly. If the design matrix X is low-rank and we have quantum access to the data via QRAM (Giovannetti *et al.*, 2008; Kerenidis and Prakash, 2017), the HHL algorithm can estimate $\hat{\beta}$ in $O(\text{polylog}(d))$ time (Rebentrost *et al.*, 2014; Schuld *et al.*, 2016), an exponential speedup in the dimension d, though subject to the same sparsity, conditioning, and readout caveats discussed throughout this section.

Remark 4 (When HHL helps, and when it doesn't)**:** HHL's exponential-in-N speedup needs both sparsity and good conditioning. Statistical design matrices $X^\top X$ are rarely sparse, and ill-conditioning is common in high-dimensional or collinear regression, exactly the regime where most challenging statistical problems live. In such settings classical ridge regression or conjugate gradient may outperform the quantum approach; sparsity and condition number should be verified explicitly before invoking HHL.

9. Bayesian statistics and posterior mode

Bayesian statistics provides a coherent framework for updating beliefs in light of data, and the posterior distribution it produces encodes all of our uncertainty about the

parameters after observing that data. The basic machinery, Bayes' rule and the law of total probability, was already introduced in Section 3; here we specialise that machinery to the Bayesian modelling workflow and then turn, as in every previous section, to its quantum analogue.

Definition 24 (Bayesian model)**:** A Bayesian model consists of a prior distribution $p(\theta)$ over parameters $\theta \in \Theta$, expressing our beliefs about θ before seeing any data, together with a likelihood $p(y \mid \theta)$ for the observed data y given the parameters. Bayes' theorem, exactly as stated in Section 3, then gives the posterior distribution $p(\theta \mid y) = p(y \mid \theta)p(\theta)/p(y) \propto p(y \mid \theta)p(\theta)$, where $p(y) = \int p(y \mid \theta)p(\theta)\, d\theta$ is the marginal likelihood, or evidence (Gelman *et al.*, 2013).

The posterior distribution is, in principle, the complete answer to any inference question we might ask about θ. In practice, however, it is often high-dimensional and difficult to summarise directly, so two single-number summaries are reported most often. The **posterior mean** is $\mathbb{E}[\theta \mid y] = \int \theta\, p(\theta \mid y)\, d\theta$; the **posterior mode**, also called the maximum a posteriori (MAP) estimate, is $\hat{\theta}_{\mathrm{MAP}} = \arg\max_\theta p(\theta \mid y) = \arg\max_\theta [\log p(y \mid \theta) + \log p(\theta)]$. These two summaries coincide only when the posterior happens to be symmetric; in general they can differ substantially, particularly for skewed posteriors. When the prior is uniform, the MAP estimate coincides exactly with the MLE from Section 8, since the prior term then contributes nothing to the maximisation.

Example 21 (Beta prior)**:** Suppose we place a Beta(α, β) prior on the coin bias, $p(\theta) = \theta^{\alpha-1}(1-\theta)^{\beta-1}/B(\alpha,\beta)$. The Beta distribution is conjugate to the binomial likelihood, meaning the posterior is again a Beta distribution: after observing k heads in n flips, $p(\theta \mid k) \propto p(k \mid \theta)p(\theta) \propto \theta^k(1-\theta)^{n-k} \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}$, which we recognise as proportional to a Beta$(k+\alpha, n-k+\beta)$ density. For $k+\alpha > 1$ and $n-k+\beta > 1$, the mode of a Beta(a,b) distribution is $(a-1)/(a+b-2)$, giving $\hat{\theta}_{\mathrm{MAP}} = (k+\alpha-1)/(n+\alpha+\beta-2)$. When $\alpha = \beta = 1$ (the uniform prior), this reduces exactly to the MLE k/n, confirming the general fact noted above (Casella and Berger, 2002).

Quantum Bayesian inference via QMCMC

The quantum posterior state $|\pi\rangle = \sum_\theta \sqrt{\pi(\theta)}|\theta\rangle$ for $\pi(\theta) = p(\theta \mid y)$ is exactly the quantum sample state of Section 4 applied to the posterior. The Szegedy walk construction of Section 7 applies verbatim: given a classical chain on the parameter space with stationary distribution π and spectral gap δ, $W(P)$ has $|\pi\rangle$ as an eigenvalue-1 eigenstate, with $O(1/\sqrt{\delta})$ walk-step dependence versus $O(1/\delta)$ classically, subject to the same $\pi_{\min}$ caveat from Section 7, especially relevant here since high-dimensional posteriors are exactly where $\pi_{\min}$ tends to be exponentially small. For a detailed treatment of quantum Bayesian inference, see Wiebe *et al.* (2012).

Posterior mode via quantum search

Finding the posterior mode is fundamentally an optimization problem: we are searching over the parameter space for the point of highest posterior density. Grover's search al-

gorithm (Grover, 1996) provides a quadratic speedup for unstructured search, as we saw in Section 6, and its generalization, due to Dürr and Høyer, adapts naturally to the related task of finding the maximum of a function rather than a single marked item.

Definition 25 (Quantum maximum finding algorithm (Durr and Hoyer, 1996))**:** The algorithm proceeds iteratively, using Grover's algorithm as a subroutine at each step. First, randomly select an initial index y and compute $f(y)$, establishing a baseline threshold. Then repeat the following until convergence: use the current threshold $f(y)$ to mark all items x such that $f(x) > f(y)$ (these are the items that would improve on our current best guess); apply Grover's algorithm to amplify the marked items, exactly as in Section 6; measure to obtain a new index y' with $f(y') > f(y)$; and update $y = y'$ before repeating. Each round of this loop costs $O(\sqrt{N})$ Grover queries against the current (shrinking) set of marked items, and a careful accounting of the expected number of rounds needed shows that the algorithm finds the global maximum using expected $O(\sqrt{N})$ oracle calls in total (Durr and Hoyer, 1996), the same square-root scaling we have now seen repeatedly throughout this article.

Theorem 9 (Quantum MAP speedup)**:** Finding $\hat{\theta}_{\text{MAP}} = \arg\max_\theta \pi(\theta)$ over N discrete values needs expected $O(\sqrt{N})$ queries to a posterior-comparison oracle, versus $O(N)$ classically.

Example 22 (Quantum MAP)**:** Suppose we discretise the space of possible coin biases into $\Theta = \{0.001, 0.002, \ldots, 0.999\}$, a set of size $N = 999$. After observing some data, we want to find the bias with the highest posterior probability. Classical exhaustive search must check all 999 values, requiring 999 posterior evaluations. Quantum maximum finding requires only approximately $\sqrt{999} \approx 32$ queries to the posterior evaluation oracle, a striking reduction even at this modest scale. The advantage becomes dramatic as the parameter space grows: for $N = 1$ million candidate values, classical search still requires the full 1 million evaluations, while quantum maximum finding requires only about 1000.

Table 5: Quadratic speedups for Bayesian computation, in respective query/step complexity models (coherent oracle access assumed; see Section 10 for caveats governing wall-clock advantage)

Task	Classical	Quantum	Why it differs
Posterior sampling (MCMC)	$O(1/\delta)$	$O(1/\sqrt{\delta})$	Szegedy walk (Szegedy, 2004); coherent rotation vs. random diffusion.
Posterior mode (MAP)	$O(N)$	$O(\sqrt{N})$ expected	Grover/max-finding (Durr and Hoyer, 1996).
Posterior mean	$O(1/\epsilon^2)$	$O(1/\epsilon)$	QAE (Brassard *et al.*, 2002; Montanaro, 2015); away from boundary regime.

10. Practical considerations and open problems

The exponential-in-N speedups of quantum algorithms like HHL and quantum principal component analysis (Lloyd *et al.*, 2014) all assume that data can be loaded into quantum states efficiently. This assumption deserves scrutiny: it is not always true, and the two bottlenecks below, data loading and state readout, together govern whether a provable

query-complexity speedup ever shows up as a real, wall-clock advantage on actual hardware.

Definition 26 (QRAM (Giovannetti *et al.*, 2008))**:** QRAM is a data structure that allows quantum superposition access to classical data, the quantum analogue of an ordinary computer's random access memory, but built to accept a query in superposition and return the corresponding superposition of answers. It performs $\sum_i a_i|i\rangle|0\rangle \mapsto \sum_i a_i|i\rangle|x_i\rangle$ with query *depth* $O(\log N)$, but the bucket-brigade architecture achieving this requires $O(N)$ physical components and very high-fidelity operations on all of them simultaneously, challenging with current hardware, and a major reason QRAM remains more a theoretical assumption than an engineering reality at present. An alternative architecture is given by Kerenidis and Prakash (2017).

Closely related to QRAM is the separate question of *amplitude encoding* cost. Loading an arbitrary vector $x \in \mathbb{C}^N$ into amplitudes $|x\rangle = \sum_i x_i|i\rangle$ requires $O(N)$ operations in general, since each amplitude must in principle be individually programmed via a sequence of controlled rotations (Prakash, 2014); there is no shortcut around this for a truly generic vector, since the circuit must, in effect, specify N independent real numbers. Structured data can sometimes avoid this cost entirely: Grover and Rudolph (Grover and Rudolph, 2002) showed that distributions with an efficiently integrable cumulative distribution function, the normal distribution being the canonical example, can be prepared using only $O(\text{polylog}\, N)$ operations, an exponential improvement over the generic bound. But Herbert (2021) showed that this method is fundamentally limited to distributions simple enough to not need Monte Carlo in the first place, precisely the distributions for which a quantum speedup is least needed in practice. This raises real doubt about whether efficient quantum state preparation is available for the complex, empirical distributions that statisticians actually encounter in real applications, as opposed to the clean textbook distributions used to illustrate the algorithms.

Definition 27 (Quantum and shadow tomography)**:** Full state tomography of an N-dimensional pure state requires $\Theta(N/\epsilon^2)$ copies to achieve error ϵ in trace distance (O'Donnell and Wright, 2016), impossible for large N (*e.g.* $N = 2^{100}$). This means that for many quantum algorithms that produce an N-dimensional solution vector, we genuinely cannot read out the entire vector, no matter how cheap state preparation was. **Shadow tomography** (Huang *et al.*, 2020) instead estimates M chosen observables to precision ϵ using only $O(\log M)$ copies (with additional polynomial dependence on $1/\epsilon$), via random unitary rotations before measurement, sufficient for the low-dimensional summaries (regression coefficients, moments, threshold tests) statisticians usually want, even when full tomography is hopeless.

To make the contrast concrete: suppose HHL has just solved a regression problem with $d = 20$ qubits worth of coefficients, so $N = 2^{20} \approx 10^6$ coefficients in total. Reading out the entire solution vector to modest precision $\epsilon = 0.01$ via full tomography would require on the order of $N/\epsilon^2 \approx 10^{10}$ copies of the state, wildly impractical even on a perfect quantum computer. But if a data analyst only actually needs, say, the 20 largest coefficients or a single linear combination of them, shadow tomography can extract that summary using only on the order of $\log(20) \approx$ 4-ish copies, up to the polynomial overhead in $1/\epsilon$: a difference of many orders of magnitude that determines, in practice, whether HHL's exponential speedup in state *preparation* survives contact with the need to actually read an answer back out.

Putting the data-loading and readout bottlenecks together, we can now summarise, in one place, when quantum advantage is plausible for a given statistical problem and when it is not. Table 6 below is meant to function as a quick diagnostic checklist: a problem with several entries from the left column is a genuinely promising candidate for a quantum approach, while a problem with several entries from the right column is likely to see its theoretical speedup erased in practice by one bottleneck or another, however elegant the underlying algorithm.

Table 6: Conditions for quantum advantage. NISQ (Noisy Intermediate-Scale Quantum) was coined by Preskill (2018)

Favorable conditions	**Unfavorable conditions**
Sparse or low-rank matrices	Dense, full-rank matrices
Well-conditioned (κ small)	Ill-conditioned (κ large)
Quantum access to data (QRAM available)	Classical data requiring amplitude encoding
Low-dimensional summary needed	Full solution vector required
Quantum queries dominate runtime	Data loading or readout dominates
Quadratic speedup is sufficient	Exponential speedup needed but unrealistic
Current NISQ devices (Preskill, 2018)	Fault-tolerant quantum computers required

Open problems for statisticians

The intersection of quantum computing and statistics has many open problems that are ripe for research. We close this section with six of them.

1. **Efficient quantum sample preparation**: Preparing quantum sample states for arbitrary distributions of practical interest remains an open problem. Current methods either require unrealistic assumptions (Grover–Rudolph (Grover and Rudolph, 2002)) or scale poorly (general amplitude encoding (Prakash, 2014)). Statisticians could help by designing preparation methods tailored to structured distributions that arise often in practice, exponential families, mixture models, Bayesian priors, rather than aiming for full generality.
2. **Quantum sequential Monte Carlo**: Quantizing particle filters for state-space models is an open research question. Sequential Monte Carlo is an important alternative to MCMC for high-dimensional, time-varying problems, and a quantum version could in principle offer similar speedups to those we saw for Szegedy walks.
3. **Quantum variational Bayes**: While hybrid quantum-classical variational algorithms already exist (Schuld *et al.*, 2020), implementing variational inference entirely on a quantum computer, rather than splitting the work between quantum and classical processors, is still an open problem.
4. **Quantum normalizing flows**: Using parameterized quantum circuits to transform quantum sample states may enable more efficient readout than generic shadow tomography, inspired by classical normalizing flows that transform simple distributions into complex ones.

5. **Quantum-inspired classical algorithms**: Some quantum algorithms have, perhaps surprisingly, led to improved *classical* algorithms. Quantum-inspired linear algebra (Gilyén *et al.*, 2022) achieves polynomial speedups on ordinary classical computers for certain low-rank problems; adapting these ideas to statistical computation is an active area that requires no quantum hardware at all.

6. **Better readout methods**: Quantum tomography is too expensive for large states, and shadow tomography (Huang *et al.*, 2020) is a promising start, but more work is needed to develop methods that extract precisely the summaries statisticians care about, posterior means, credible intervals, Bayes factors, rather than generic observable expectations.

What unites all six of these open problems is that they require genuine statistical expertise to make progress on, not just quantum computing expertise: knowing which distributions actually arise in practice, which summaries practitioners actually need, and which approximations are acceptable for a given application are statistical judgments, not algorithmic ones. This is precisely why the field needs more statisticians working in it, and why this article exists at all.

11. Conclusion

We have travelled a long road together. We began with Kolmogorov's axioms and built up, step by step, to the quantum algorithms that promise to transform statistical computation, returning at every stage to the same single biased coin to ground the abstraction in something concrete. Sections 2–4 established the dictionary between classical probability and quantum states: Hilbert spaces stand in for sample spaces, state vectors for distributions, observables for random variables, and quantum sample states $|P\rangle = \sum_x \sqrt{p(x)}|x\rangle$ encode entire distributions in $O(\log N)$ qubits of storage, an idea we first met as an abstraction in Section 4 and then watched concretely turn our 101-outcome binomial distribution into a 7-qubit object. Section 5 carried expectation and variance across the same dictionary, showing that $\langle O\rangle = \mathrm{tr}(O\rho)$ is exactly the quantum mirror of $\mathbb{E}[X] = \sum_x xp(x)$, with the coin's Pauli-$Z$ expectation $2\theta - 1$ as the worked check. Sections 6–9 then built the four central algorithms on top of this dictionary, each illustrated on the coin or on a discrete approximation to it: Grover's search and Quantum Amplitude Estimation (Section 6) give a quadratic speedup for mean estimation, taking our coin-bias estimation problem from 10^6 classical flips down to roughly 1000 quantum oracle queries; Szegedy quantum walks (Section 7) give a quadratic speedup for MCMC mixing; the HHL algorithm (Section 8) gives an exponential speedup for sparse, well-conditioned linear systems, worked through explicitly on a 2×2 example; and quantum maximum-finding (Section 9) gives a quadratic speedup for locating the posterior mode of the coin's bias. Section 10 then stepped back and detailed the two practical bottlenecks, data loading and state readout, that govern whether any of these provable speedups translates into real-world, wall-clock advantage on actual hardware.

$$\boxed{\text{Probability } p(x) \xrightarrow{\sqrt{\cdot}} \text{Amplitude } \sqrt{p(x)} \xrightarrow{|\cdot\rangle} \text{State } \sum_x \sqrt{p(x)}|x\rangle}$$

This is the quantum dialect of probability: not a replacement for classical probability but an extension of it. Classical probability is the special case where every relevant decomposition is orthogonal, so amplitudes effectively add like ordinary probabilities and nothing distinctively quantum ever happens. Quantum probability allows coherent superpositions of non-orthogonal alternatives, whose amplitudes can interfere constructively or destructively *before* the Born rule converts them into probabilities, and this interference, which has simply no classical counterpart whatsoever, is the engine of every speedup in this article, from Grover's search through Szegedy walks to HHL. If there is one idea a statistician should carry away from this entire article, it is this: quantum advantage is not magic, and it is not merely "more computing power." It is a specific, well-understood mathematical mechanism, constructive and destructive interference between coherently maintained amplitudes, that has no analogue anywhere in classical probability theory, and understanding where that mechanism can and cannot be exploited is the key to understanding when quantum computing will actually help with a given statistical problem.

Table 7: Classical probability and statistics vs. quantum analogues (concepts)

Classical	**Quantum**	**Why it differs**
Sample space Ω	Hilbert space $\mathcal{H}$	States are vectors over $\mathbb{C}$, not sets.
Distribution $p(\omega)$	State vector $\lvert\psi\rangle$	Amplitudes $\psi_i \in \mathbb{C}$; $P(i) = \lvert\psi_i\rvert^2$.
$p_i \geq 0$, add	$\psi_i \in \mathbb{C}$; add within a basis	Interference for non-orthogonal superpositions.
$X : \Omega \to \mathbb{R}$	Observable $O = \sum_o o\lvert o\rangle\langle o\rvert$	Eigenvalues replace function values.
$\mathbb{E}[X] = \sum_x xp(x)$	$\langle O\rangle = \mathrm{tr}(O\rho) = \langle\psi\rvert O\lvert\psi\rangle$	Same structure; p from Born rule.
Condition on ω	Collapse to $\lvert i\rangle$; $\rho_m = M_m\rho M_m^\dagger/p_m$	Measurement changes the physical state.
Bayes' rule (unique)	A quantum Bayes' update (Wiebe *et al.*, 2012) (one of several)	No single canonical analogue exists.
Mixture (epistemic)	Mixed state $\rho = \sum_i p_i\lvert\psi_i\rangle\langle\psi_i\rvert$	No interference (classical ignorance).
No classical analogue	Pure superposition $\lvert\psi\rangle$	Ontological uncertainty, not ignorance.

What quantum computing can do

Quadratic speedups in query complexity are real and achievable for two of the central tasks of statistical computation. Quantum Amplitude Estimation (Brassard *et al.*, 2002; Montanaro, 2015) and Szegedy quantum walks (Szegedy, 2004) offer provable quadratic speedups for mean estimation and MCMC sampling respectively, subject to the boundary-regime and $\pi_{\min}$ caveats discussed throughout this article. These algorithms require coherent quantum computation, but they are an active target for near-term NISQ devices (Preskill, 2018) under reasonable assumptions about hardware quality. Exponential speedups also exist, most notably HHL, but they require sparsity and good conditioning, and the precise polynomial dependence on these two parameters matters a great deal in practice; this dependence was substantially improved by Childs *et al.* (2017) relative to the original 2009

Table 8: Classical probability and statistics vs. quantum analogues (algorithmic speedups; query/step complexity under coherent oracle access, see Section 10 for data-loading and readout caveats)

Classical	Quantum	Why it differs
Monte Carlo: $O(1/\epsilon^2)$ (Metropolis and Ulam, 1949)	QAE: $O(1/\epsilon)$ (Brassard *et al.*, 2002; Montanaro, 2015)	Phase estimation replaces repeated sampling; away from boundary regime.
MCMC mixing: $O(1/\delta)$ (Geyer, 1992)	Quantum walk: $O(1/\sqrt{\delta})$ (Szegedy, 2004)	Coherent rotation; possible $\pi_{\min}$ overhead.
Linear systems: $O(Ns\kappa)$	HHL: $O(s^2\kappa^2 \log N/\epsilon)$ (Harrow *et al.*, 2009); improved $O(s\kappa \,\mathrm{polylog})$ (Childs *et al.*, 2017)	Exponential saving in N only for sparse, well-conditioned matrices.
Maximum finding: $O(N)$	Grover/Dürr–Høyer: expected $O(\sqrt{N})$ (Grover, 1996; Durr and Hoyer, 1996)	Interference amplifies marked-state amplitude.
Posterior mode: $O(N)$	Quantum max-finding: expected $O(\sqrt{N})$ (Durr and Hoyer, 1996)	Same mechanism applied to MAP.

algorithm. Finally, quantum sample states are exponentially efficient to *store*: an N-outcome distribution that would need N classical numbers fits into $\log_2 N$ qubits. Both preparation (for unstructured distributions) and readout, however, remain genuinely hard in general, as Section 10 detailed.

What it cannot (yet) do

Quantum computers are not known to solve NP-complete problems in polynomial time. The class BQP (bounded-error quantum polynomial time) is believed to be incomparable with NP rather than a strict superset of it, though this remains an open question in complexity theory and should not be overstated either way. More practically: speedups are problem-specific, not universal. For many real statistical problems, the data-loading bottleneck (Giovannetti *et al.*, 2008; Grover and Rudolph, 2002) or the readout bottleneck (O'Donnell and Wright, 2016) silently erases the asymptotic advantage that looked so attractive on paper. Quantum methods, at least for the foreseeable future, will complement classical statistics rather than replace it; the most promising near-term path forward is hybrid quantum-classical algorithms that split work between the two kinds of processor (Schuld *et al.*, 2020).

A practical checklist

When considering whether a quantum approach is worth pursuing for a particular statistical problem, four questions are worth asking in order. **Can the data be loaded ef-**

ficiently? If it has known structure (*e.g.* comes from an efficiently integrable distribution), Grover–Rudolph-style preparation (Grover and Rudolph, 2002) may work; otherwise QRAM (Giovannetti *et al.*, 2008; Kerenidis and Prakash, 2017) or another loading scheme will be needed, at real hardware cost. **Is the full solution vector actually needed, or just a summary?** If only a few coefficients, a credible interval, or a hypothesis-test result is required, shadow tomography (Huang *et al.*, 2020) can extract it efficiently; if the entire vector must be read out classically, full tomography (O'Donnell and Wright, 2016) may erase the speedup entirely. **Is the relevant matrix sparse and well-conditioned?** HHL's exponential advantage depends on both properties holding, and most statistical design matrices satisfy neither. **Is a quadratic speedup actually enough for the application, or is an unrealistic exponential one being implicitly assumed?** Quadratic speedups for mean estimation and MCMC are proven and practical targets; exponential speedups require much more careful justification.

Where to go next

This article has provided the conceptual foundation; to go deeper, several resources are worth knowing about.

For the full mathematical machinery of quantum computation: Nielsen and Chuang's (Nielsen and Chuang, 2010) *Quantum Computation and Quantum Information* is the standard graduate textbook, covering all the algorithms we discussed and a great deal more; Kitaev, Shen, and Vyalyi's *Classical and Quantum Computation* offers a more theoretical, complexity-focused treatment. For a comprehensive statistician-oriented survey with detailed complexity analyses that goes well beyond what we covered here: Lopatnikova, Tran, and Sisson (Lopatnikova *et al.*, 2024), the review that directly inspired this article. For quantum machine learning specifically: Schuld and Petruccione's (Schuld *et al.*, 2016) *Supervised Learning with Quantum Computers*.

For hands-on practice rather than further reading: Qiskit (IBM) is a Python-based quantum computing SDK with free access to real IBM quantum hardware and extensive tutorials; Cirq (Google) is a comparable Python framework oriented toward NISQ-era algorithms; and Amazon Braket offers cloud-based access to multiple quantum computing platforms from different hardware vendors in one place. The Qiskit Global Summer School runs annually and is free to attend, and is a good way to go from reading about these algorithms to actually running them.

The field is young, the problems are rich, and the opportunities for statisticians are vast. The next great statistical method may be quantum; the next great quantum algorithm may come from a statistician. The door is open.

Acknowledgements

Both authors were supported by a generous grant to CMI by the Infosys Foundation. We thank the anonymous reviewers for their valuable time, thorough evaluation, and constructive feedback, which significantly enhanced the quality of this manuscript.

Conflict of interest

The authors do not have any financial or non-financial conflict of interest to declare for the research work included in this article.

References

Berry, D. W., Ahokas, G., Cleve, R., and Sanders, B. C. (2007). Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, **270**, 359–371.

Billingsley, P. (1995). *Probability and Measure*. Wiley.

Brassard, G., Hoyer, P., Mosca, M., and Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics*, **305**, 53–74.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury.

Childs, A. M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., and Spielman, D. A. (2003). Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 59–68.

Childs, A. M., Kothari, R., and Somma, R. D. (2017). Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, **46**, 1920–1950.

Cleve, R., Ekert, A., Macchiavello, C., and Mosca, M. (1998). Quantum algorithms revisited. *Proceedings of the Royal Society A*, **454**, 339–354.

Durr, C. and Hoyer, P. (1996). A quantum algorithm for finding the minimum. Technical report.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC Press.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, **7**, 473–483.

Gilyén, A., Song, Z., and Tang, E. (2022). An improved quantum-inspired algorithm for linear regression. *Quantum*, **6**, 754.

Giovannetti, V., Lloyd, S., and Maccone, L. (2008). Quantum random access memory. *Physical Review Letters*, **100**, 160501.

Grover, L. and Rudolph, T. (2002). Creating superpositions that correspond to efficiently integrable probability distributions.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA. Association for Computing Machinery.

Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, **103**, 150502.

Herbert, S. (2021). No quantum speedup with Grover-Rudolph state preparation for quantum monte carlo integration. *Physical Review E*, **103**, 063302.

Huang, H.-Y., Kueng, R., and Preskill, J. (2020). Predicting many properties of a quantum system from very few measurements. *Nature Physics*, **16**, 1050–1057.

Kempe, J. (2003). Quantum random walks: An introductory overview. *Contemporary Physics*, **44**, 307–327.

Kerenidis, I. and Prakash, A. (2017). Quantum recommendation systems. In Papadimitriou, C. H., editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Kitaev, A. Y. (1995). Quantum measurements and the Abelian Stabilizer Problem. *Electron. Colloquium Comput. Complex.*, **TR96**.

Lloyd, S. (1996). Universal quantum simulators. *Science*, **273**, 1073–1078.

Lloyd, S., Mohseni, M., and Rebentrost, P. (2014). Quantum principal component analysis. *Nature Physics*, **10**, 631–633.

Lopatnikova, A., Tran, M.-N., and Sisson, S. A. (2024). An introduction to quantum computing for statisticians and data scientists. *Foundations of Data Science*, **6**, 278–307.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, **44**, 335–341.

Montanaro, A. (2015). Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A*, **471**, 20150301.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

O'Donnell, R. and Wright, J. (2016). Efficient quantum tomography. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 899–912.

Prakash, A. (2014). *Quantum Algorithms for Linear Algebra and Machine Learning*. PhD thesis, University of California, Berkeley, Berkeley, CA.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, **2**, 79.

Rebentrost, P., Mohseni, M., and Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters*, **113**, 130503.

Schuld, M., Bocharov, A., Svore, K. M., and Wiebe, N. (2020). Circuit-centric quantum classifiers. *Physical Review A*, **101**, 032308.

Schuld, M., Sinayskiy, I., and Petruccione, F. (2016). Prediction by linear regression on a quantum computer. *Physical Review A*, **94**, 022342.

Szegedy, M. (2004). Quantum speed-up of markov chain based algorithms. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–41.

Wiebe, N., Braun, D., and Lloyd, S. (2012). Quantum algorithm for data fitting. *Physical Review Letters*, **109**, 050505.