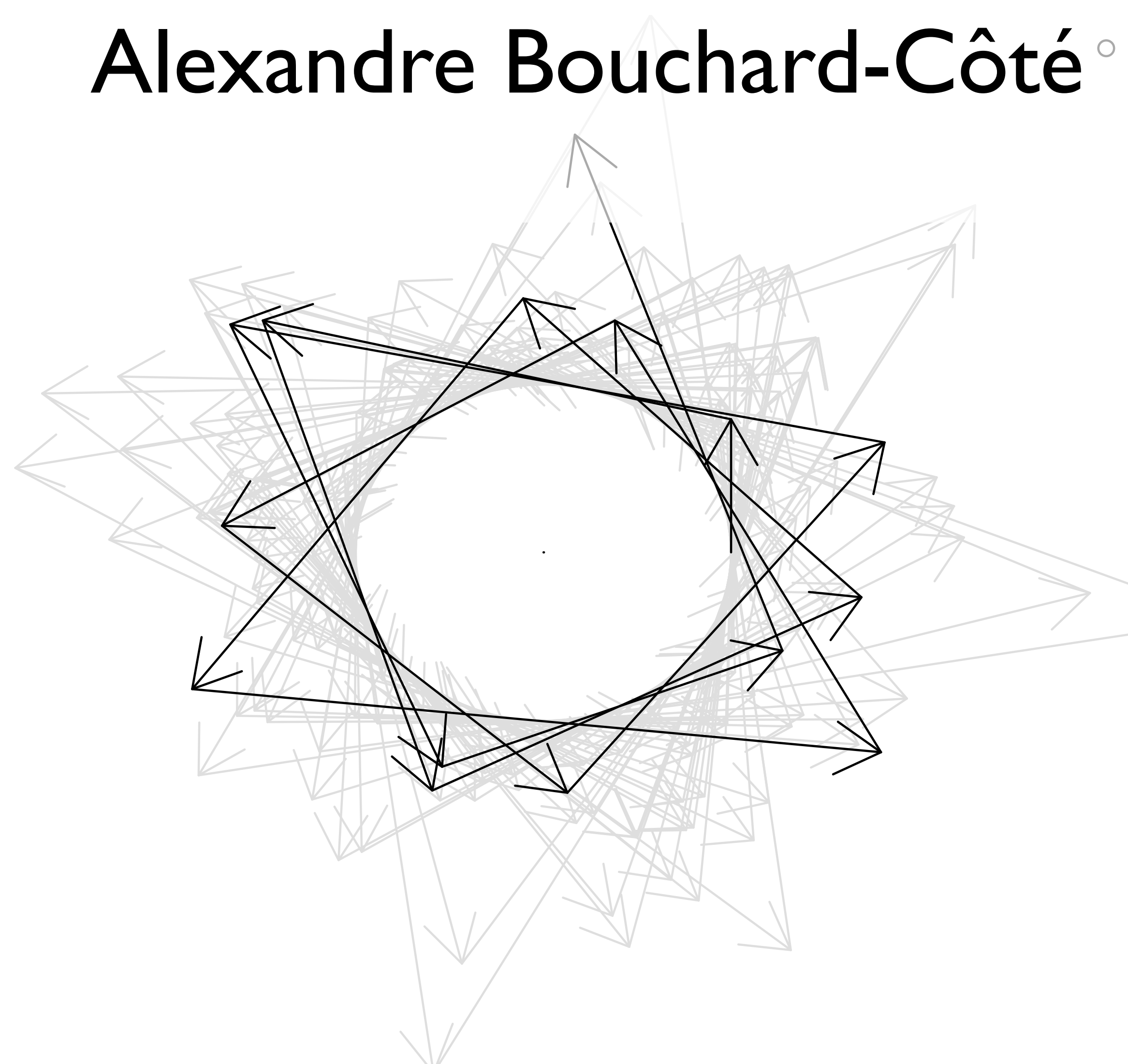
# Monte Carlo methods

Alexandre Bouchard-Côté



# Review / Q&A

# *Computing difficult integrals using the law of large numbers*

**Goal:** Compute $I = \int_{\mathbb{X}} \phi(x)\,\pi(x)\,dx$

**Monte Carlo Methods:** Truncate a *Law of Large Numbers* (LLN) converging to $I$.

**Example (simple Monte Carlo):**

- Simulate independent $X_1, \dots, X_n$ from $\pi$.
- Return $\widehat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \phi(X_i)$.

# Optimization and simulation

*x* : unknown
*y* : data

Cases where this is advantageous...

$$x^\star \in \operatorname{argmax}\, p(y|x)$$

$$x^{(i)} \sim \frac{1}{Z} p(x) p(y|x)$$

# Setup: large scale Bayesian or random effects models

- We are given a density known up to a normalization constant

$$\pi(x) = \frac{\gamma(x)}{Z}$$

- Example: $$\pi(x) = \frac{\mathrm{joint}(x, y)}{\mathrm{evidence}(y)}$$

$x$ : unknown
$y$ : data

- We want a law of large number

$$\frac{1}{N}\sum_{i=1}^{N} \varphi(X^{(i)}) \to \int \varphi(x)\pi(\mathrm{d}x) \text{ a.s.}$$

*'test function'*

***Note:** We almost never care about the samples themselves!*

Note 5

# Overview of the literature

Computationally expensive → Computationally efficient

Few assumptions ← More assumptions

| $y \sim p(\cdot \mid x)$ | $\gamma(x) = p(x)p(y \mid x)$ | $\nabla \log \gamma(x)$ |
|---|---|---|
| • Approximate Bayesian Computation (ABC)<br>• 'Plug-and-play' Sequential Monte Carlo (SMC) | • Random walk Metropolis<br>• Gibbs sampling  | • Hamiltonian Monte Carlo (HMC)<br>• Langevin |

# Overview of the literature

**'Naive' MCMC**

- Deterministic start
- Apply kernel *ad nauseam*
- Burn-in, etc

**Modern methods**

- Sequential change of measure-based
- Replica-based methods

# Simple example

- Consider d-dimensional iid random standard normal vectors
- What is the mean of the distance to the origin?
- Write as:

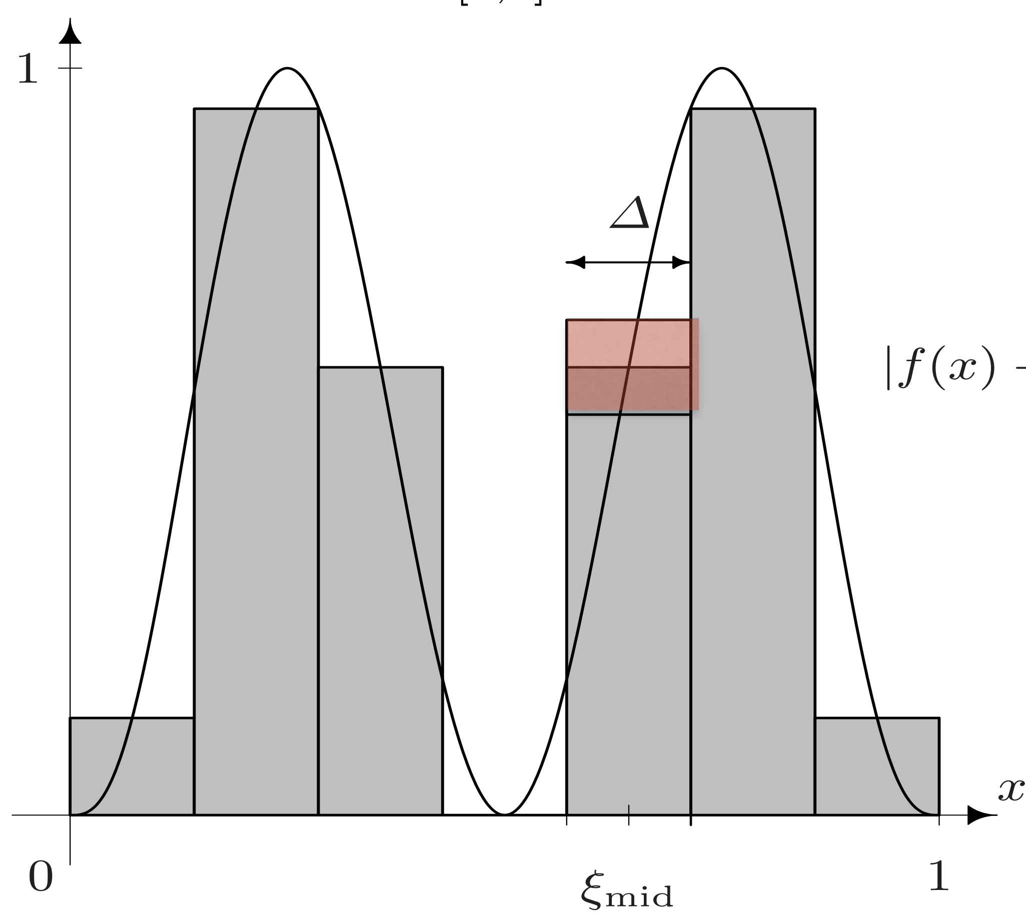
$$I = \int_{\mathbb{X}} f\left(x\right) dx$$

# Computational motivation

Integrals: 1D case $I = \int_{\mathbb{X}} f(x)\, dx$

$$\widehat{I}_n = \frac{1}{n} \sum_{i=0}^{n-1} f\left((i + 1/2)/n\right).$$

$f$ is differentiable and $\sup_{x \in [0,1]} |f'(x)| < M < \infty$ then the approximation error is $\mathcal{O}\left(n^{-1}\right)$




$|f(x) - f(\xi_{\mathrm{mid}})| < \frac{\Delta}{2} \cdot \max |f'(x)|$ for $|x - \xi_{\mathrm{mid}}| \leq \frac{\Delta}{2}$

# Computational motivation

**Integrals: 2D case** $I^{(2)} = \int_{\mathbb{X}} f(x)\,dx$

$$\widehat{I}_n^{(2)} = \frac{1}{n} \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} f\left(\left(i+1/2\right)/n, \left(j+1/2\right)/n\right) \qquad m = \sqrt{n}$$

approximation error is $\mathcal{O}\left(n^{-1/2}\right)$

**Integrals: *d*D case** $I^{(d)} = \int_{\mathbb{X}} f(x)\,dx$

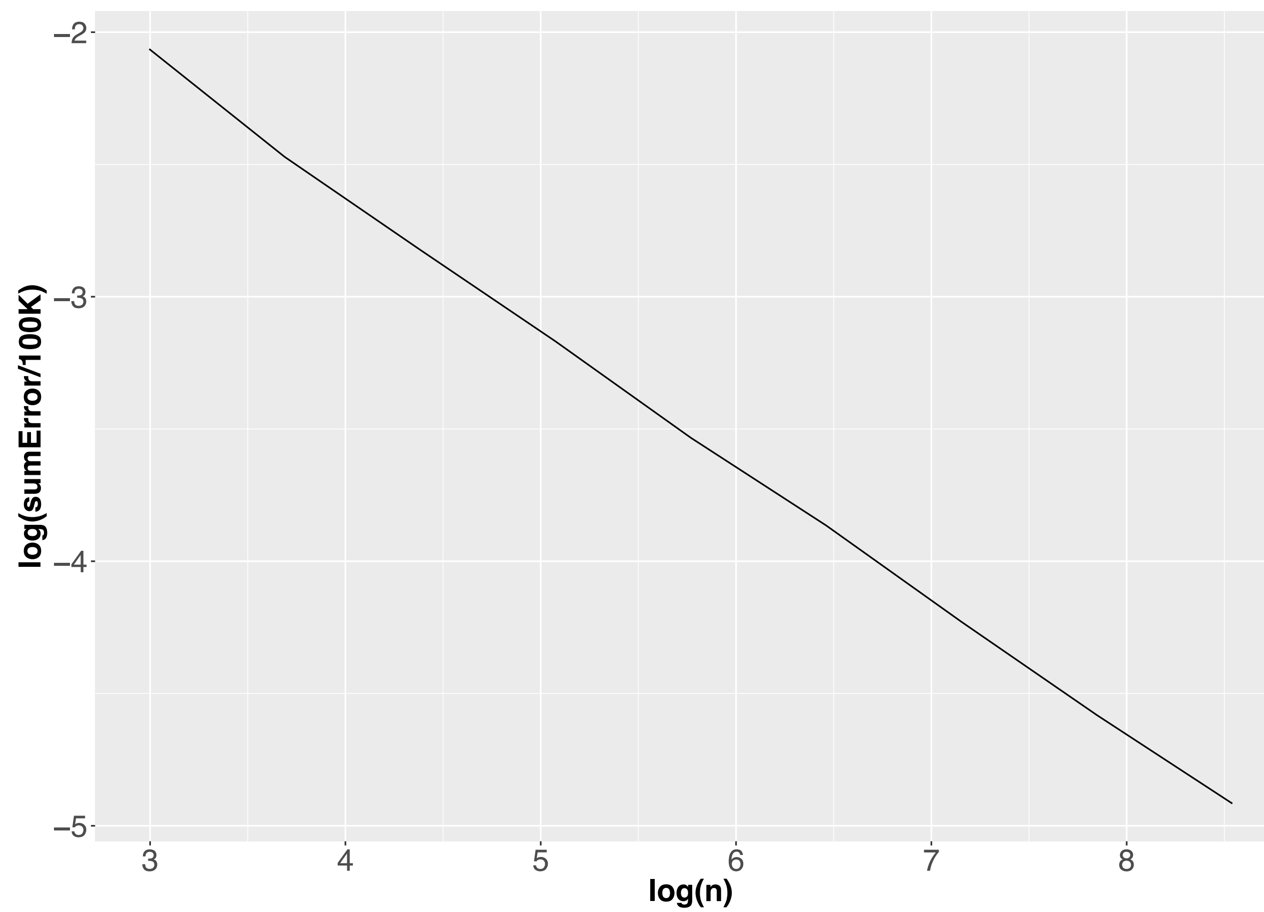
error in $\mathcal{O}\left(n^{-1/d}\right)$

One instance of the ‘curse of dimensionality’

# Interpretation

- Say found the first two decimal for the integral `0.45???` using a naive numerical integration...
  - in 1d, to get one more decimal correct, need 10x more work (2 digits: 100x)
  - in 2d, to get one more decimal correct, need 100x more work (2 digits: 10,000x)
  - in 3d, to get one more decimal correct, need 1000x more work (2 digits: 1e6x)
  - ...

# Enters Simple Monte Carlo...




- Exercise:
  - Use this plot to empirically derive the running time of Simple Monte Carlo for a given tolerence *tol*
  - Create the plot for Example 7

# Theoretical foundations

- Law of large number for iid random variable
- Central limit theorem for iid random variable
- IMPORTANT: there are LLNs and CLTs for dependent random variable (later!)

# Exact sampling methods

# Exact sampling: simple cases

- Inversion method
- Transformation method
- Augmentation method

# Inversion method

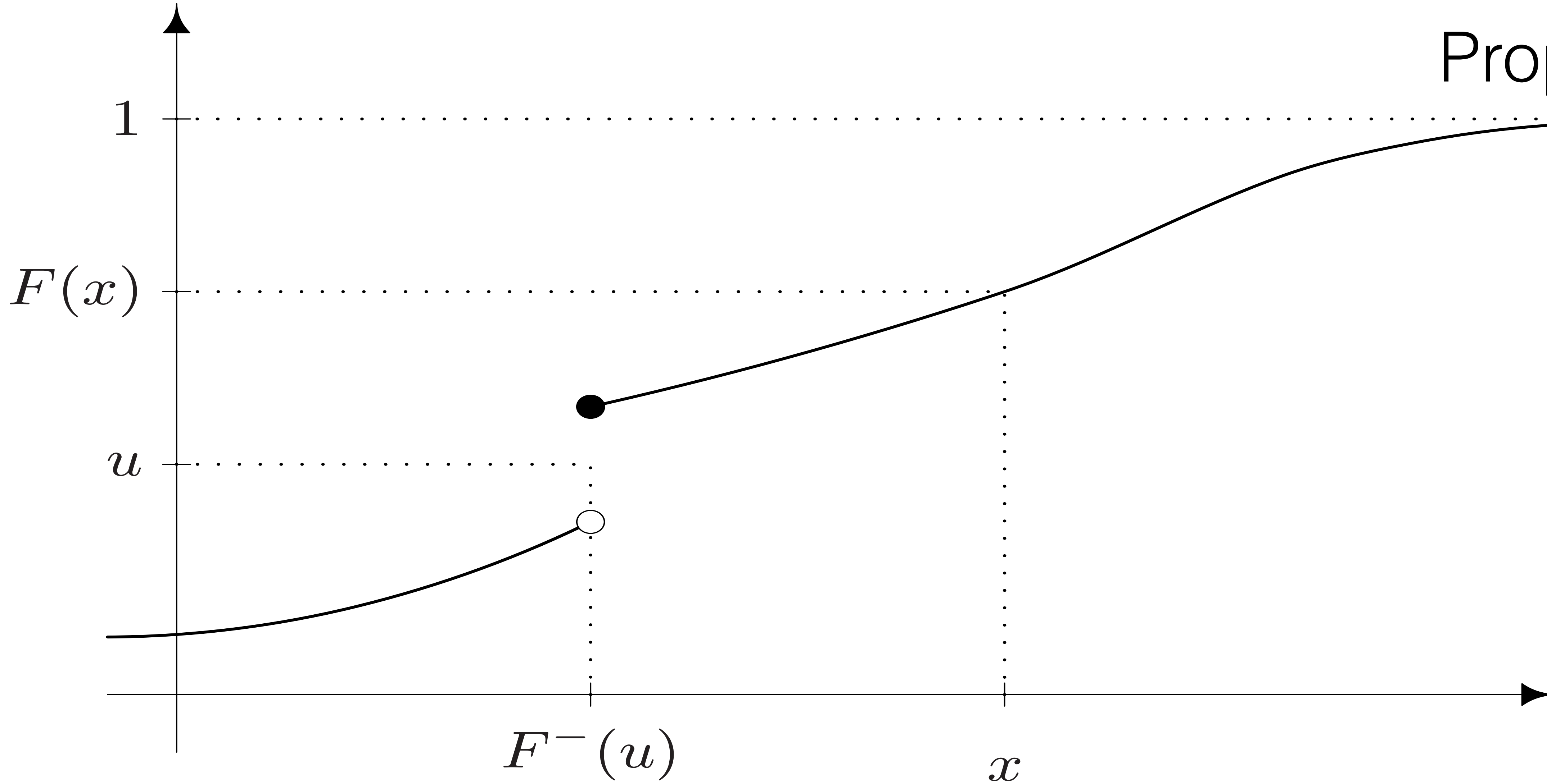
- Consider a real-valued random variable $X$ and its associated cumulative distribution function (cdf)

$$F(x) = \mathbb{P}(X \leq x) = F(x)$$

- The cdf $F : \mathbb{R} \to [0, 1]$ is
  - increasing; i.e. if $x \leq y$ then $F(x) \leq F(y)$
  - right continuous; i.e. $F(x + \epsilon) \to F(x)$ as $\epsilon \to 0$ $(\epsilon > 0)$
  - $F(x) \to 0$ as $x \to -\infty$ and $F(x) \to 1$ as $x \to +\infty$.
- We define the generalised inverse

$$F^{-}(u) = \inf \{x \in \mathbb{R}; F(x) \geq u\}$$

also known as the quantile function. Note that $F^{-}(u) = F^{-1}(u)$ if $F$ is continuous.




- **Proposition**. Let $F$ be a cdf and $U \sim \mathcal{U}_{[0,1]}$. Then $X = F^{-}(U)$ has cdf $F$.

*Proof*. $F^{-}(u) \leq x \Leftrightarrow u \leq F(x)$ so $U \sim \mathcal{U}_{[0,1]}$, we have

$$\mathbb{P}\left(F^{-}(U) \leq x\right) = \mathbb{P}\left(U \leq F(x)\right) = F(x).$$

**Exercise**: construct a RNG for exponential random variables of a given rate

# What about multivariate distributions?

# Transformation method

- Let $Y \sim q$ be a $\mathbb{Y}$-valued random variable (rv) of, which we can simulate (eg, by inversion)
- Let $X \sim \pi$ be a $\mathbb{X}$-valued rv, which we wish to simulate.
- It may be that we can find a function $\varphi : \mathbb{Y} \to \mathbb{X}$ with the property that if we simulate $Y \sim q$ and then set $X = \varphi(Y)$ then we get $X \sim \pi$.
- Inversion is a special case of this idea.

# Augmentation / auxiliary variables

Def. 16

- Example: how to simulate from a mixture model?
- **Key idea**: marginalization is easy with Monte Carlo methods
  - Contrast with analytical marginalization
- Build $(X, Y)$ such that distribution of interest is a marginal
- **Exercises**:
  - write pseudo-code to simulate from a mixture distribution with 2 normal components,
  - ~~show that rejection sampling is an augmentation sampling scheme~~

# Running time of Monte Carlo methods

# Fundamental equation to analyze Monte Carlo methods

running time = $\boxed{\text{number of samples needed to get a tolerance (with probability 95\%)}}$ x $\boxed{\text{compute cost per sample}}$

- Exercise:
  - Compute the running time in *tol* and *d* for Example 7 but with non-diagonal covariance normal vectors

# Using CLT in practice

- Let $X$ be a random tree
- We are looking at a clade indicator $f(X)$ as in Example 2.
- After 500 iid trees your MC estimate for the clade support is roughly 10%
- Should you extract more samples?
  - Say we want a scheme with relative error of less than 10% for approximately 95% of the random seeds