1 – Big Data: The challenge...?
What some people mean
- xxxBytes • Bigger computers • How to interpret?
What does it mean for computational statistics...?
- Develop methodology scalable to large data sizes (N).
  - Increasing N breaks current statistical algorithms.
  - To future-proof we need sub-linear cost with N.
- Reconsider algorithmic design and data ‘sufficiency’.
What is the contribution of our research...
- A statistical algorithm with sub-linear/no cost with N!!

2 – Computational Statistics...?
- We are interested in averages...
  - e.g. Do we expect Conservative/Labour election win?
  - Mathematically: Evaluating expressions of the form, 
    \[ E_p[h(X)] = \int h(x) \cdot p(x) \, dx \]  
- \( h \) is some complicated model we have of reality...
  - e.g. Voting intentions at the 2015 general election.
- \( h \) is some function of our model we are interested in...
  - e.g. How votes translate into seats at Westminster.

3 – Monte Carlo Methods (Existing)
- If \( v \) is intractable (typical scenario), then to evaluate (i) we can instead use representative samples from \( v \) – a so called Monte Carlo Method:
  - Inversion Sampling: Sample uniformly from \( v \)...
  - Rejection Sampling: Sample from something like \( v \) and correct...
  - MCMC: Construct a sample ‘chain’ giving correlated draws from \( v \)...
- Existing Monte Carlo Methods for Big Data...
  - \( v \) becomes more complicated and expensive to sample with large \( N \).
  - Super-linear cost • Inexact • Restrictive model class • Data wasteful

4 – Retrospective Monte Carlo (I)
- What are ‘Retrospective Monte Carlo’ methods...
  - Monte Carlo algorithms with re-ordered steps.
  - Resulting algorithms are equivalent to original.
  - e.g. Is flipping a fair coin and getting ‘Heads’ equivalent to rolling a fair die and getting an ‘Evens’?
- How they differ from regular Monte Carlo...
  - High Deterministic Cost → Low Random Cost.
  - High / Infinite Dimensional → Low Dimensional.
  - Wider applicability to real problems...
  - However, unbiasedness is a key concern...

5 – Retrospective Monte Carlo (II)
- ‘Bernoulli Sampling’ Example: Simulate a coin which lands heads with probability \( p \)...
  - \( p \) is unknown, but we’ve lower and upper convergent bounds (\( a_k \) and \( b_k \))...
- Consider the following two algorithms...
  - Algorithm 1 (Left Picture)...
    - Inefficient (high deterministic cost & biased)!
    - 1) Find \( p \) to high precision (\( \hat{p} \)).
    - 2) Simulate \( u \sim U(0,1) \).
    - 3) ‘Heads’ if \( u \leq \hat{p} \).
  - Algorithm 2 (Right Picture)...
    - Efficient (low random cost & unbiased)!
    - 1) Simulate \( u \sim U(0,1) \).
    - 2) ‘Heads’ if \( u < a_k \) & \( p \) and ‘Tails’ if \( u > b_k \) & \( p \).

6 – Diffusions & Exact Algorithms
- Some real world processes are best modelled using continuously evolving dynamics...
  - e.g. biological processes, climatic processes, stock market fluctuations...
- A diffusion is an infinite dimensional stochastic process capturing this behaviour,
  \[ \text{Diffusion} \]
  \[ \mu(X) \]
  \[ \sigma(X) \]
  \[ \text{Brownian Motion} \]
  \[ dx = \mu(x) \, dt + \sigma(x) \, db \]
  \[ \text{Function of Diffusions} \]
- What is an Exact Algorithm...? A retrospective Monte Carlo method enabling the simulation of (jump) diffusion sample path skeletons with no error (see pictures).

7 – ScaLE: Scalable Langevin Exact Algorithm
- Key Idea: The Langevin Diffusion has the desirable property that sample paths simulated from it can be used as samples from our model of reality if
- Existing methods use approximate solutions from Langevin, and either:
  - Don’t correct: samples not from \( v \) • Correct: computationally expensive with \( N \).
- We have developed theory and methodology to simulate from Langevin exactly!
  - \( \log v \) drift is amenable to unbiased estimation, and so subsampling is viable!
  - Resulting approach has sub-linear (or no) cost with \( N \) and we can sample \( v \)!