Approximate Inference

- In more realistic scenarios, more complex models are required, and the marginal likelihood is usually intractable, thus the posterior cannot be solved analytically
- we have to approximate the posterior distribution $p(y \mid \theta)$ by:
    - Simulation: generate a random sample to approximate the posterior empirically
    - Distributional approximation: approximate the posterior directly by some simpler parametric distribution (e.g. Normal)
- We will focus on approximating the posterior distribution by generating random samples

Grid Sampling (Grid Approximation, Discrete Approximation)

- Create an even-spaced grid: $g_1 = a + i/2, \ldots, g_m = b - i/2$ where a is the lower, and b is the upper limit of the interval on which we want to evaluate the posterior, i is the increment of the grid, and m is the number of grid points.
- Evaluate values of the unnormalized posterior density in the grid points $q(g_1; \mathbf{y}), \ldots, q(g_m; \mathbf{y})$, and normalized them to obtain the estimated values of the posterior distribution at the grid points:
  - $\hat{p}_1 := \frac{q(g_1;\mathbf{y})}{\sum_{i=1}^{m} q(g_i;\mathbf{y})}, \ldots, \hat{p}_m := \frac{q(g_m;\mathbf{y})}{\sum_{i=1}^{m} q(g_i;\mathbf{y})}$
- For every $s = 1, \ldots, S$:
  - Generate λ_s from a categorical distribution with outcomes $g_1, \ldots, g_m$ which have the probability $\hat{p}_1, \ldots, \hat{p}_n$
  - Add jitter which is uniformly distributed around zero, and whose interval length is equal to the grid spacing, to the generated values: $\lambda_s = \lambda_s + X$, where $X \sim U(-i/2, i/2)$ to push generated values out of the grid points.

Grid Sampling and Curse of Dimensionality

- 10 parameters
- if we don't know beforehand where the posterior mass is
  - need to choose wide box for the grid
  - need to have enough grid points to sample essential mass
- e.g. 50 or 1000 grid points per dimension
  - $\rightarrow$ $50^{10} \approx$ 1e17 grid points
  - $\rightarrow$ $1000^{10} \approx$ 1e30 grid points
- Suppose a laptop can compute density of normal distribution about 20 million times per second
  - $\rightarrow$ evaluation in 1e17 grid points would take 150 years
  - $\rightarrow$ evaluation in 1e30 grid points would take 1 500 billion years

How to Generate Random Numbers?

- Pseudo Random Number Generator(PRNG) refers to an algorithm that uses mathematical formulas to produce sequences of random numbers.
- PRNGs generate a sequence of numbers approximating the properties of random numbers.(They would pass various statistical tests for checking the random/independent property, thus sufficient for Bayesian inference)
  - Linear congruential generater: $X_{n+1} = (aX_n + c)\ mod\ m$
  - $a = 1103515245$, $c = 12345$, $m = 2^{31}$, X_0 is the seed

Direct Sampling

- Produces independent draws
  - Using analytic transformations of uniform random numbers (e.g. Polar method for normal random variable)
    If U_1 and U_2 are independent draws from distribution $\mathrm{U}(0,1)$, and

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

    then X_1 and X_2 are independent draws from the distribution $\mathrm{N}(0,1)$

Direct Sampling

- Produces independent draws
  - Using analytic transformations of uniform random numbers (e.g. Polar method for normal random variable)
    If U_1 and U_2 are independent draws from distribution $\mathrm{U}(0,1)$, and

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

    then X_1 and X_2 are independent draws from the distribution $\mathrm{N}(0,1)$
  - Inverse-CDF
    A continuous CDF, F, is a one-to-one mapping of the domain of the CDF into the interval (0,1)
    Lemma: if $U \sim Unif(0,1)$, then $X = F^{-1}(U)$ is a simulation from $f(x)$

Direct Sampling

- Produces independent draws
  - Using analytic transformations of uniform random numbers (e.g. Polar method for normal random variable)
    If U_1 and U_2 are independent draws from distribution $\mathrm{U}(0,1)$, and

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$$
$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

    then X_1 and X_2 are independent draws from the distribution $\mathrm{N}(0,1)$
  - Inverse-CDF
    A continuous CDF, F, is a one-to-one mapping of the domain of the CDF into the interval (0,1)
    Lemma: if $U \sim Unif(0,1)$, then $X = F^{-1}(U)$ is a simulation from $f(x)$
- Still restricted to limited set of models

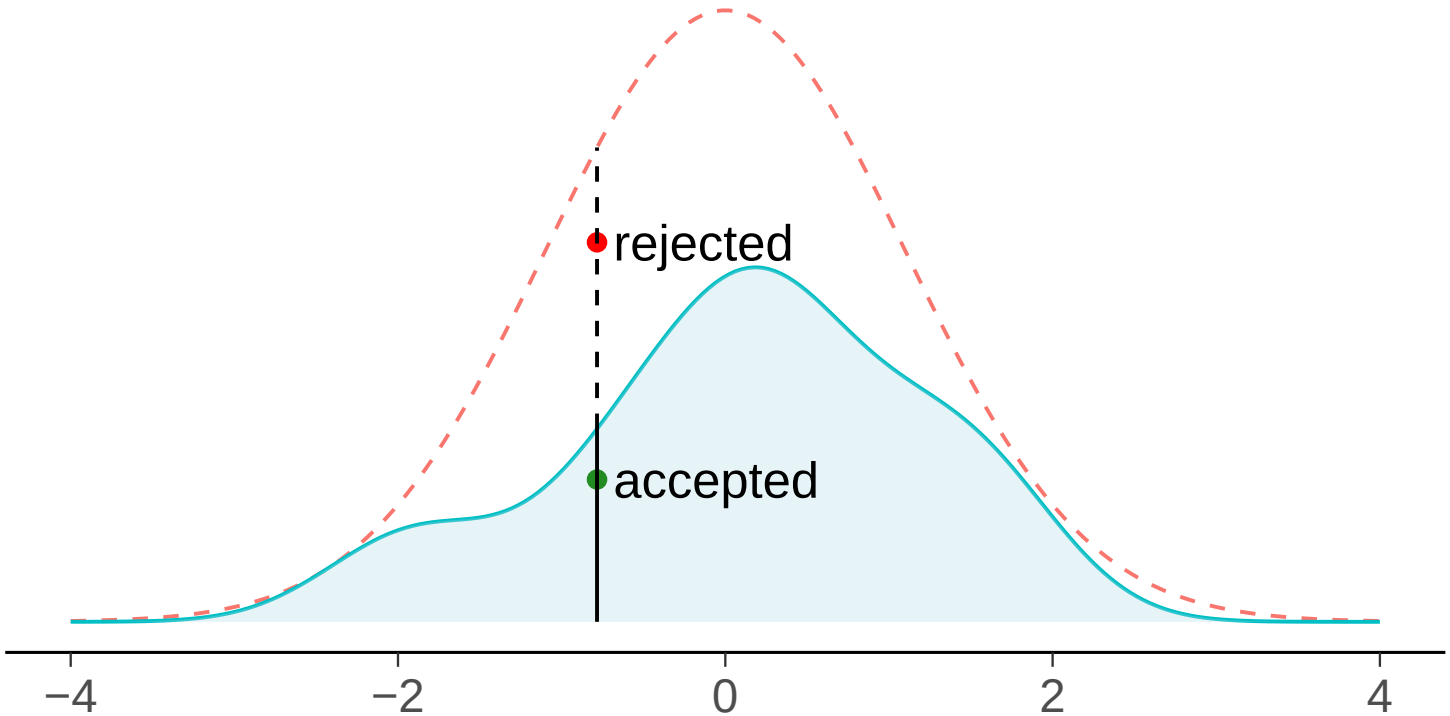
Indirect Sampling

- Rejection sampling
- Importance sampling
- Markov chain Monte Carlo

Rejection Sampling

Draw directly from a proposal distribution, reject some draws, remaining draws are independent draws from the target distribution

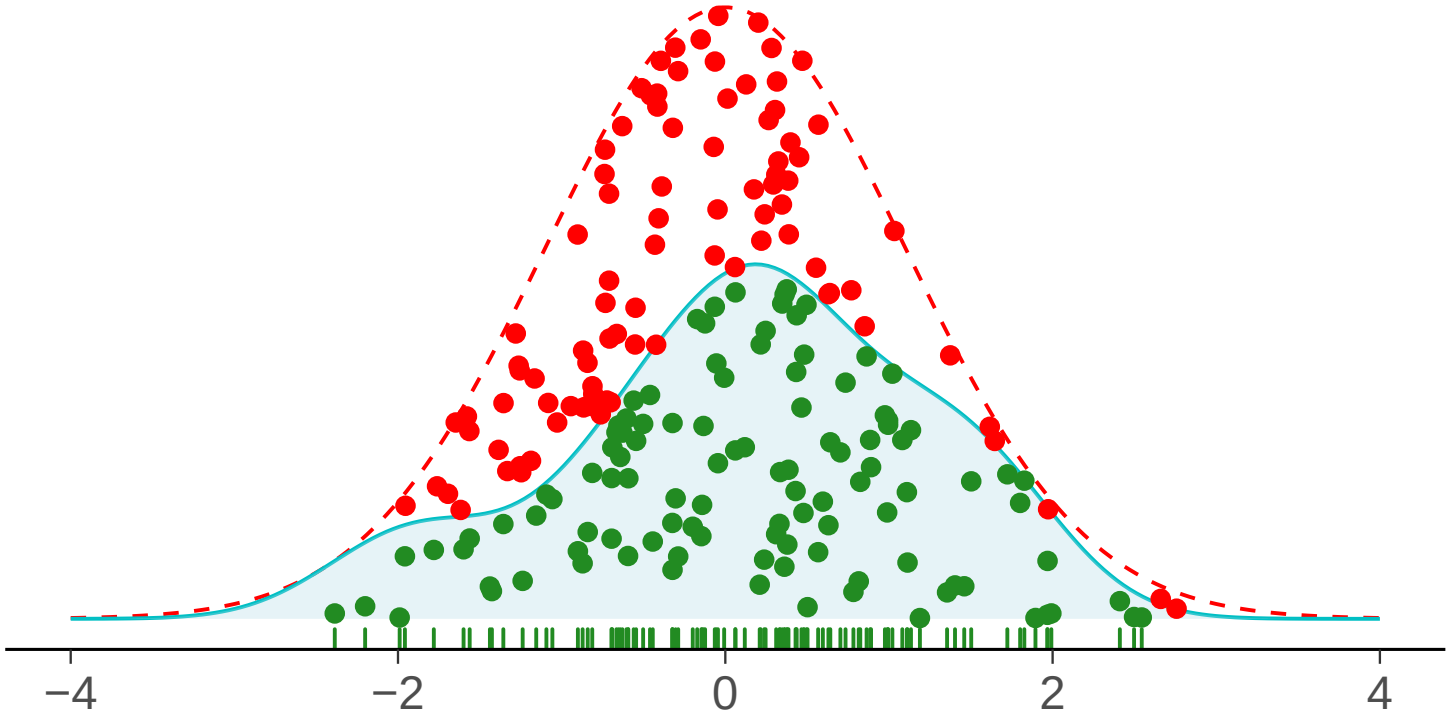
- Proposal forms envelope over the target distribution $q(\theta|y)/Mg(\theta) \leq 1$
- Draw from the proposal and accept with probability $q(\theta|y)/Mg(\theta)$



Rejection Sampling

Draw directly from a proposal distribution, reject some draws, remaining draws are independent draws from the target distribution

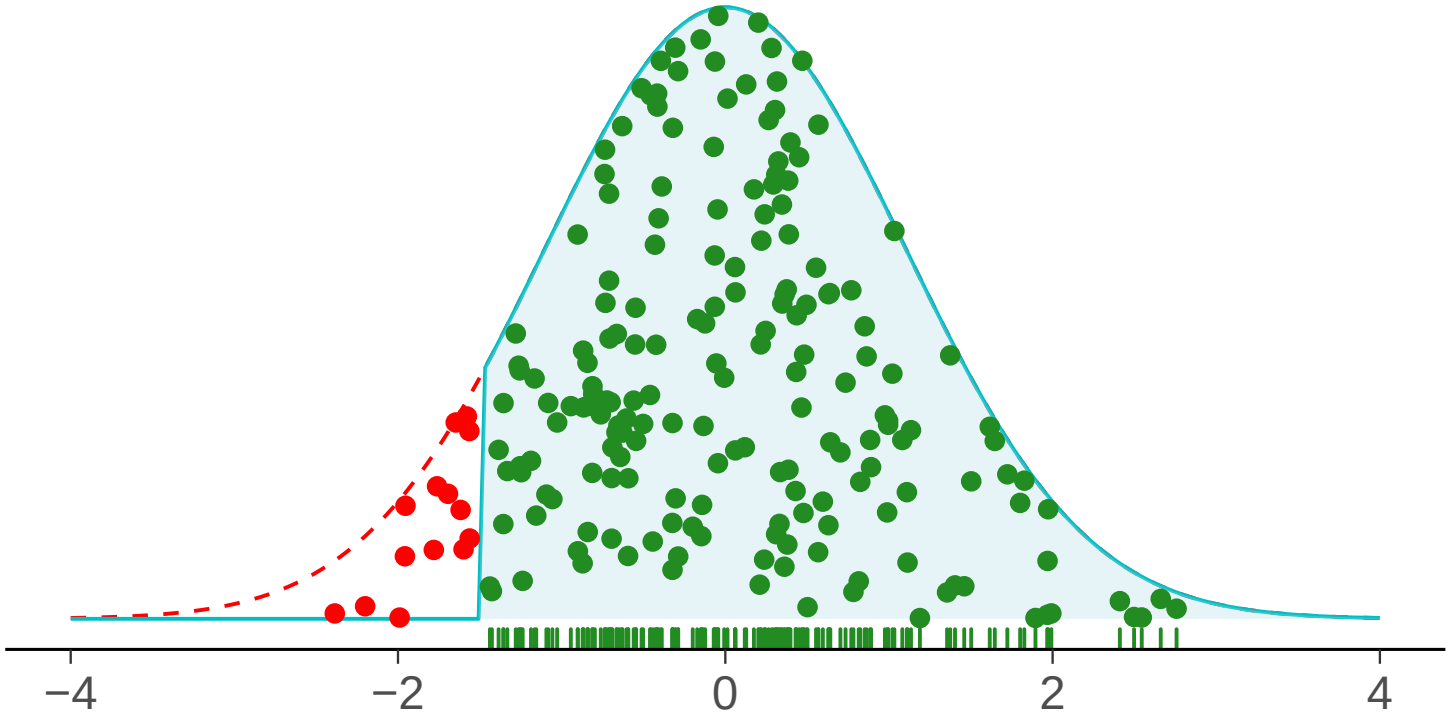
- Proposal forms envelope over the target distribution $q(\theta|y)/Mg(\theta) \leq 1$
- Draw from the proposal and accept with probability $q(\theta|y)/Mg(\theta)$



Rejection Sampling

Draw directly from a proposal distribution, reject some draws, remaining draws are independent draws from the target distribution

- Proposal forms envelope over the target distribution $q(\theta|y)/Mg(\theta) \leq 1$
- Draw from the proposal and accept with probability $q(\theta|y)/Mg(\theta)$
- Common for truncated distributions



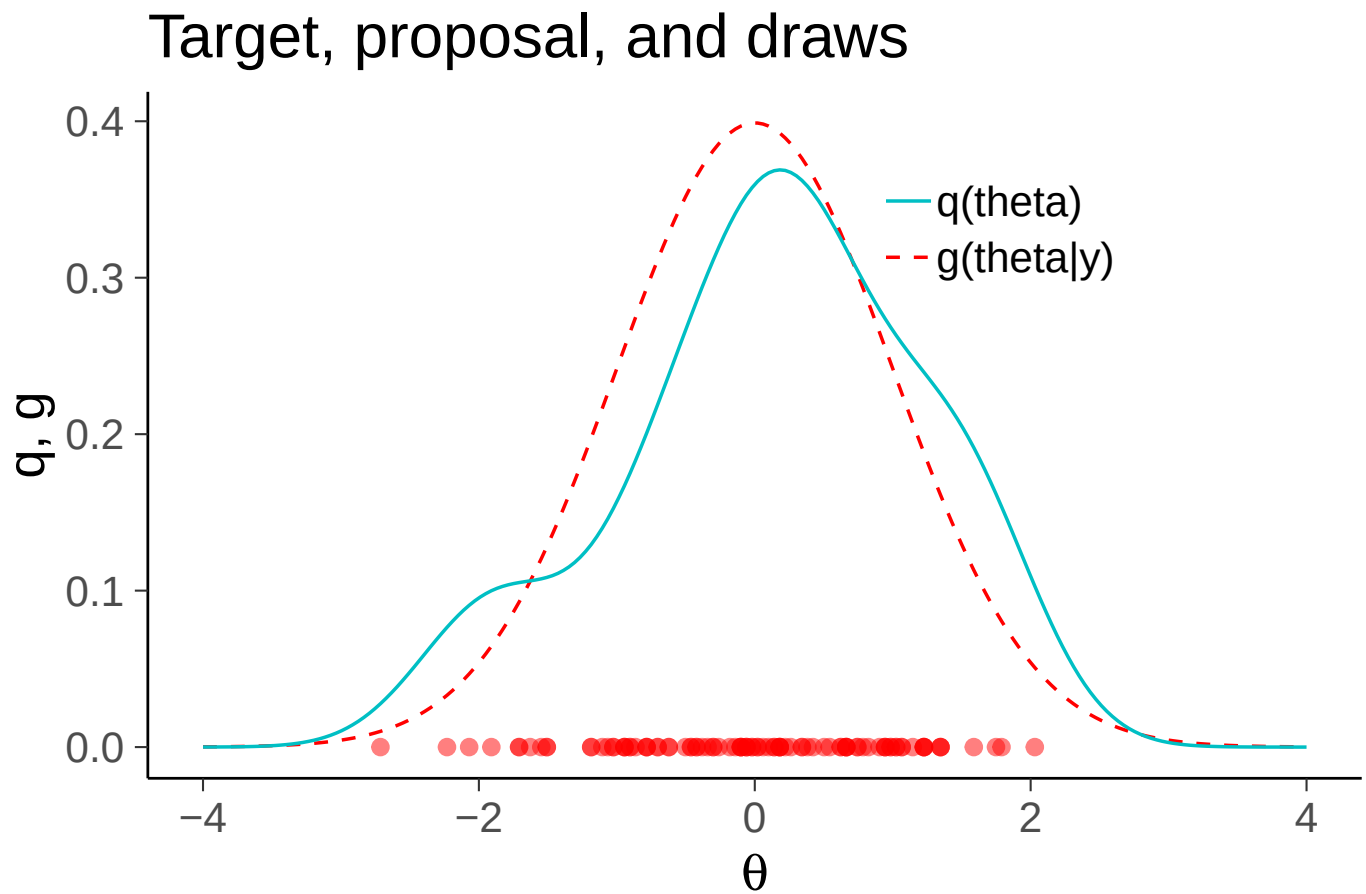
Rejection Sampling

- The number of accepted draws is the effective sample size
  - with bad proposal distribution may require a lot of trials
  - selection of good proposal gets very difficult when the number of dimensions increase

Importance Sampling

Draw samples directly from a proposal distribution, then weigh the draws

- Proposal does not need to have a higher value everywhere

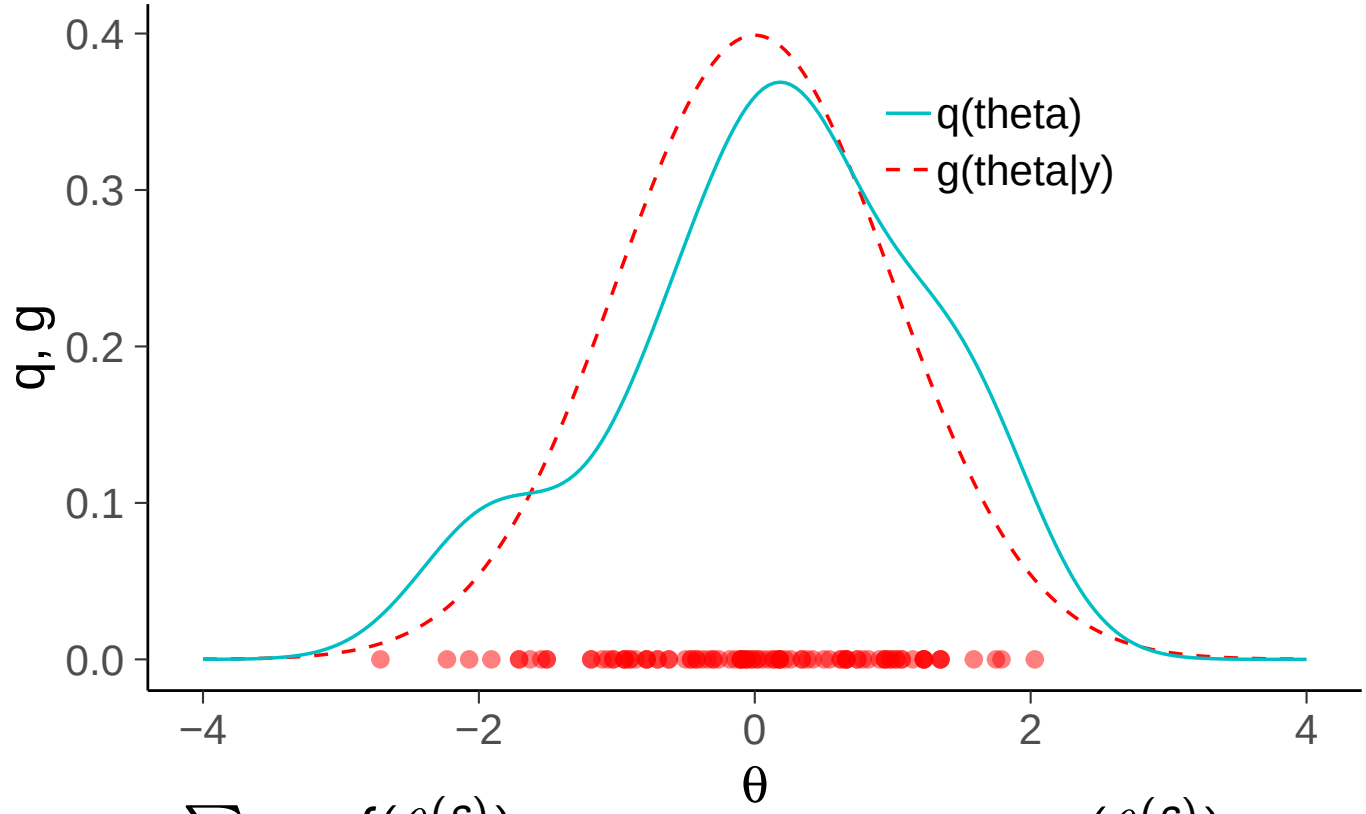


Importance Sampling

Draw samples directly from a proposal distribution, then weigh the draws

- Proposal does not need to have a higher value everywhere

Target, proposal, and draws



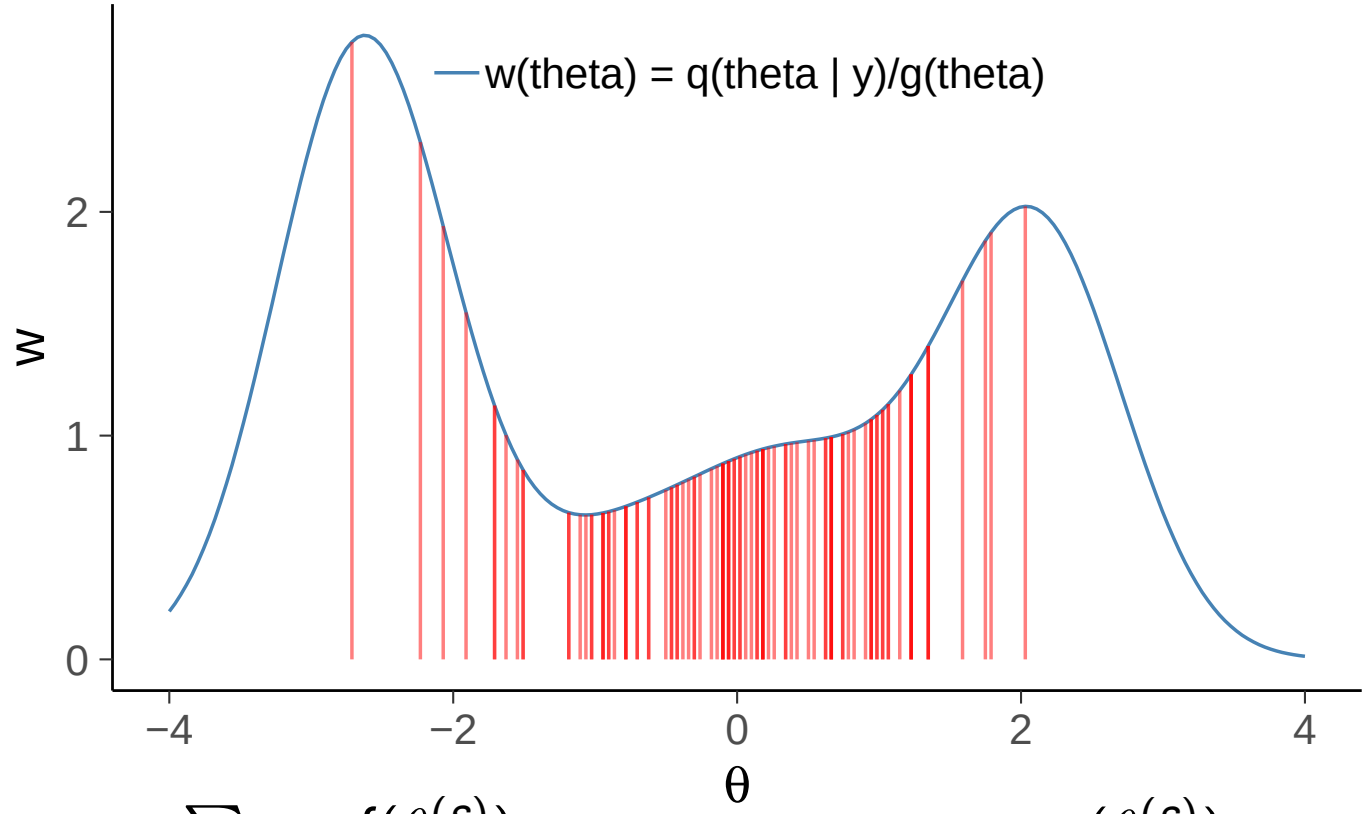
$$\mathrm{E}[f(\theta)] \approx \frac{\sum_s w_s f(\theta^{(s)})}{\sum_s w_s}, \qquad \text{where} \quad w_s = \frac{q(\theta^{(s)})}{g(\theta^{(s)})}$$

Importance Sampling

Draw samples directly from a proposal distribution, then weigh the draws

- Proposal does not need to have a higher value everywhere

Draws and importance weights



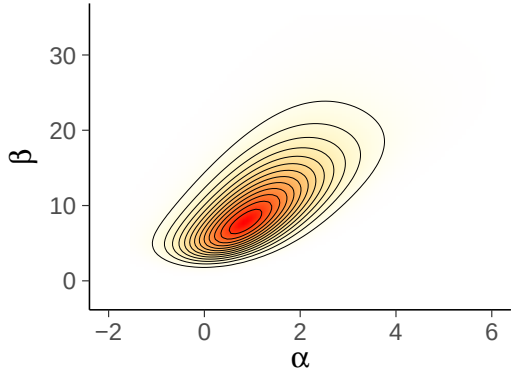
$$\mathrm{E}[f(\theta)] \approx \frac{\sum_s w_s f(\theta^{(s)})}{\sum_s w_s}, \qquad \text{where} \quad w_s = \frac{q(\theta^{(s)})}{g(\theta^{(s)})}$$

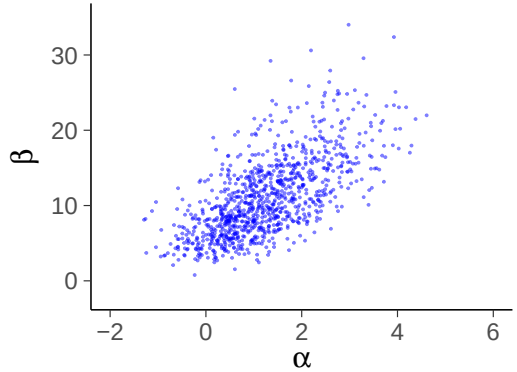
Importance Sampling

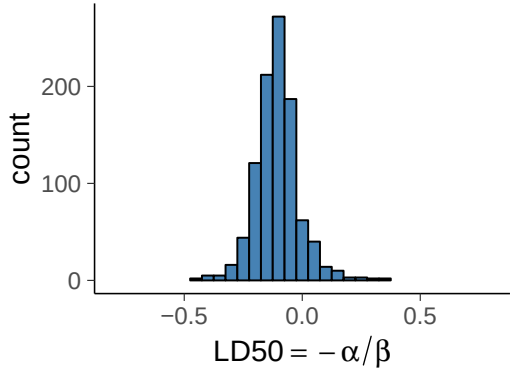
- Resampling using normalized importance weights can be used to pick a smaller number of draws with uniform weights
- Selection of good proposal gets more difficult when the number of dimensions increase
- Often used to correct distributional approximations

Example: Importance sampling in Bioassay

Grid

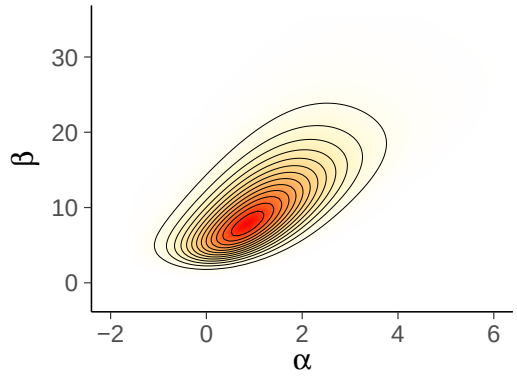


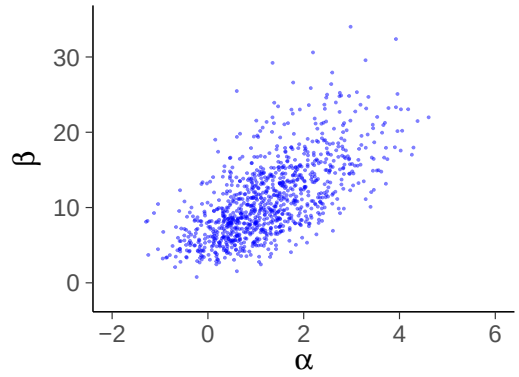


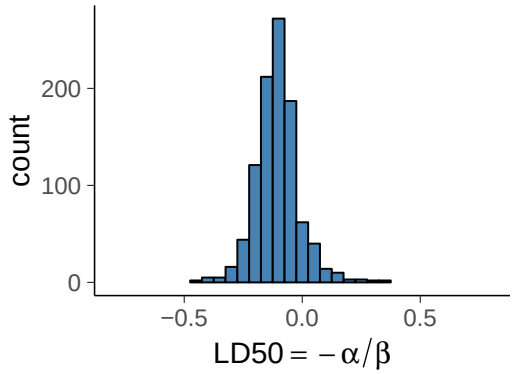


Example: Importance sampling in Bioassay

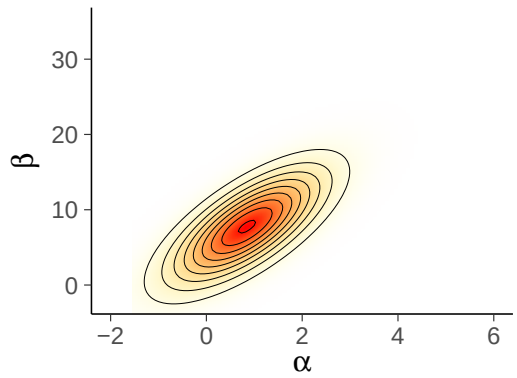
Grid

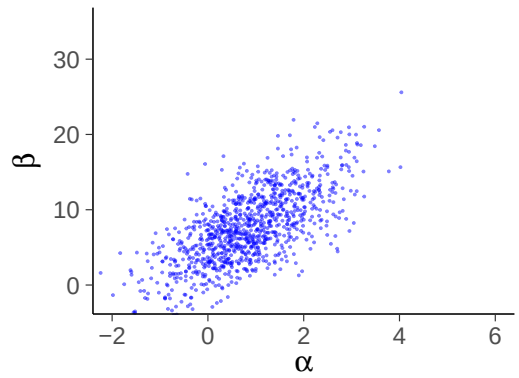


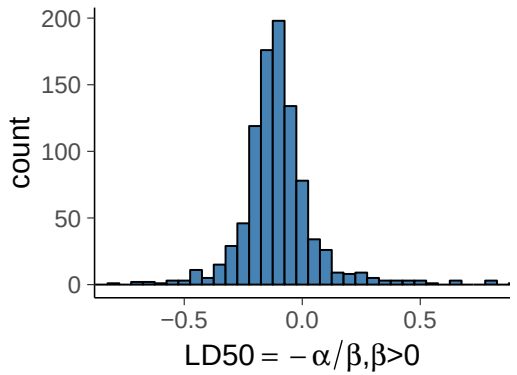




Normal

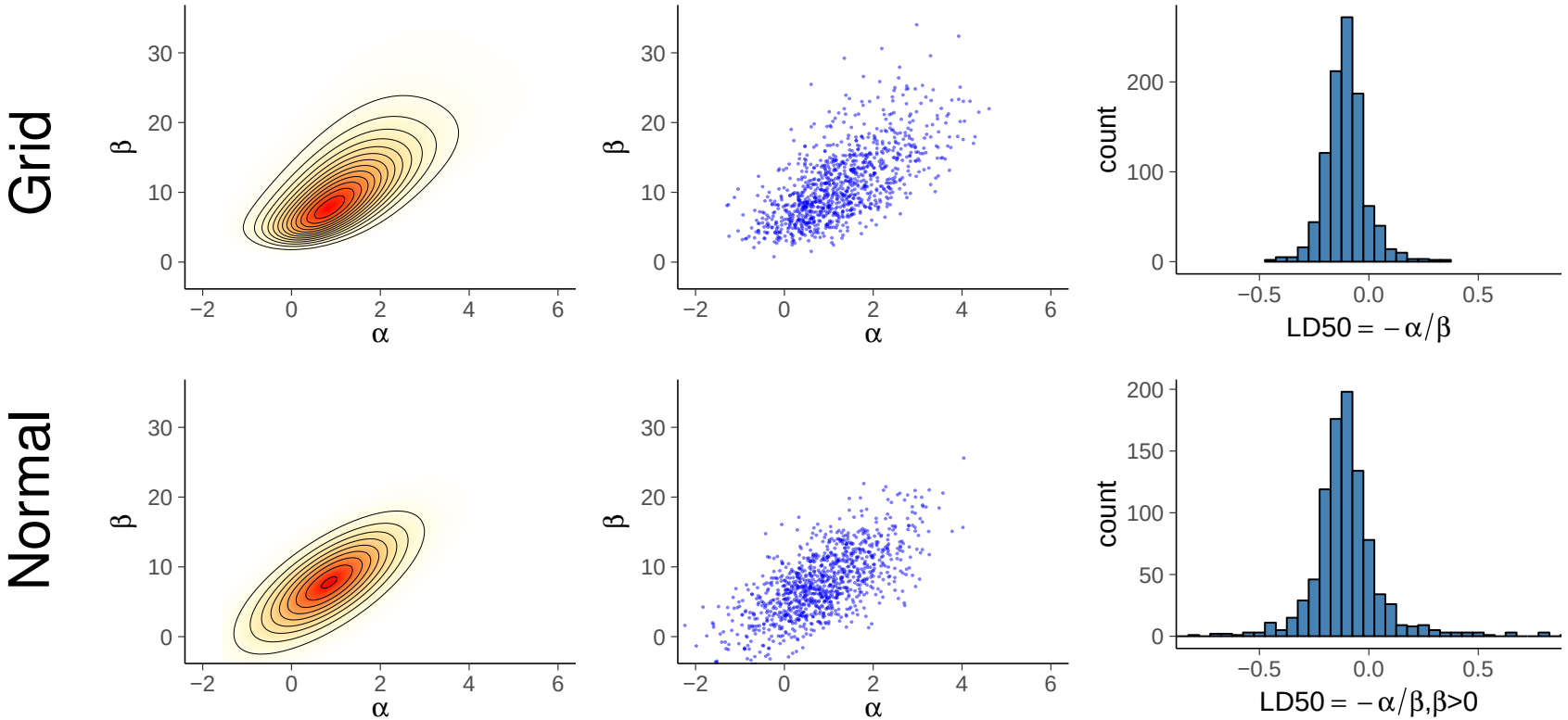






Normal approximation is discussed more in BDA3 Ch 4

Example: Importance sampling in Bioassay



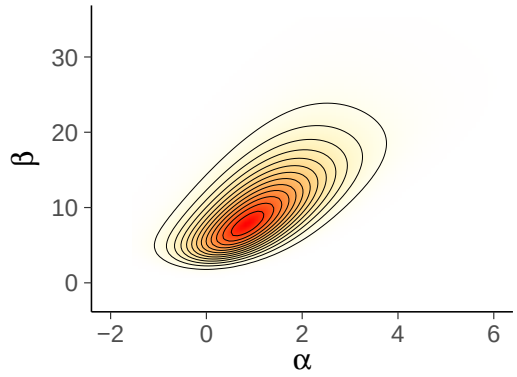
Normal approximation is discussed more in BDA3 Ch 4

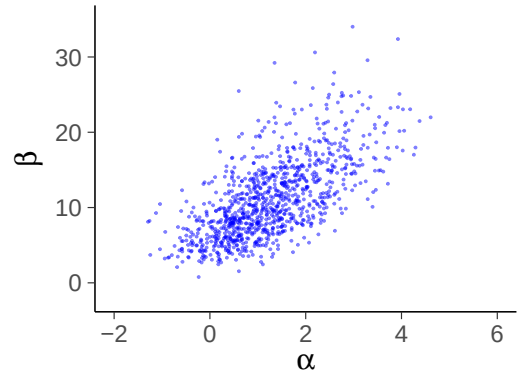
But the normal approximation is not that good here:

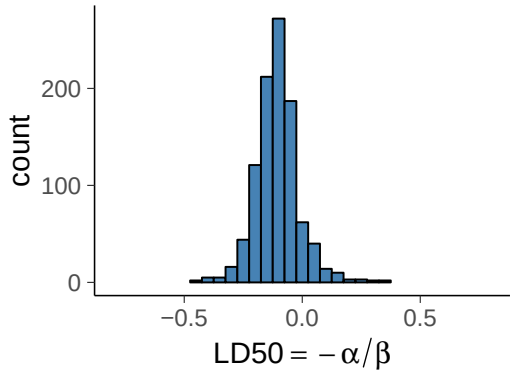
Grid sd(LD50) $\approx$ 0.1, Normal sd(LD50) $\approx$ .75!

Example: Importance sampling in Bioassay

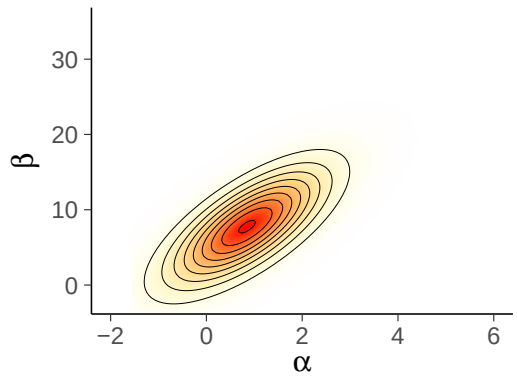
Grid

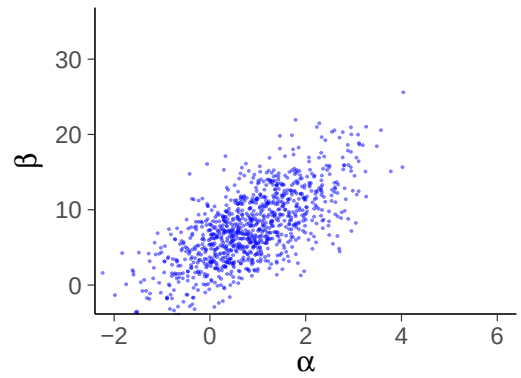


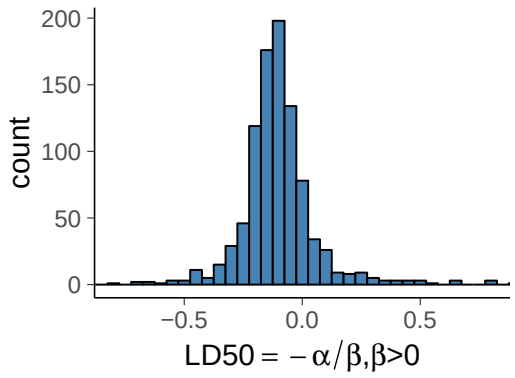




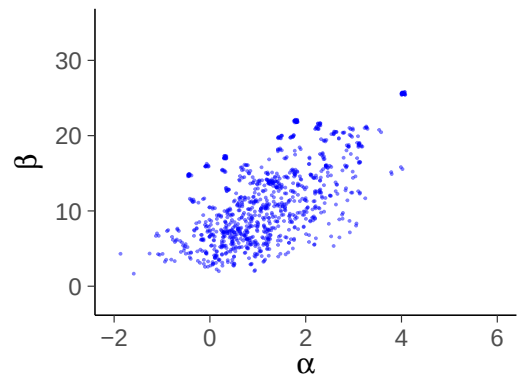
Normal

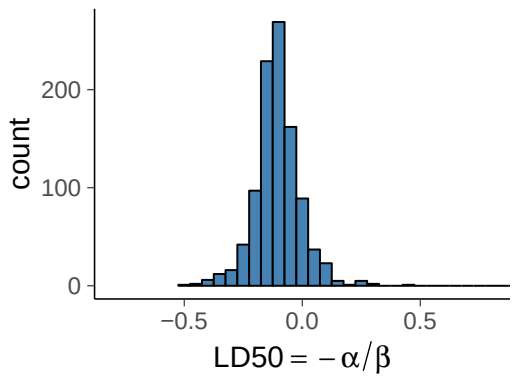






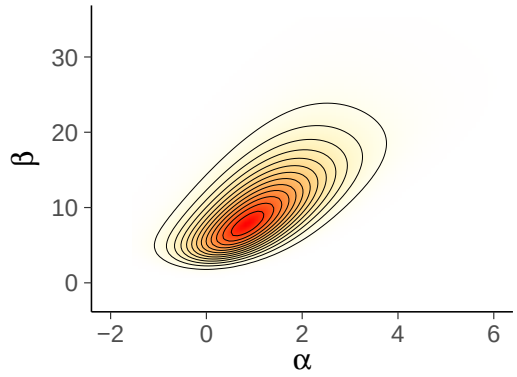
IR

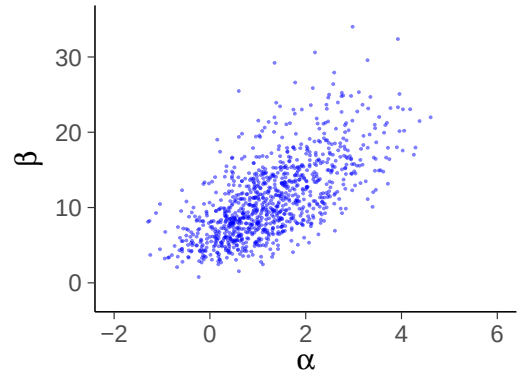


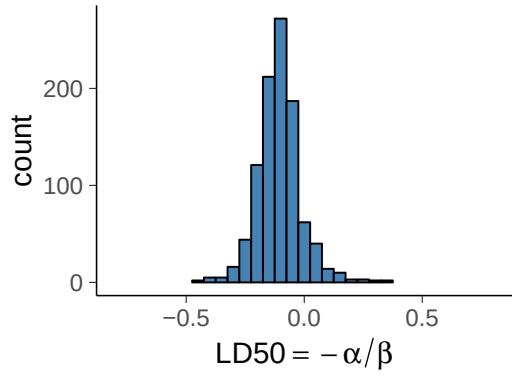


Example: Importance sampling in Bioassay

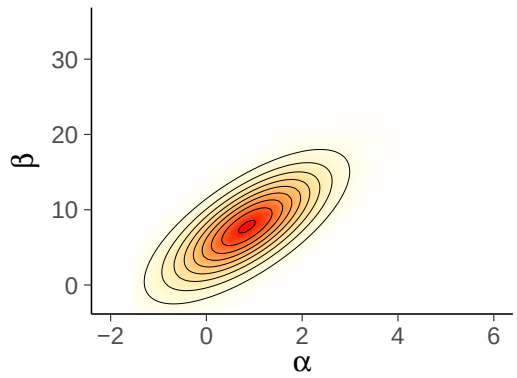
Grid

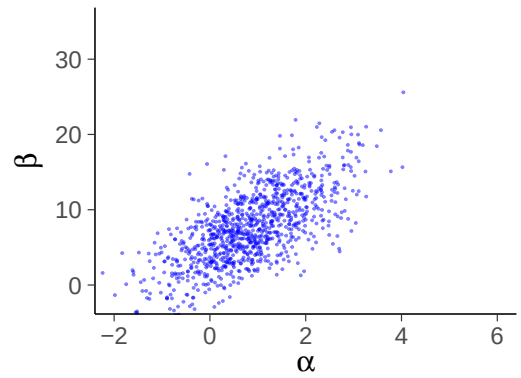


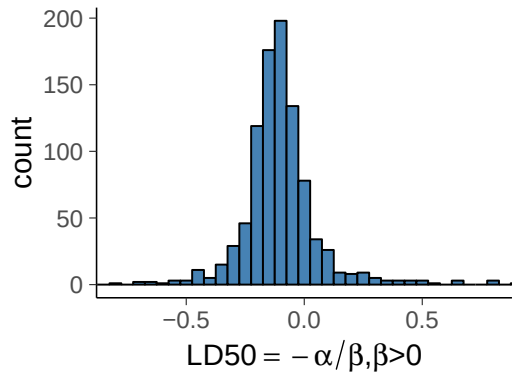




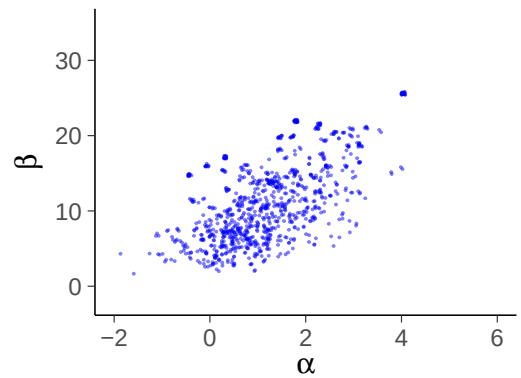
Normal

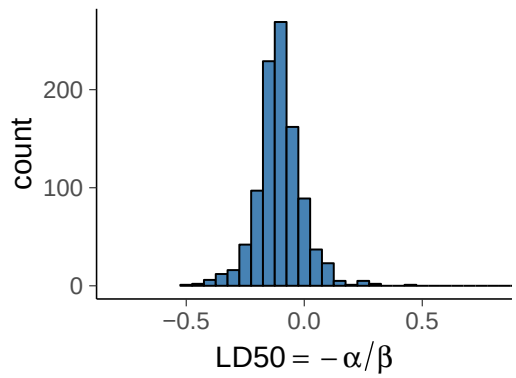






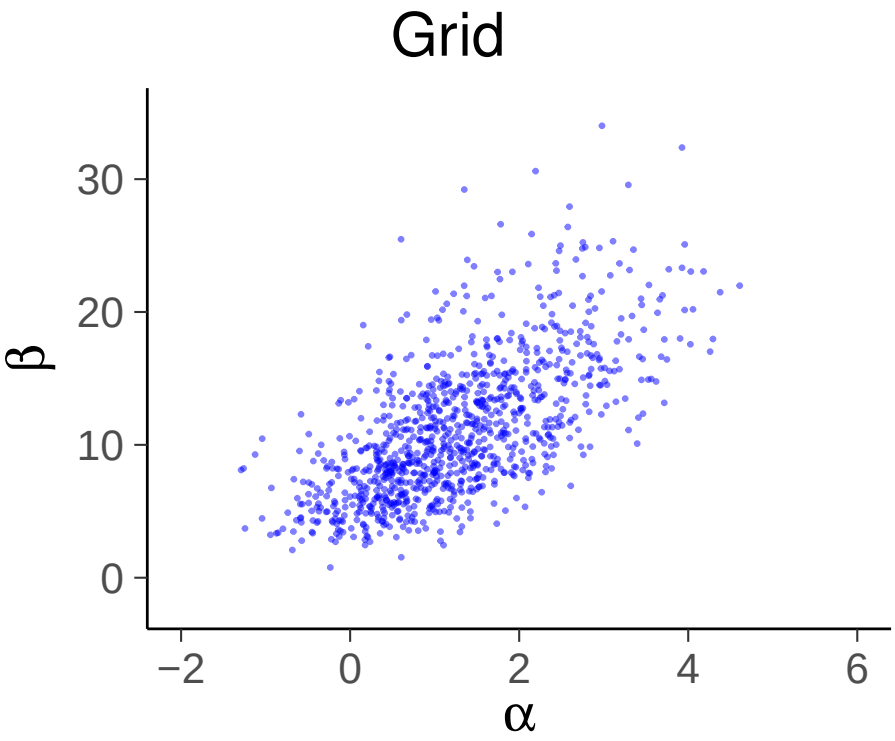
IR

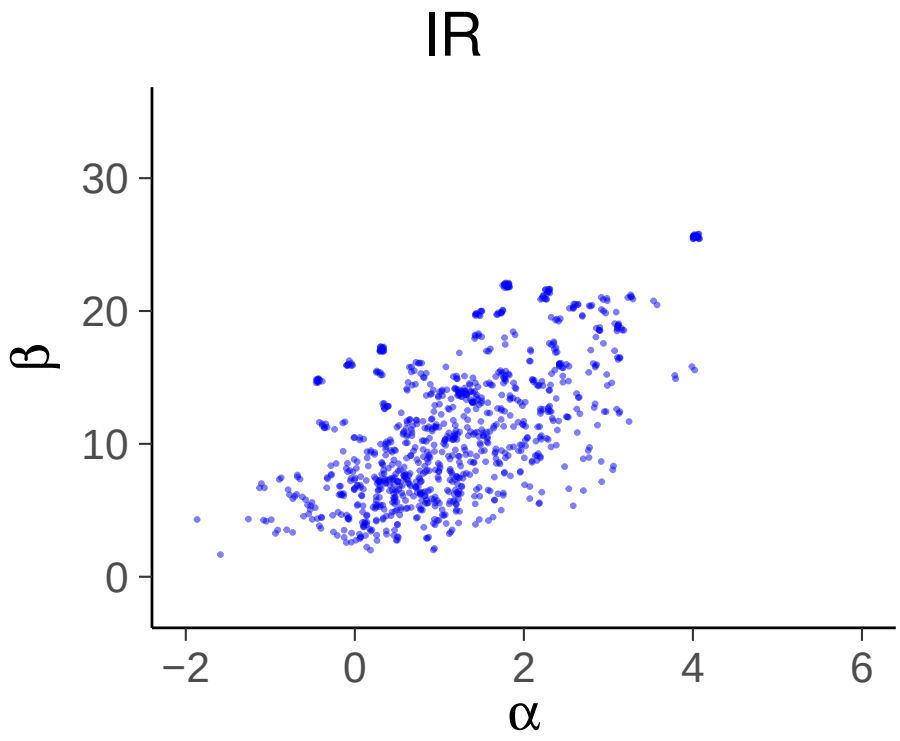




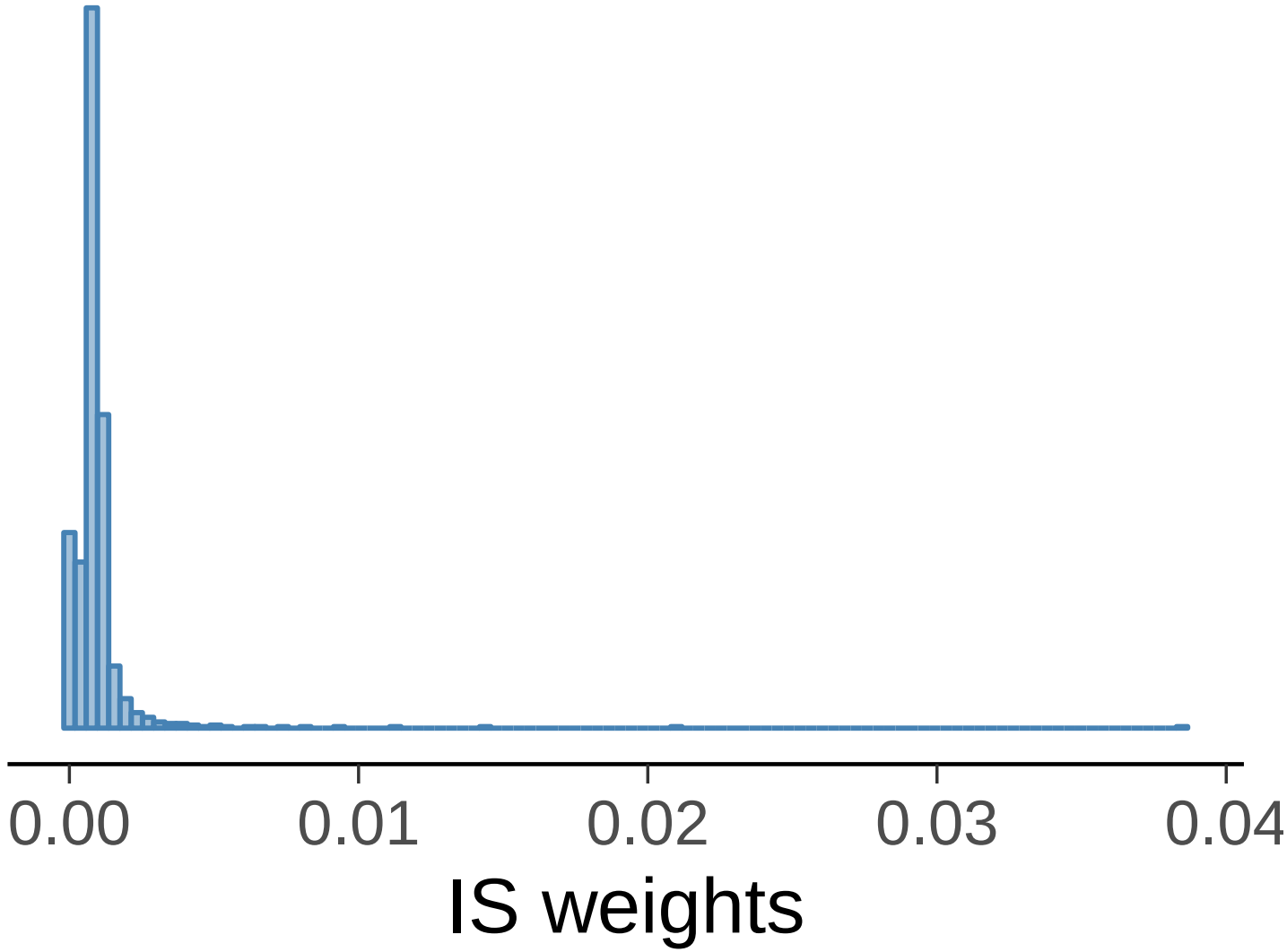
Grid sd(LD50) $\approx$ 0.1, IR sd(LD50) $\approx$ 0.1

Example: Importance sampling in Bioassay

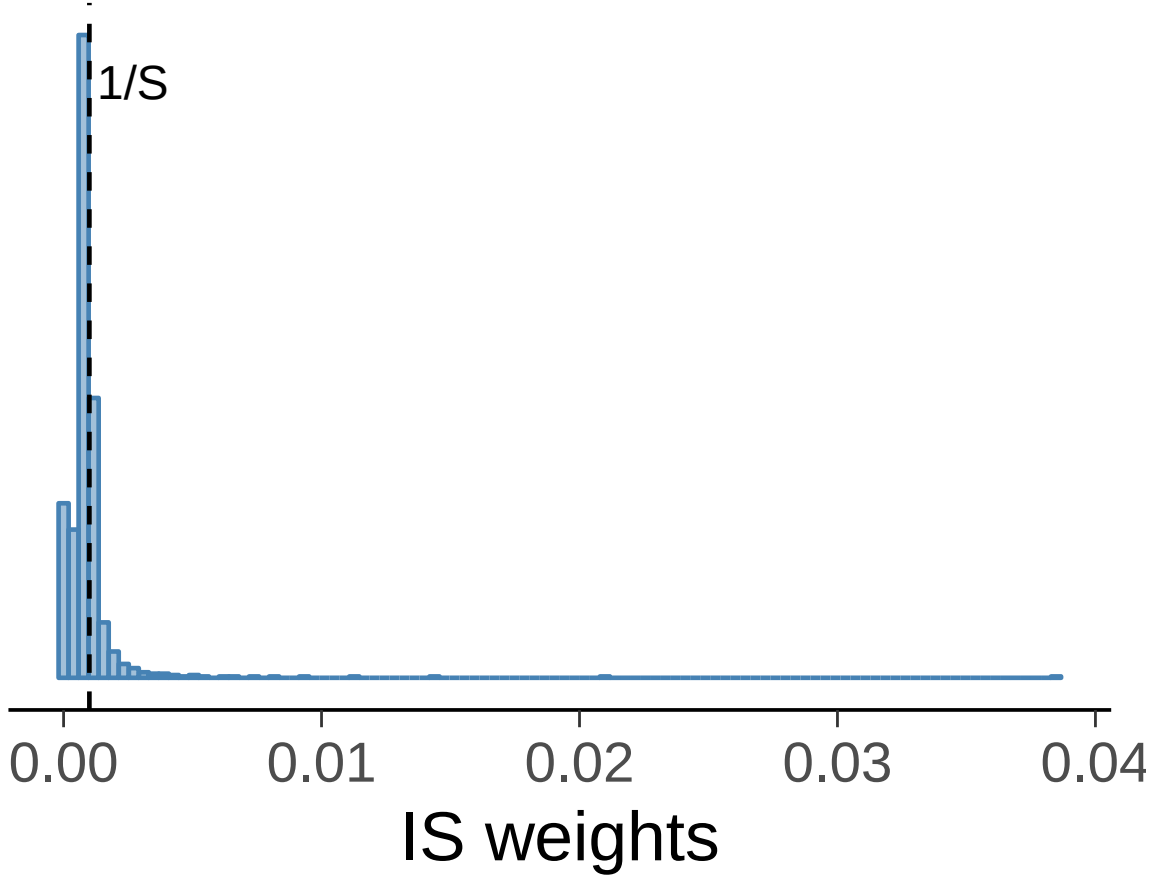




Example: Importance sampling in Bioassay



Example: Importance sampling in Bioassay



$$S_{\text{eff}} = \frac{1}{\sum_{s=1}^{S} (\tilde{w}(\theta^s))^2}, \quad \text{is based on variance of } \tilde{w}(\theta^s)$$

$S_{\text{eff}} \approx 270$

Importance sampling leave-one-out cross-validation

- Later in the course you will learn how $p(\theta|y)$ can be used as a proposal distribution for $p(\theta|y_{-i})$
  - which allows fast computation of leave-one-out cross-validation

$$p(y_i|y_{-i}) = \int p(y_i|\theta)p(\theta|y_{-i})d\theta$$

Monte Carlo - History

Computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters.

- Used already before computers
  - Buffon (18th century; needles)
  - De Forest, Darwin, Galton (19th century)
  - Pearson (19th century; roulette)
  - Gosset (Student, 1908; hat)

Monte Carlo - History

Computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters.

- Used already before computers
  - Buffon (18th century; needles)
  - De Forest, Darwin, Galton (19th century)
  - Pearson (19th century; roulette)
  - Gosset (Student, 1908; hat)
- ”Monte Carlo method” term was proposed by Metropolis, von Neumann and Ulam in the end of 1940s
  - they worked together in atomic bomb project
  - Metropolis and Ulam, ”The Monte Carlo Method”, 1949

Monte Carlo - History

Computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters.

- Used already before computers
  - Buffon (18th century; needles)
  - De Forest, Darwin, Galton (19th century)
  - Pearson (19th century; roulette)
  - Gosset (Student, 1908; hat)
- ”Monte Carlo method” term was proposed by Metropolis, von Neumann and Ulam in the end of 1940s
  - they worked together in atomic bomb project
  - Metropolis and Ulam, ”The Monte Carlo Method”, 1949
- Cheaper computation became available in 1990s
  - BUGS project started 1989 (last OpenBUGS release 2014)
  - Gelfand & Smith, 1990
  - Stan initial release 2012

Monte Carlo

- Simulate draws from the target distribution
  - these draws can be treated as any observations
  - a collection of draws is sample
- Use these draws, for example,
  - to compute means, deviations, quantiles
  - to draw histograms
  - to marginalize
  - etc.

Monte Carlo vs. Deterministic

- Monte Carlo = simulation methods
  - Samples are selected stochastically (randomly)
- Deterministic methods (e.g. grid)
  - Evaluation points are selected by some deterministic rule
  - Proper deterministic methods converge faster (common for low dimensions)

Markov chain Monte Carlo (MCMC)

- Pros
  - Markov chain goes where most of the posterior mass is
  - Certain MCMC methods scale well to high dimensions
- Cons
  - Draws are dependent (affects how many draws are needed)
  - Convergence in practical time is not guaranteed
- MCMC methods in this course
  - Gibbs: “iterative conditional sampling”
  - Metropolis: “random walk in joint distribution”
  - Dynamic Hamiltonian Monte Carlo: “state-of-the-art” used in Stan

How many simulation draws are needed?

- How many draws or how big sample size?
- If draws are independent
  - usual methods to estimate the uncertainty due to a finite number of observations (finite sample size)
- Markov chain Monte Carlo produces dependent draws
  - requires additional work to estimate the effective sample size