Stan:
Probabilistic Modeling & Bayesian Inference

Development Team

Andrew Gelman, **Bob Carpenter**, Daniel Lee, Ben Goodrich,
Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Allen Riddell,
Marco Inacio, Jeffrey Arnold, **Mitzi Morris**, Rob Trangucci,
Rob Goedman, Brian Lau, Jonah Sol Gabry, Robert L. Grant,
Krzysztof Sakrejda, Aki Vehtari, Rayleigh Lei, Sebastian Weber,
Charles Margossian, Vincent Picaud, Imad Ali, **Sean Talts**,
Ben Bales, Ari Hartikainen, Matthijs Vàkàr, Andrew Johnson,
Dan Simpson

Stan 2.17 (November 2017) http://mc-stan.org

Simulation

Repeated i.i.d. Trials

- Suppose we repeatedly generate a random outcome from among several potential outcomes
- Suppose the outcome chances are the same each time
  - i.e., outcomes are independent and identically distributed (i.i.d.)
- For example, spin a fair spinner (without cheating), such as one from *Family Cricket*.

Image source: http://replaycricket.com/2010/10/29/family-cricket/

Repeated i.i.d. Binary Trials

- Suppose the outcome is binary and assigned to 0 or 1; e.g.,
  - 20% chance of outcome 1: *ball in play*
  - 80% chance of outcome 0: *ball* not *in play*
- Consider different numbers of bowls delivered.
- How will proportion of successes in sample differ?

Simulating i.i.d. Binary Trials

- R Code: `rbinom(10, N, 0.2) / N`
  - **10 bowls** (10% to 50% success rate)

    2 3 5 2 4 1 2 2 1 1
  - **100 bowls** (16% to 26% success rate)

    26 18 23 17 21 16 21 15 21 26
  - **1000 bowls** (18% to 22% success rate)

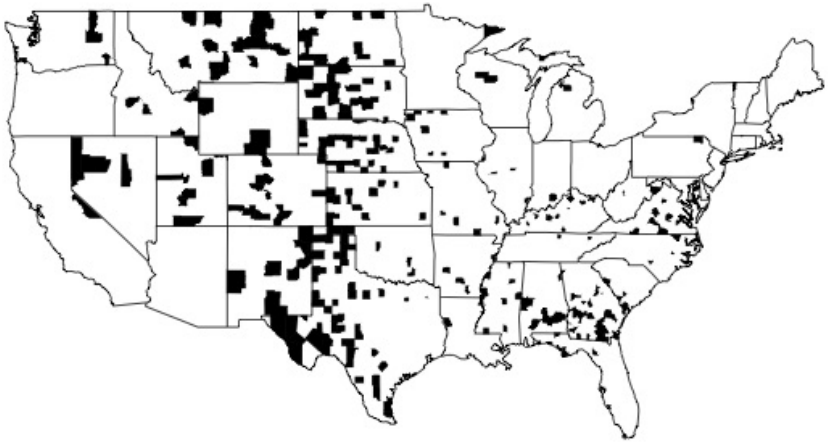
    181 212 175 213 216 179 223 198 188 194
  - **10,000 bowls** (19.3% to 20.3% success rate)

    2029 1955 1981 1980 2001 2014 1931 1982 1989 2020

Pop Quiz! Cancer Clusters

- Why do lowest and highest cancer clusters look so similar?

Lowest kidney cancer death rates



Highest kidney cancer death rates

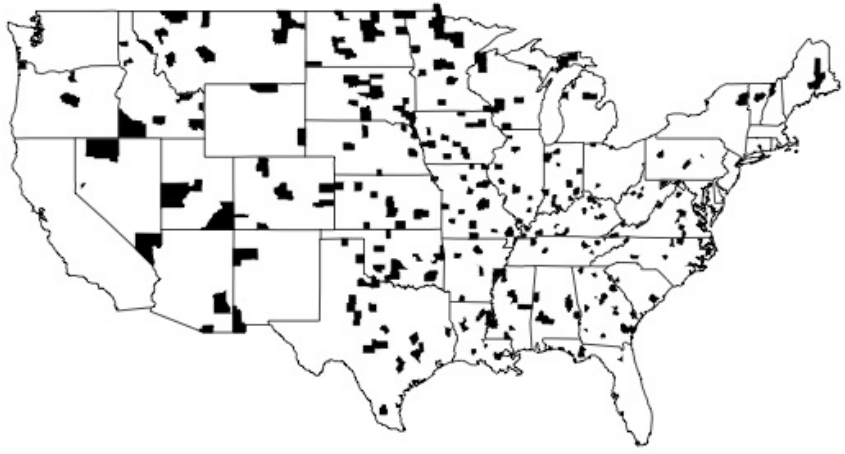


Image from Gelman et al., *Bayesian Data Analysis, 3rd Edition* (2013)

Pop Quiz Answer

- Hint: mix earlier simulations of repeated i.i.d. trials with 20% success and sort:

1/10	1/10	1/10	15/100	16/100
17/100	175/1000	179/1000	18/100	181/1000
188/1000	194/1000	198/1000	2/10	2/10
2/10	2/10	21/100	21/100	21/100
212/1000	213/1000	216/1000	223/1000	23/100
26/100	26/100	3/10	4/10	5/10

- More variation in observed rates with smaller sample sizes

- *Answer*: High cancer and low cancer counties are small populations

Maximum Likeilihood

- Estimate chance of success θ by proportion of successes:

$$\theta^* = \frac{\text{successes}}{\text{attempts}}$$

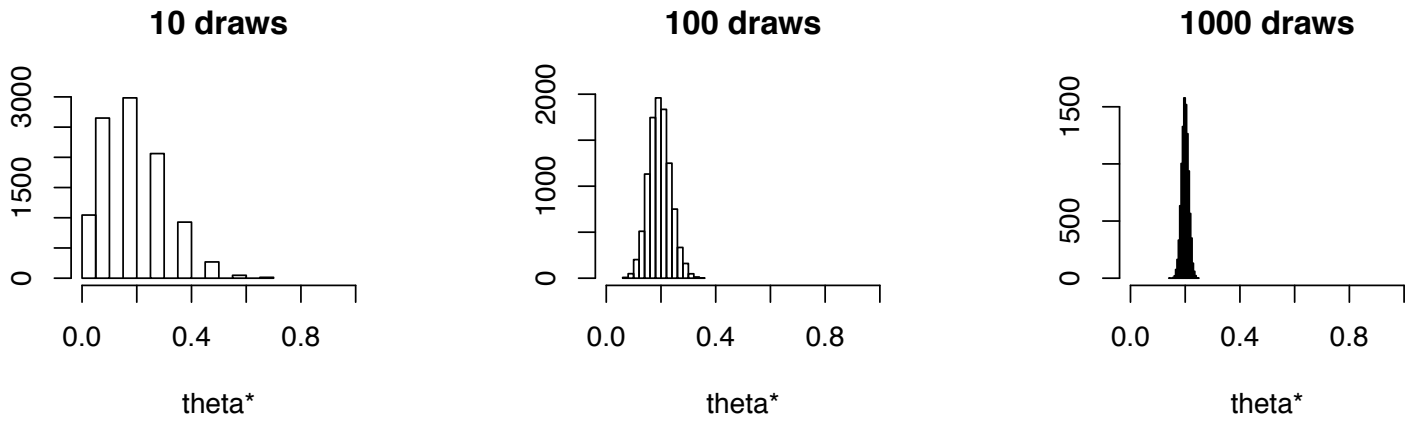
- Simulation shows accuracy depends on the amount of data.
- Statistics is about quantifying uncertainty.
- Bayesian statistics is about using uncertainty in inference.

Notation: θ^* denotes the *maximum likelihood estimate* of θ.

Confidence via Simulation

- Estimator uncertainty (*not* Bayesian posterior)

```
num_sims <- 10000
N <- 100;
theta <- 0.2;
hist(rbinom(num_sims, N, theta) / N,
     main=sprintf("%d simulations",N), xlab="theta*");
```



Example Interval Calculation

- *$P\%$ confidence interval:* interval in which $P\%$ of the estimates are expected to fall.
- Simulation computes intervals to any accuracy.
- Simulate, sort, and inspect the central empirical interval.

```
> sims <- rbinom(10000, 1000, 0.2) / 1000
> sorted_sims <- sort(sims)
> sorted_sims[c(250, 9750)]
[1] 0.176 0.225
```

- The 95% confidence interval is thus $(0.176, 0.225)$
- i.e., if true $\theta = 0.2$, then 95% of the samples of size 1000 used will produce estimates in $(0.176, 0.225)$

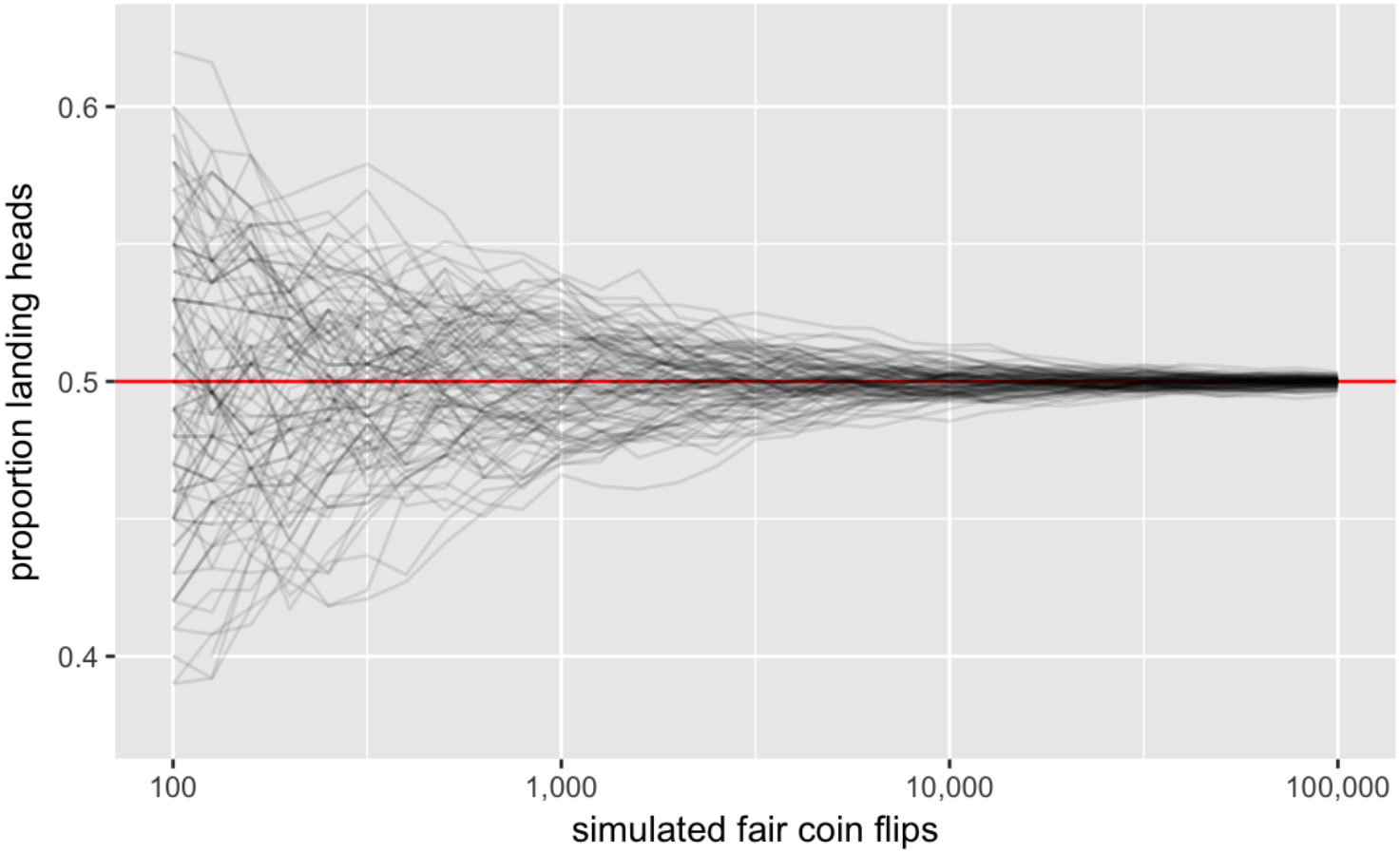
Estimator Bias

- **Bias:** expected difference of estimate from true value
- Continuing previous example

```
> sims <- rbinom(10000, 1000, 0.2) / 1000
> mean(sims)
[1] 0.2002536
```

- Value of 0.2 is estimate of expectation
- Shows this estimator is *unbiased*

Central Limit Theorem (picture)



- proportion heads for 100 sequences of 100,000 flips
- converges gradually to expected value of 0.5

Central Limit Theorem (words)

- **The** theorem of statistics
  - Cardano (1501–1576) conjectured convergence; (Jacob) Bernoulli (1713) proved convergence for binomials (law of large numbers); de Moivre (1733) conjectured the CLT; Laplace (1812) proved i.i.d. version; Lyapunov (1901) removed i.i.d. constraint
- Sample **mean** of N i.i.d. variables with finite expectation
  - **converges** to their expectation as $N \to \infty$
  - **rate** of convergence is $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$
  - constant factor determined by standard deviation
- Each decimal place of accuracy requires $100\times$ more draws

Central Limit Theorem (math)

- Simple i.i.d. version—can be established more generally
- Given N i.i.d. variables $\theta_1, \ldots, \theta_N$ with
  - $\mathbb{E}[\theta_n] = \mu$
  - $\text{sd}[\theta_n] = \sigma$

  the **central limit theorem** states

$$\lim_{N \to \infty} \frac{\theta_1 + \cdots + \theta_N}{N} \sim \text{Normal}\left(\mu, \frac{\sigma}{\sqrt{N}}\right)$$

Numerical Analysis

Floating-Point Standard: IEEE 754

- **Finite numbers** (s: sign; c: mantissa; q: exponent)

$$x = (-1)^s \times c \times 2^q$$

size	s, c *bits*	q *bits*	*range*	*precision*
32-bit	24	8	$\pm 3.4 \times 10^{38}$	7.2 digits
64-bit	53	11	$\pm 1.8 \times 10^{308}$	16 digits

- Quiet and signaling **not-a-number** (NaN)
- Positive and negative **infinity** ($+\infty, -\infty$)
- **Stan** uses 64-bit floating point

Catastrophic Cancellation

- Subtraction risks **catastrophic cancellation**
- Consider $0.99802 - 0.99801 = 0.00001$
  - input has five digits of precision
  - output has single digit of precision
- E.g., problem for sample variance of sequence x

$$\text{var}(x) = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \overline{x})^2$$

if elements x_n close to sample mean

$$\overline{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Welford's Algorithm

- **Streaming computation** uses fixed memory

```
N = 0;    mean = 0;    sum_sq_err = 0

handle(y):
    N += 1
    diff = y - mean
    mean = mean + diff / N
    diff2 = y - mean
    sum_sq_err += diff * diff2

mean():  return mean

var():  return sum_sq_err / (N - 1)
```

- Two stage difference is **less prone to cancellation**

Gaps Between Numbers

- Smallest number greater than zero
  - single precision: 1.4×10^{-45}
  - double precision: 4.9×10^{-324}
- Largest number less than one
  - single precision: $1 - 10^{-7.2}$
  - double precision: $1 - 10^{-16}$
- Gap size **depends on scale**

Lack of Transitivity

- For real numbers $x, y, z \in \mathbb{R}$,

$$x + (y + z) = (x + y) + z$$

- This can fail for floating point due to rounding
  - `(1 + 6e-17) + 6e-17 == 1`
  - `1 + (6e-17 + 6e-17) != 1`
- For square matrices $L L^\top$ is symmetric
- This won't hold for efficient matrix multiplications
  - `(L * L')[1, 2] != (L * L')[2, 1]`

Rounding and Equality

- Dangerous to compare floating point numbers
  - they may have lost precision during calculation
- Rounding
  - default: round toward nearest
  - round toward zero, round to plus or minus infinity

Overflow and Rounding

- Because there is a max size, operations can overflow
  - e.g., `exp(1000)`, `1e200 * 1e200`, ...
- Because there are gaps, operations can round to zero
  - e.g., `exp(-1000)`, `1e-200 * 1e-200`, ...
  - e.g., evaluating $\prod_{n=1}^{N} p(y_n|\theta)$ underflows for $N = 2000$ if $p(y_n|\theta) < 0.1$.

Example: `log1p` and CCDFs

- `log1p(x)` is for evaluating $\log$ near one
  - when x is near zero, `1 + x` catastrophically rounds to 1
  - this forces `log(1 + x)` to round to 0
  - `log1p(x)` avoids `1 + x` operation
  - `log1p(x)` uses Taylor series expansion of $\log(1 + x)$
- Complementary CDFs evaluate CDFs with values near one
  - X is some random variable, e.g., $X \sim \text{Normal}(0, 1)$
  - CDF: $F_X(x) = \Pr[X \leq x]$
  - CCDF: $F_X^{\complement}(x) = 1 - \Pr[X \leq x]$
  - converts range around one to range around zero

Example: log and log_sum_exp

- **Multiplication on the log scale**: log
  - $\log(a \times b) = \log a + \log b$
  - log converts multiplication to addition
  - $\log \prod_n x_n = \sum_n \log x_n$
  - avoids underflow and overflow even if $x_n \ll 1$ or $x_n \gg 1$
  - useful absolutely everywhere (e.g., log likelihoods)
- **Addition on the log scale**: log_sum_exp
  - $\log(a + b) = \log(\exp(\log a) + \exp(\log b))$
  - log converts addition to log sum of exponentials
  - avoids underflow and overflow, preserves precision
  - useful for mixtures (e.g., HMMs, zero-inflated Poisson)

Example: `log_sum_exp`

- Without loss of generality, assume $a > b$ (otherwise swap)

$$\begin{aligned}
\text{log_sum_exp}(a, b) &= \log(\exp(a) + \exp(b)) \\
&= a + \log(\exp(a - a) + \exp(b - a)) \\
&= a + \log(1 + \exp(b - a)) \\
&= a + \text{log1p}(\exp(\text{b} - \text{a}))
\end{aligned}$$

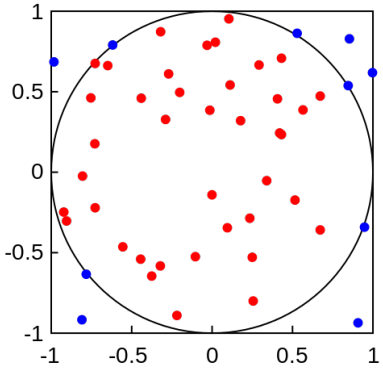
  - **increase precision**: pull a out of $\log()$ and $\exp()$
  - **increase precision**: use `log1p`
  - **prevents overflow**: can't overflow because $b - a \leq 0$

- Generalize to more than two inputs: subtract max

Monte Carlo Integration

Monte Carlo Calculation of π

- Computing $\pi = 3.14\ldots$ via simulation is *the* textbook application of Monte Carlo methods.
- Generate points uniformly at random within the square
- Calculate proportion within circle ($x^2 + y^2 \leq 1$) and multiply by square's area (4) to produce the area of the circle.
- This area is π (radius is 1, so area is $\pi r^2 = \pi$)



plot by Mysid Yoderj,

courtesy of Wikipedia.

Monte Carlo Calculation of π (cont.)

- R code to calculate π with Monte Carlo simulation:

```
> x <- runif(1e6,-1,1)
> y <- runif(1e6,-1,1)

> prop_in_circle <- sum(x^2 + y^2 <= 1) / 1e6

> 4 * prop_in_circle
[1] 3.144032
```

π as an Expectation

- If probability is uniform over the sample space, then an event's probability is its area (volume in general)
- Suppose $X, Y \sim \text{Uniform}(-1, 1)$
- Then $\Pr[X^2 + Y^2 \leq 1] \;=\; \pi/4.$
- To calculate using Monte Carlo draws $(x^{(m)}, y^{(m)})$,

$$\Pr[X^2 + Y^2 \leq 1] \quad = \quad \mathbb{E}\big[\,\mathrm{I}[X^2 + Y^2 < 1]\,\big]$$

$$= \quad \int_{-1}^{1} \int_{-1}^{1} \mathrm{I}[x^2 + y^2 < 1]\; p_X(x)\; p_Y(y)\; \mathrm{d}x\, \mathrm{d}y$$

$$\approx \quad \frac{1}{M}\, \mathrm{I}\left[\left(x^{(m)}\right)^2 + \left(y^{(m)}\right)^2 < 1\right]$$

Calculating π with Stan

- Complete Stan program to compute $\Pr[X^2 + Y^2 \leq 1]$:

```
generated quantities {
  real x = uniform_rng(-1, 1);
  real y = uniform_rng(-1, 1);
  real pi_div_4 = hypot(x, y) <= 1;
}
```

- **Simulates** X and Y
- **Codes indicator function** implicitly with comparison
  - uses Stan's built-in hypotenuse function
  - `hypot(a, b) = sqrt(a^2 + b^2)`

Fitting Stan model for π in R

- Fixed_param algorithm for no parameters

```
> fit <- stan("pi.stan", algorithm="Fixed_param",
                  iter=100000)
```

- Print only what's needed (print output elided manually)

```
> print(fit, digits=3, probs=c(), pars=c("pi_div_4"))

           mean se_mean
pi_div_4 0.786   0.001
```

- Estimate accurate to **within estimated tolerances**
  - $4 \times 0.786 = 3.144$
  - predicted accuracy is 0.004 (four times standard error)

Accuracy of Monte Carlo

- Monte Carlo Integration computes the exact posterior to within any ϵ (*not* like variational Bayes which yields an approximation of the posterior)
- Monte Carlo draws are i.i.d. by definition
- Central limit theorem: expected error decreases at rate of

$$\frac{1}{\sqrt{N}}$$

- 3 decimal places of accuracy with sample size 1e6
- Need $100\times$ larger sample for each digit of accuracy

General Monte Carlo Integration

- MC can calculate arbitrary definite integrals,

$$\int_a^b f(x)\,dx$$

- Let d upper bound $f(x)$ in (a, b); tightness determines computational efficiency
- Then generate random points uniformly in the rectangle bounded by (a, b) and $(0, d)$
- Multiply proportion of draws (x, y) where $y < f(x)$ by area of rectangle, $d \times (b - a)$.
- Can be generalized to multiple dimensions in obvious way

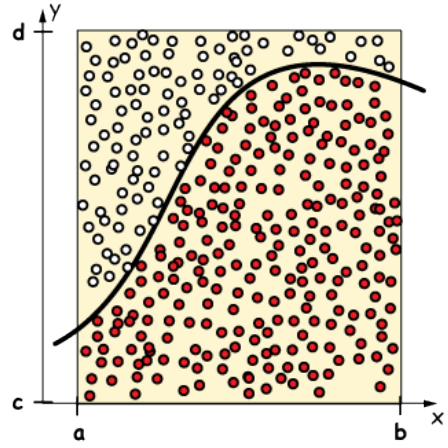


Image courtesy of Jeff Cruzan, http://www.drcruzan.com/NumericalIntegration.html

Expectations of Function of R.V.

- Suppose $f(\theta)$ is a function of random variable vector θ
- Suppose the density of θ is $p(\theta)$
  - *Warning*: θ overloaded as random and bound variable
- Then $f(\theta)$ is also random variable, with expectation

$$\mathbb{E}[f(\theta)] = \int_{\Theta} f(\theta)\ p(\theta)\ d\theta.$$

  - where Θ is support of $p(\theta)$ (i.e., $\Theta = \{\theta \mid p(\theta) > 0\}$

QoI as Expectations

- Most Bayesian quantities of interest (QoI) are expectations over the posterior $p(\theta \mid y)$ of functions $f(\theta)$
- **Bayesian parameter estimation**: $\hat{\theta}$
  - $f(\theta) = \theta$
  - $\hat{\theta} = \mathbb{E}[\theta|y]$ minimizes expected square error
- **Bayesian parameter (co)variance estimation**: $\mathrm{var}[\theta \mid y]$
  - $f(\theta) = (\theta - \hat{\theta})^2$
- **Bayesian event probability**: $\Pr[A \mid y]$
  - $f(\theta) = \mathrm{I}(\theta \in A)$

Expectations via Monte Carlo

- Generate draws $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ drawn from $p(\theta)$
- Monte Carlo Estimator **plugs in average** for expectation:

$$\mathbb{E}[f(\theta)|y] \approx \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

- Can be made **as accurate as desired**, because

$$\mathbb{E}[f(\theta)] = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

- *Reminder*: By CLT, error goes down as $1 / \sqrt{M}$

The Curse of Dimensionality

The Curse

- Intuitions formed in low dimensions break down **do not generalize**
- In high dimensions, **everything is far away**
  - random draws are far away from each other
  - random draws are far away from the mode or meaan
- Sampling algorithms that work in low dimensions often **fail in high dimensions**

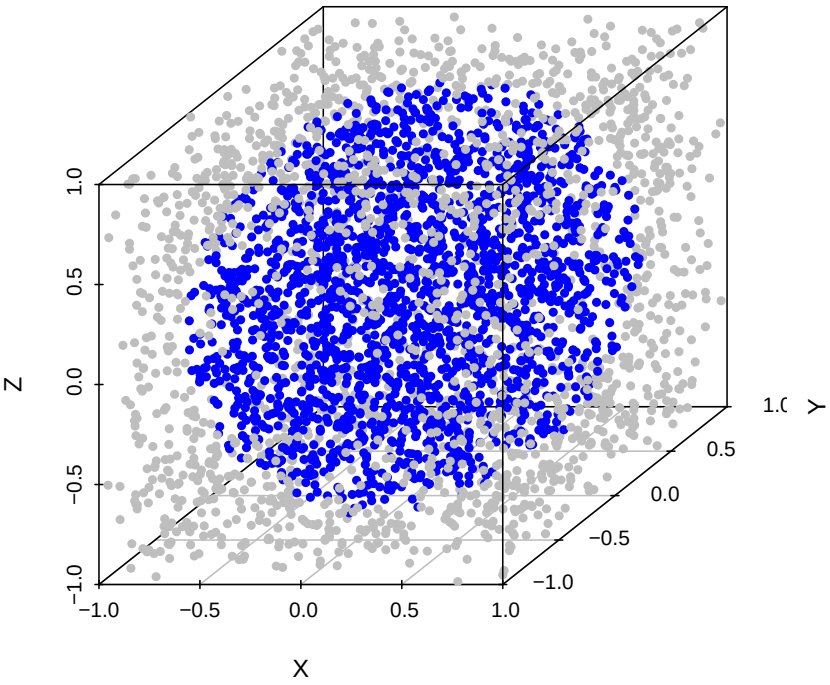
Volume of Ball in Cube

- Assume $x, y, z \sim \text{Uniform}(-1, 1)$,
- $\Pr[(x, y, z) \in \text{unit ball}]$
  is unit ball's fraction of volume.
- Analytic solution:

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathrm{I}[x^2 + y^2 + z^2 \leq 1]\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$$

- Monte Carlo solution:
  - simulate multiple (x, y, z) uniformly in cube
  - count proportion in ball, i.e.,

    $x^2 + y^2 + z^2 \leq 1$



4000 simulations;

blue inside unit ball.

Ball in Cube in Stan

```
generated quantities {
  int<lower = 0, upper = 1> in_ball;
  {
    real x = uniform_rng(-1, 1);
    real y = uniform_rng(-1, 1);
    real z = uniform_rng(-1, 1);
    in_ball = (x^2 + y^2 + z^2 <= 1);
  }
}
```

- `in_ball` is value of indicator (implicit in <=).
- Posterior mean of `in_ball` is fraction draws in ball.
- Posterior mean estimates Pr$[x^2 + y^2 + z^2 \leq 1]$.

Ball in Cube in Stan from RStan

- Use the `Fixed_param` algorithm:

```
> fit <- stan("ball-in-cube.stan",
                algorithm="Fixed_param",
              iter=10000)

> print(fit, probs=c(), digits=3)

          mean se_mean    sd n_eff Rhat
in_ball 0.528   0.004 0.499 19400    1
```

- Thus Pr$[X^2 + Y^2 + Z^2 \leq 1] \approx 0.53$
- with standard error of 0.004, yielding a 95% interval of ± 0.008, i.e., roughly (0.52, 0.54)

Hyperballs in Hypercubes

- **sample uniformly** from container (square, cube, ...)
- 2 dimensions (x, y): compute Pr$[X^2 + Y^2 \leq 1]$
  - unit **disc** inscribed in square
  - calculate π given known area of circle (2π)
- 3 dimensions (x, y, z): compute Pr$[X^2 + Y^2 + Z^2 \leq 1]$
  - unit **ball** inscribed in cube
- N-dimensions $(x_1, \ldots, x_N)$: compute Pr$[X_1^2 + \cdots X_N^2 \leq 1]$
  - unit **hyperball** inscribed in hypercube
- Code event probability as **expectation of indicator**

Hyperballs in Hypercubes in Stan

```
generated quantities {
  int<lower=0, upper=1> in_ball[10];
  {
    real len = 0;
    for (n in 1:10) {
      len = len + uniform_rng(-1, 1)^2;
      in_ball[n] = (len <= 1);
    }
  }
}
```

- draw $x_1, \ldots, x_N$ is implicit in `uniform_rng`
- `in_ball[n]` is 1 iff $x_1^2 + \cdots + x_n^2 \leq 1$; coded as indicator `(len <= 1)`
- sum of squares accumulation reduces quadratic time to linear

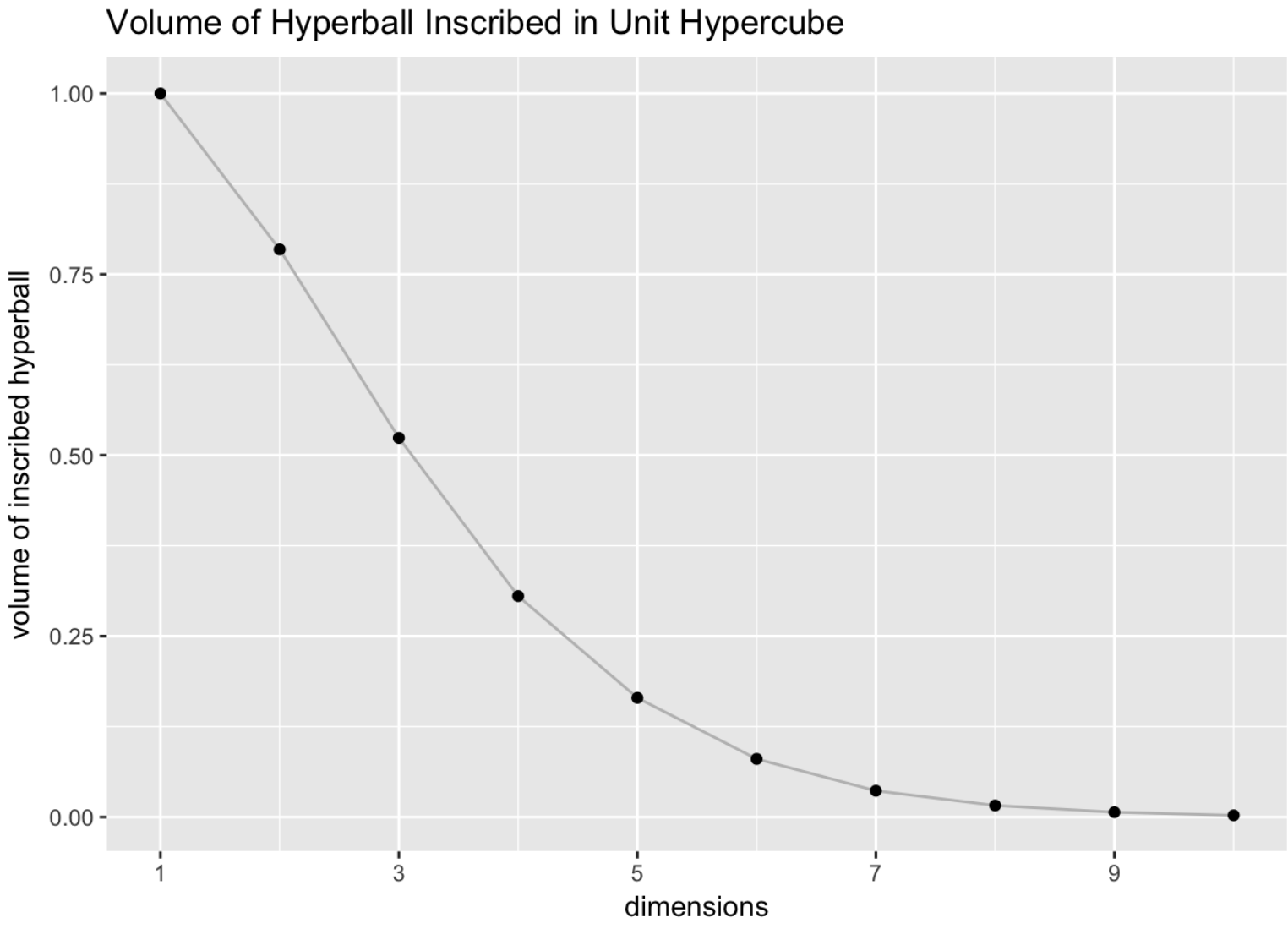
Hyperballs in Hypercubes in RStan

```
> fit <- stan("hyperballs.stan", algorithm="Fixed_param",
              iter=1e4)

> print(fit, probs=c())

             mean se_mean   sd n_eff Rhat
in_ball[1]   1.00       0 0.00 20000  NaN
in_ball[2]   0.78       0 0.41 20000    1
in_ball[3]   0.52       0 0.50 20000    1
in_ball[4]   0.31       0 0.46 20000    1
in_ball[5]   0.17       0 0.38 20000    1
in_ball[6]   0.08       0 0.27 20000    1
in_ball[7]   0.04       0 0.19 18460    1
in_ball[8]   0.02       0 0.12 19370    1
in_ball[9]   0.01       0 0.08 20000    1
in_ball[10] 0.00       0 0.05 20000    1
```

Proportion Volume in Hyperball



Typical Sets

Typical Set Example (1)

- Consider a game of chance with an 80% chance of winning
- Play the game 100 times independently

- What is most likely outcome?

Typical Set Example (2)

- For each trial, there is a 80% chance of success
- For each trial, most likely outcome is success
- Overall, the single **most likely outcome** is all successes

- What's the **most likely number of successes**?

Typical Set Example (3)

- Let $y_n \sim \text{Bernoulli}(0.9)$ for $n \in 1:100$ be the trials
- Expected number of successes

$$\begin{aligned}
\mathbb{E}\left[\sum_{n=1}^{100} y_n\right] &= \sum_{n=1}^{100} \mathbb{E}[y_n] \\
&= \sum_{n=1}^{100} 0.8 \\
&= 0.8 \times 100 \\
&= 80
\end{aligned}$$

- **most likely outcome** (all successes) **is an outlier**!

$$\Pr[100 \text{ successes}] = 0.8^{100} < 10^{-10}$$

Typical Set Example (4)

- Maximum likelihood (most likely) outcome is **atypical**
- Expectations involve count times probability
- 100 success sequences: $\binom{100}{100} = \frac{100!}{100! \times 1!} = 1$
- 80 success sequences: $\binom{100}{80} = \frac{100!}{80! \times 20!} > 10^{20}$
- Thus chance of 80 success is much higher than 100

$$\begin{aligned} \text{Binomial}(80 \mid 100, 0.8) &= \binom{100}{20} \times 0.8^{80} \times 0.2^{20} \\ &\gg \binom{100}{1} \times 0.8^{100} \\ &= \text{Binomial}(100 \mid 100, 0.8) \end{aligned}$$

Typical Set

- Goal is to **evaluate posterior expectations using draws**

$$\mathbb{E}\left[f(\theta) \mid y\right] = \int_{\Theta} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

- A **typical set** A_ϵ (at some level) is the set
  - of values with typical log density (near distribution entropy)
  - containing $1 - \epsilon$ of the probability mass
- A typical set A_ϵ **suffices for integration**

$$\int_{\Theta} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta = \int_{A_\epsilon} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta$$

Typical Draws from Multi-Normal

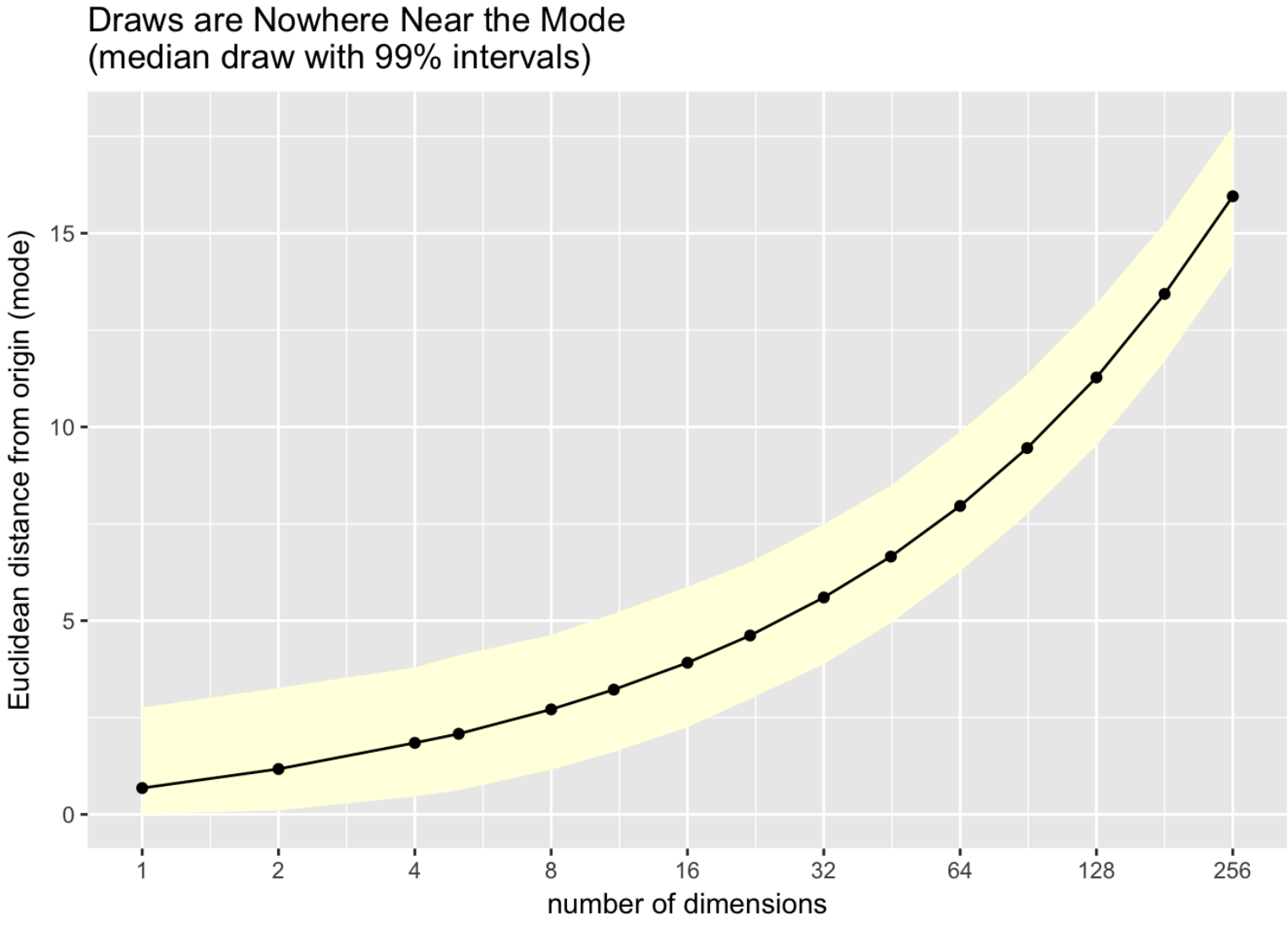
- $Y \sim \text{MultiNormal}(0, \text{I}_N)$ is standard multivariate normal
- $Y_n \sim \text{Normal}(0, 1)$ is thus independently standard normal
- Joint density: $p_Y(y) = \prod_{n=1}^{N} \text{Normal}(y_n \mid 0, 1)$
- Mean, median, and mode (max) of $p_Y(y)$ at $y = 0$

- How far do we expect Y to be from the mode?
- What is the log density of a typical draw of Y?

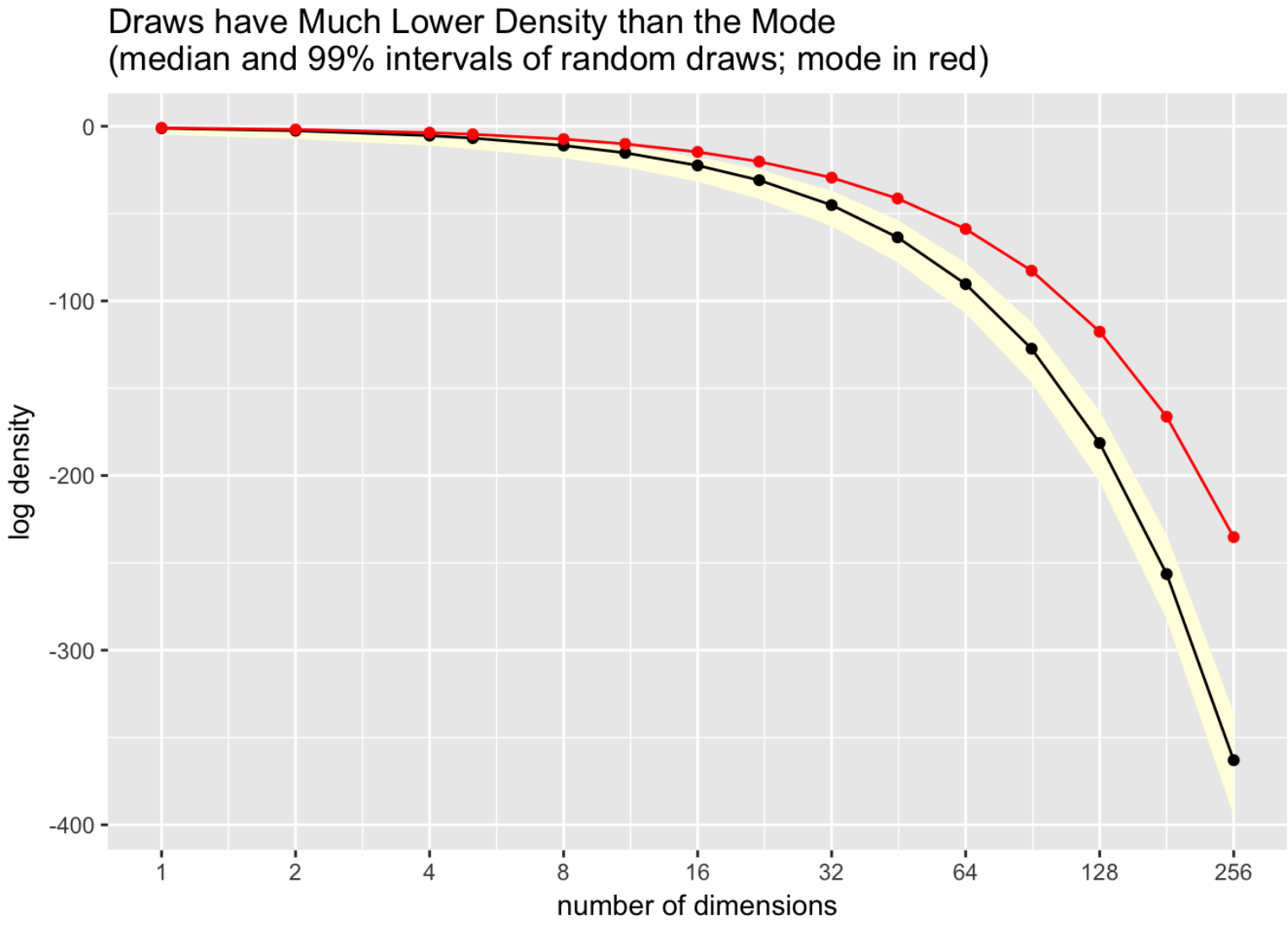
Multi-Normal Draws in Stan

```
generated quantities {
  real dist_to_origin[256];
  real log_lik[256];
  real log_lik_mean[256];
  {
    real sq_dist = 0;  real ll = 0;  real llm = 0;
    for (n in 1:256) {
      real y = normal_rng(0, 1);
      ll = ll + normal_lpdf(y | 0, 1);
      llm = llm + normal_lpdf(0 | 0, 1);
      sq_dist = sq_dist + y^2;
      dist_to_origin[n] = sqrt(sq_dist);
      log_lik[n] = ll;
      log_lik_mean[n] = llm;
    }
  }
}
```

Normal Variate Distance to Mode



Normal Variate Log Density



Normal Mode not in Typical Set

- Plots show that in a standard normal of **more than 5 dimensions**, that the **mode is not in the typical set**
- An **Asimov data set** uses an average member of a set represent the whole set
  - based on Isaac Asimov's short story "Franchise" in which a single average voter represented everyone
  - the average member of a multivariate normal is the mean
  - thus no members of the typical set are average in this sense
  - popular in physics
  - **very poor** solution for most inferential purposes

Concentration of Measure

Concentration of Measure

- We care about probability **mass**, not **density**
- Events with non-zero probability have probability mass, e.g., $\Pr[\theta_0 > \theta_1 \mid y]$
- Mass arises from integrating over density
- As data size increases, posterior concentrates around true value

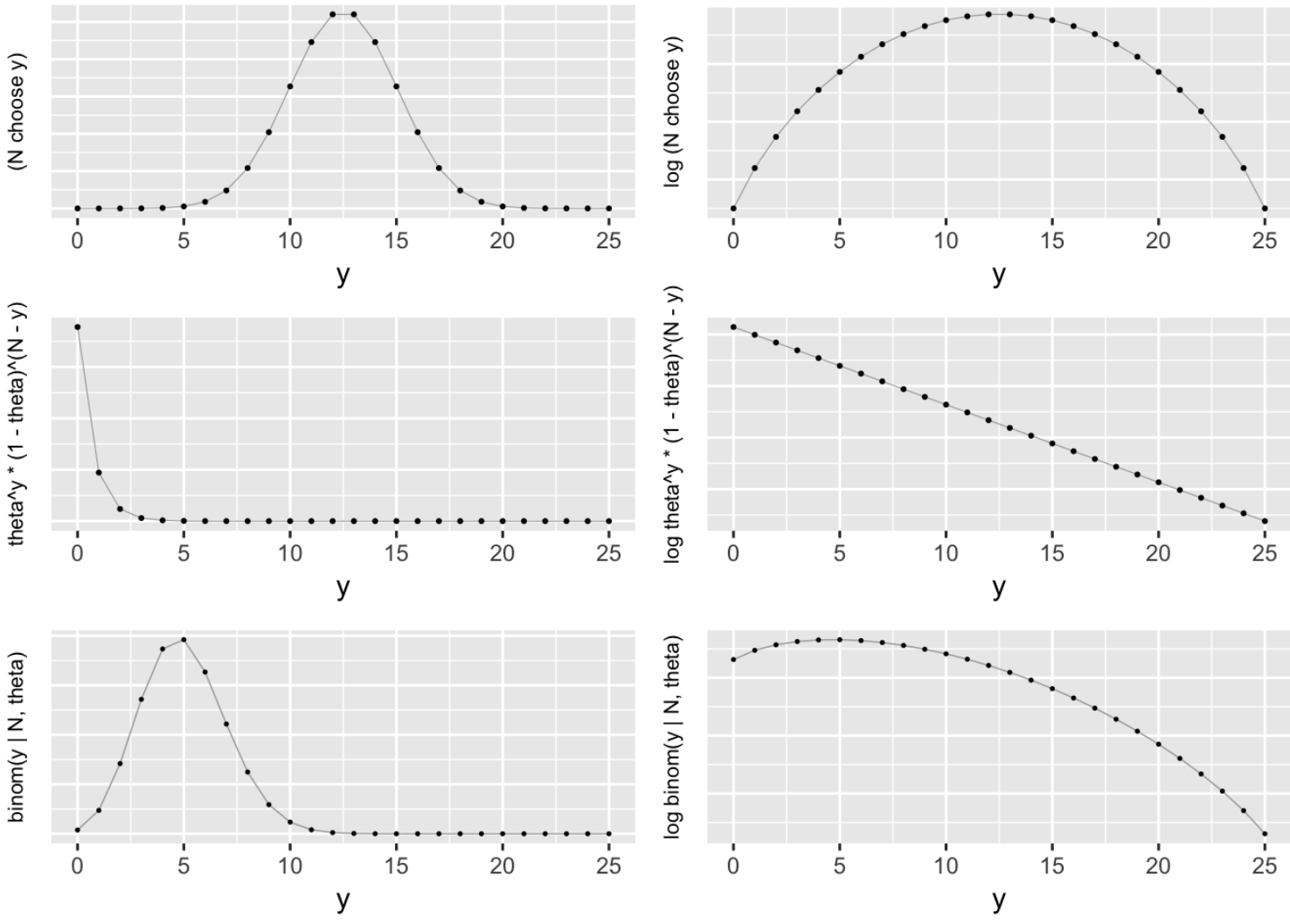
E.g., Binomial Concentration

- $y \sim \text{Binomial}(N, \theta)$

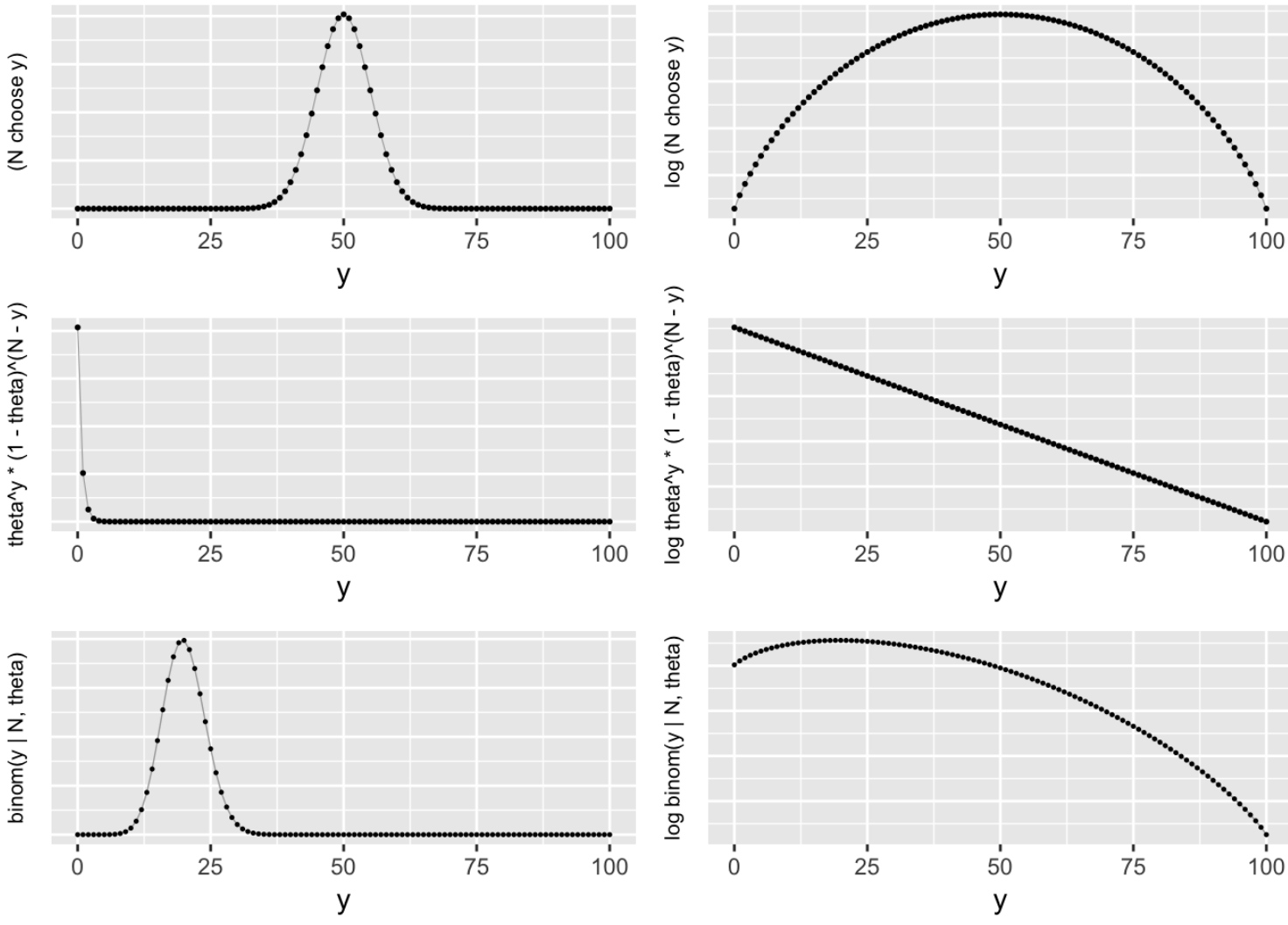
$$\text{Binomial}(y \mid N, \theta) = \binom{N}{y} \theta^y \, (1 - \theta)^{N-y}$$

- As $N \to \infty$, posterior average y/N concentrates around θ
- Concentration governed by central limit theorem

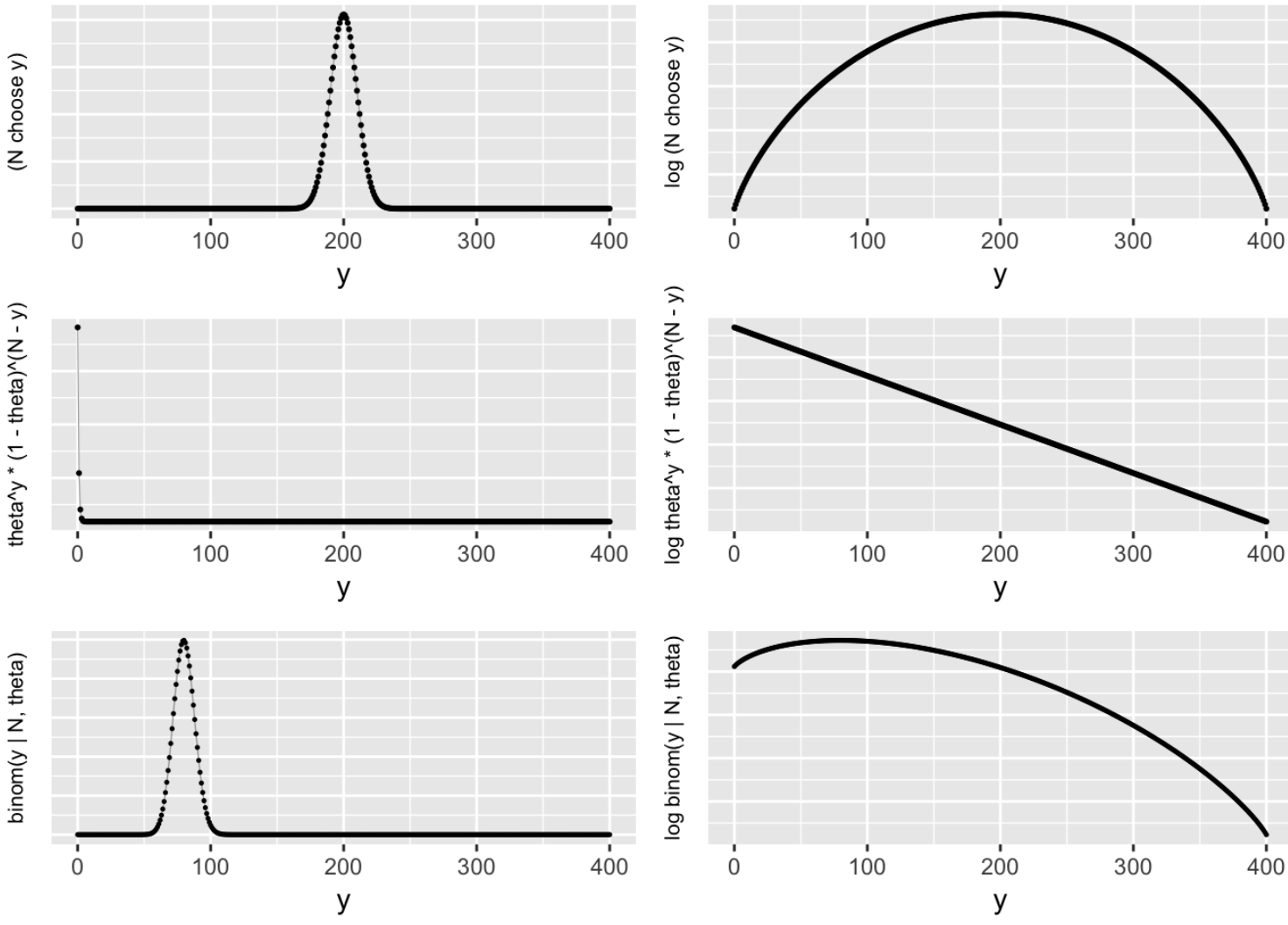
Binomial Concentration, $N = 25$



Binomial Concentration, $N = 100$



Binomial Concentration, $N = 400$



Continuous Hypervolumes

- Generalize discrete to continuous
  - discrete: combinations times probability mass
  - continuous: volume times probabilty density

- Volume of ball at given radius (r) grows exponentially with dimension (d):

$$\text{volume} \propto r^d$$

  - line has length $\propto r$
  - disc has area $\propto r^2$
  - ball has volume $\propto r^3$
  - 4-dimensional hyperball has volume $\propto r^4$

Markov Chain Monte Carlo

Markov Chain Monte Carlo

- Standard Monte Carlo draws i.i.d. samples

$$\theta^{(1)}, \ldots, \theta^{(M)}$$

  according to a probability function $p(\theta)$

- Drawing i.i.d. samples typically impossible for complex densities like Bayesian posteriors $p(\theta|y)$

- Instead, use Markov chain Monte Carlo (MCMC) to draw $\theta^{(1)}, \ldots, \theta^{(M)}$ from a Markov chain with appropriate stationary distribution $p(\theta|y)$.

Markov Chains

- A Markov Chain is a sequence of random variables

$$\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$$

such that $\theta^{(m)}$ only depends on $\theta^{(m-1)}$, i.e.,

$$p(\theta^{(m)}|y, \theta^{(1)}, \ldots, \theta^{(m-1)}) = p(\theta^{(m)}|y, \theta^{(m-1)})$$

Markov Chain Monte Carlo

- Simulating independent draws from the posterior $p(\theta|y)$ usually intractable
- Simulating a Markov chain $\theta^{(1)}, \ldots, \theta^{(M)}$ with marginals equal to posterior, i.e.,

$$p(\theta^{(m)}|y) = p(\theta|y)$$

  often is tractable

- Replace indepedent draws with Markov chain of draws
  - Plug in just like ordinary (non-Markov chain) Monte Carlo
  - Adjust standard errors for correlation in Markov chain

MCMC for Posterior Mean

- Standard Bayesian estimator is posterior mean

$$\hat{\theta} = \int_{\Theta} \theta \, p(\theta|y) \, d\theta$$

  - Posterior mean minimizes expected square error

- Estimate is a conditional expectation

$$\hat{\theta} = \mathbb{E}[\theta|y]$$

- Compute by averaging

$$\hat{\theta} \approx \frac{1}{M} \sum_{m=1}^{M} \theta$$

MCMC for Posterior Variance

- Posterior variance works the same way, given previous result

$$\mathbb{E}[(\theta - \mathbb{E}[\theta])^2] \approx \frac{1}{M} \sum_{m=1}^{M} (\theta^{(m)} - \hat{\theta})^2$$

MCMC for Posterior Median

- Alternative Bayesian estimator is posterior median
  - Posterior median minimizes expected absolute error
- Calculate as middle draw of $\theta^{(1)}, \ldots, \theta^{(M)}$
  - just sort and take halfway value
  - e.g., Stan shows 50% point (or other quantiles)

MCMC for Event Probability

- Event probabilities are also expectations, e.g.,

$$\Pr[\theta_1 > \theta_2] \;=\; \mathbb{E}[\mathsf{I}[\theta_1 > \theta_2]] \;=\; \int_{\Theta} \mathsf{I}[\theta_1 > \theta_2]\, p(\theta|y) d\theta.$$

- Estimation via MCMC just another plug-in:

$$\Pr[\theta_1 > \theta_2] \approx \frac{1}{M} \sum_{m=1}^{M} \mathsf{I}[\theta_1^{(m)} > \theta_2^{(m)}]$$

- Again, can be made as accurate as necessary

MCMC for Quantiles (incl. median)

- These are not expectations, but still plug in
- Alternative Bayesian estimator is posterior median
  - Posterior median minimizes expected absolute error
- Estimate as median draw of $\theta^{(1)}, \ldots, \theta^{(M)}$
  - just sort and take halfway value
  - e.g., Stan shows 50% point (or other quantiles)
- Other quantiles including interval bounds similar
  - estimate with quantile of draws
  - estimation error goes up in tail (based on fewer draws)

MCMC Algorithms

Random-Walk Metropolis

- Draw random initial parameter vector $\theta^{(1)}$ (in support)
- For $m \in 2{:}M$
  - Sample proposal from a (symmetric) jumping distribution, e.g.,

$$\theta^* \sim \text{MultiNormal}(\theta^{(m-1)}, \sigma\mathbf{I})$$

    where $\mathbf{I}$ is the identity matrix

  - Draw $u^{(m)} \sim \text{Uniform}(0, 1)$ and set

$$\theta^{(m)} = \begin{cases} \theta^* & \text{if } u^{(m)} < \dfrac{p(\theta^*|y)}{p(\theta^{(m)}|y)} \\ \theta^{(m-1)} & \text{otherwise} \end{cases}$$

Metropolis and Normalization

- Metropolis only uses posterior in a ratio

$$\frac{p(\theta^* \mid y)}{p(\theta^{(m)} \mid y)} = \frac{p(y, \theta^*) \ / \ p(y)}{p(y, \theta^{(m)}) \ / \ p(y)}$$

$$= \frac{p(y, \theta^*)}{p(y, \theta^{(m)})}$$

$$= \frac{p(y \mid \theta^*)\, p(\theta^*)}{p(y \mid \theta^{(m)})\, p(\theta^{(m)})}$$

- Drops $p(y)$ term with nasty integral
- Baye's rule reduces to likelihood and prior

Metropolis-Hastings

- Generalizes Metropolis to asymmetric proposals
- Acceptance ratio is

$$\frac{J(\theta^{(m)}|\theta^*) \times p(\theta^*|y)}{J(\theta^*|\theta^{(m-1)}) \times p(\theta^{(m)}|y)}$$

  where J is the (potentially asymmetric) proposal density
- i.e.,

$$\frac{\text{density at } \theta^* \text{ and jump to } \theta^{(m-1)}}{\text{density at } \theta^{(m-1)} \text{ and jump to } \theta^*}$$

- Like Metropolis, only requires ratios

Detailed Balance & Reversibility

- Sufficient for a *stationary distribution* on Markov chain

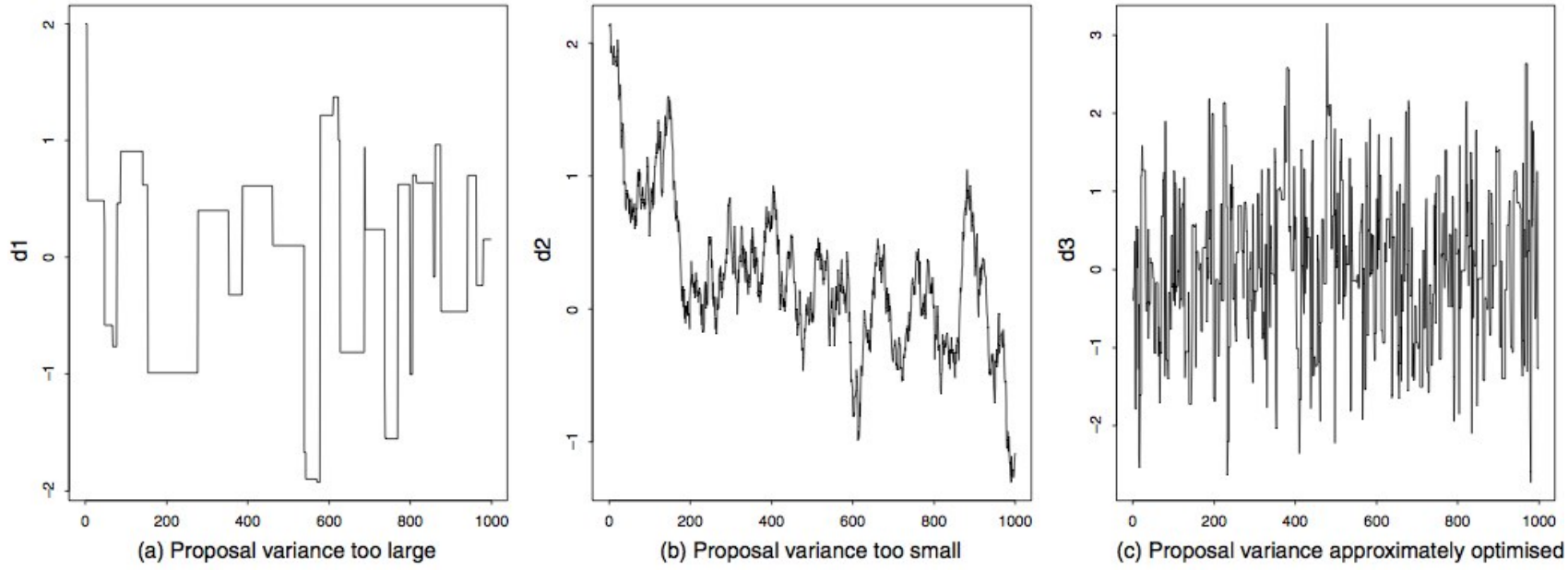
$$p(\theta^{(m)}) = p(\theta) \text{ for all } m \gg 1$$

- Suppose $\pi(\theta^{(m+1)}|\theta^{(m)})$ is Markov transition density
- Detailed balance is a reversibility equilibrium condition of
  - density at $\theta^{(m)}$ and jump density to $\theta^{(m+1)}$
  - density at $\theta^{(m+1)}$ and jump density back to $\theta^{(m)}$

$$p(\theta^{(m)}) \times \pi(\theta^{(m+1)}|\theta^{(m)}) = p(\theta^{(m+1)}) \times \pi(\theta^{(m)}|\theta^{(m+1)})$$

Optimal Proposal Scale?

- Proposal scale σ is a free; too low or high is inefficient

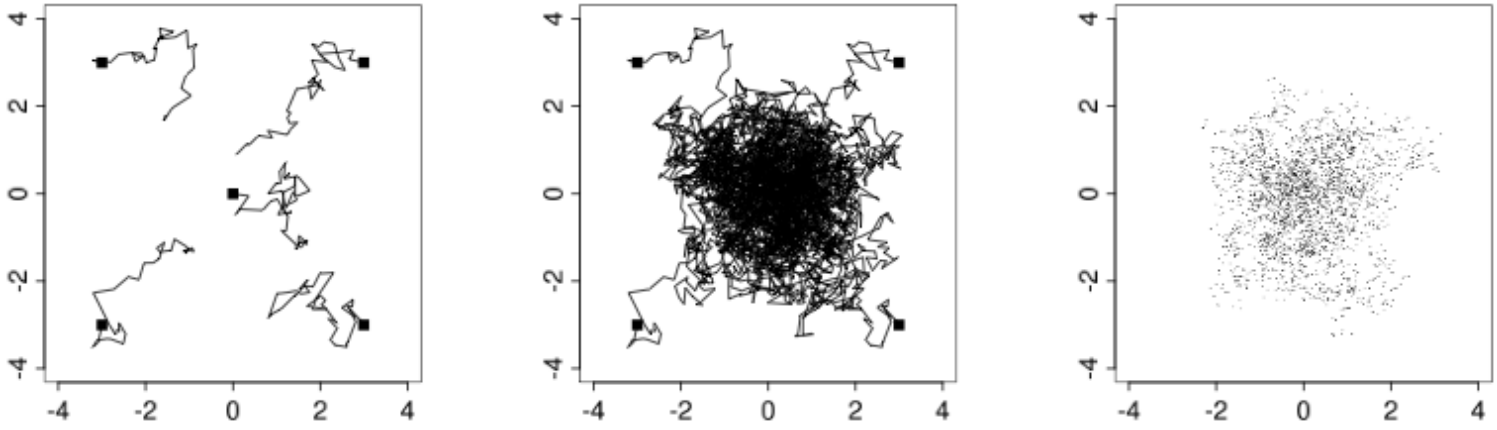


- *Traceplots* show parameter value on y axis, iterations on x
- Empirical tuning problem; theoretical optima exist for some cases

Roberts and Rosenthal (2001) Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science*.

Convergence and Stationarity

- May take many iterations for chain to reach equilibrium
- Different initializations should converge in distribution



- Four chains with different starting points. *Left*) 50 iterations; *Center*) 1000 iterations; *Right*) Draws from second half of each chain

Potential Scale Reduction ($\hat{R}$)

- Gelman & Rubin recommend M chains of N draws with **diffuse initializations**
- Measure that each chain has same posterior mean and variance
- If not, may be stuck in multiple modes or just not converged yet
- Define statistic $\hat{R}$ of chains such that **at convergence**, $\hat{R} \rightarrow 1$
  - $\hat{R} >> 1$ implies non-convergence
  - $\hat{R} \approx 1$ **does not guarantee convergence**
  - Only measures marginals

Split $\hat{R}$

- Vanilla $\hat{R}$ may not diagnose non-stationarity
  - e.g., a sequence of chains with an increasing parameter
- **Split** $\hat{R}$: Stan splits each chain into first and second half
  - start with M Markov chains of N draws each
  - split each in half to creates $2M$ chains of $N/2$ draws
  - then apply $\hat{R}$ to the $2M$ chains

Calculating $\hat{R}$ Statistic

- **Between-sample variance** estimate

$$B = \frac{N}{M-1} \sum_{m=1}^{M} (\bar{\theta}_m^{(\bullet)} - \bar{\theta}_\bullet^{(\bullet)})^2,$$

where

$$\bar{\theta}_m^{(\bullet)} = \frac{1}{N} \sum_{n=1}^{N} \theta_m^{(n)} \quad \text{and} \quad \bar{\theta}_\bullet^{(\bullet)} = \frac{1}{M} \sum_{m=1}^{M} \bar{\theta}_m^{(\bullet)}.$$

- **Within-sample variance** estimate:

$$W = \frac{1}{M} \sum_{m=1}^{M} s_m^2,$$

where

$$s_m^2 = \frac{1}{N-1} \sum_{n=1}^{N} (\theta_m^{(n)} - \bar{\theta}_m^{(\bullet)})^2.$$

Calculating $\hat{R}$ Statistic (cont.)

- **Variance** estimate:

$$\widehat{\text{var}}^{+}(\theta|y) = \frac{N-1}{N} W + \frac{1}{N} B.$$

- **Potential scale reduction** statistic ("R hat")

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}(\theta|y)}{W}}.$$

Correlations in Posterior Draws

- Markov chains typically display autocorrelation in the series of draws $\theta^{(1)}, \ldots, \theta^{(m)}$
- Without i.i.d. draws, central limit theorem *does not apply*
- Effective sample size N_{eff} divides out autocorrelation
- N_{eff} must be estimated from sample
  - Fast Fourier transform efficiently computes correlations at all lags
- Estimation accuracy proportional to

$$\frac{1}{\sqrt{N_{\text{eff}}}}$$

- Compare previous plots; good choice of σ leads to high N_{eff}

Effective Sample Size (ESS)

- Autocorrelation at lag t is correlation between subsequences
  - $(\theta^{(1)}, \ldots, \theta^{(N-t)})$ and $(\theta^{(1+t)}, \ldots, \theta^{(N)})$
- Suppose chain has density $p(\theta)$ with
  - $\mathbb{E}[\theta] = \mu$ and $\mathrm{Var}[\theta] = \sigma^2$
- Autocorrelation ρ_t at lag $t \geq 0$:

$$\rho_t = \frac{1}{\sigma^2} \int_\Theta (\theta^{(n)} - \mu)(\theta^{(n+t)} - \mu)\, p(\theta)\, d\theta$$

- Because $p(\theta^{(n)}) = p(\theta^{(n+t)}) = p(\theta)$ at convergence,

$$\rho_t = \frac{1}{\sigma^2} \int_\Theta \theta^{(n)}\, \theta^{(n+t)}\, p(\theta)\, d\theta$$

Estimating Autocorrelations

- Effective sample size is defined by

$$N_{\text{eff}} = \frac{N}{\sum_{t=-\infty}^{\infty} \rho_t} = \frac{N}{1+2\sum_{t=1}^{\infty} \rho_t}$$

- Estimate in terms of variograms at lag t,

$$V_t = \frac{1}{M} \sum_{m=1}^{M} \left(\frac{1}{N_m - t} \sum_{n=t+1}^{N_m} \left(\theta_m^{(n)} - \theta_m^{(n-t)} \right)^2 \right)$$

- Estimate autocorrelation at lag t using cross-chain variance as

$$\hat{\rho}_t = 1 - \frac{V_t}{2\,\widehat{\text{var}}^{+}}$$

- If not converged, $\widehat{\text{var}}^{+}$ overestimates variance

- Efficiently calculate using fast Fourier transform (w. padding)

Estimating N_{eff}

- Let T' be first lag s.t. $\rho_{T'+1} < 0$,
- Estimate autocorrelation by

$$\hat{N}_{\text{eff}} = \frac{MN}{1 + \sum_{t=1}^{T'} \hat{\rho}_t}.$$

- NUTS avoids negative autocorrelations, so first negative autocorrelation estimate is reasonable

- See: Charles Geyer (2013) Introduction to MCMC. In *Handbook of MCMC*. (free online at `http://www.mcmchandbook.net/index.html`)

Gibbs Sampling

- Draw random initial parameter vector $\theta^{(1)}$ (in support)
- For $m \in 2{:}M$
  - For $n \in 1{:}N$:
    - draw $\theta_n^{(m)}$ according to conditional

$$p(\theta_n | \theta_1^{(m)}, \ldots, \theta_{n-1}^{(m)}, \theta_{n+1}^{(m-1)}, \ldots, \theta_N^{(m-1)}, y).$$

- e.g, with $\theta = (\theta_1, \theta_2, \theta_3)$:
  - draw $\theta_1^{(m)}$ according to $p(\theta_1 | \theta_2^{(m-1)}, \theta_3^{(m-1)}, y)$
  - draw $\theta_2^{(m)}$ according to $p(\theta_2 | \theta_1^{(m)}, \theta_3^{(m-1)}, y)$
  - draw $\theta_3^{(m)}$ according to $p(\theta_3 | \theta_1^{(m)}, \theta_2^{(m)}, y)$

Generalized Gibbs

- "Proper" Gibbs requires the conditional Monte Carlo draws
  - typically works only for conjugate priors
- In general case, may need to use less efficient conditional draws
  - Slice sampling is a popular general technique that works for discrete or continuous θ_n
  - Adaptive rejection sampling is another alternative
  - Very difficult in more than one or two dimensions

Sampling Efficiency

- We care only about N_{eff} per second
- Decompose into
    1. Iterations per second
    2. Effective samples per iteration
- Gibbs and Metropolis have high iterations per second (especially Metropolis)
- But they have low effective samples per iteration (especially Metropolis)
- Both are particular weak when there is high correlation among the parameters in the posterior

Hamiltonian Monte Carlo & NUTS

- Slower iterations per second than Gibbs or Metropolis
- Much higher number of effective samples per iteration for complex posteriors (i.e., high curvature and correlation)
- Overall, much higher $N_{1\text{ff}}$ per second

- Details in the next talk . . .
- Along with details of how Stan implements HMC and NUTS