Statistical Techniques for Data Science & Robotics

Week 12

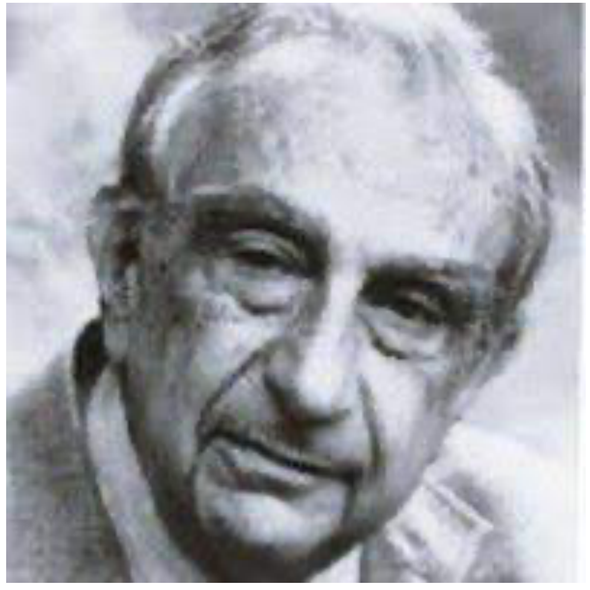
Quiz

- Q1: Describe **the detailed balance** property.

Objectives for today

- Markov Chains
- Metropolis-Hastings algorithm
  - Random walk Metropolis
  - Metropolis-Hastings
- Hamiltonian MC

Metropolis-Hastings



Metropolis: Random walk

- **Random Walk:** $q(x^{(j)} \mid x) = q(x \mid x^{(j)})$

$$\rho(x^*, x^{(j)}) = \min\left(1, \frac{f(x^*)}{f(x^{(j)})}\right)$$

Metropolis-Hastings

- You have no access to CDF
- You cannot use Accept-Reject (too slow)
- But if you still can calculate p.d.f. up to a proportionality constant

    - Then you can use MCMC:
        - Metropolis Sampling
        - Metropolis-Hastings Sampling
        - Gibbs Sampling
        - Hamiltonian Monte Carlo

Metropolis-Hastings: Method

- **Given some current value x(j) sample the next value x* using a proposal distribution $q(x)$:**
  - sample x* comes from $q(x \mid x^{(j)})$
- **Calculate acceptance probability:**
  - $$A(x^*, x^{(j)}) = \min\left(1, \frac{f(x^*)}{f(x^{(j)})} \frac{q(x^{(j)} \mid x^*)}{q(x^* \mid x^{(j)})}\right)$$
- **Set x(j+1) <- x* with probability**

$$A(x^*, x^{(j)})$$

- **Otherwise, set x(j+1) <- x(j)**

Metropolis-Hasting algorithm. discrete distribution

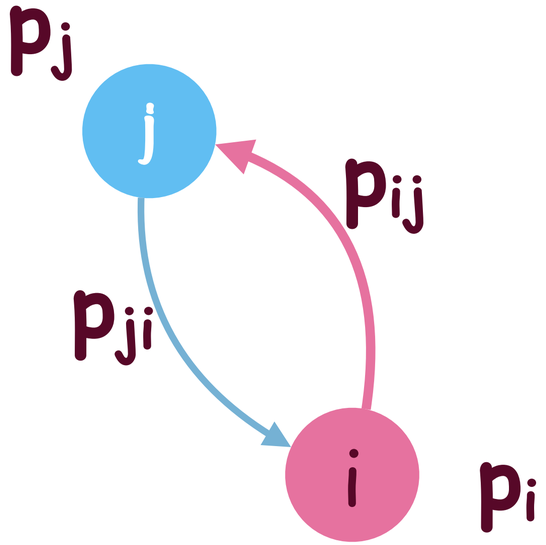
- **The Metropolis-Hasting algorithm is a general method to design a Markov chain whose stationary distribution is a given target distribution p.**
  - **Start with a connected undirected graph G on the set of states.**
- **In general, let r be the maximum degree of any vertex of G.**

Metropolis-Hasting algorithm. discrete distribution

- The transitions of the Markov chain are defined as follows:
- At state i select neighbour j with probability $\frac{1}{r}$.
- Since the degree of i may be less than r, with some probability no edge (transition) is selected and the walk remains at i.
- If a neighbour j is selected and $p_j \geq p_i$, go to j.
- If $p_j < p_i$, then

  go to j with probability $\frac{p_j}{p_i}$ and

  stay at i with probability $(1 - \frac{p_j}{p_i})$



Metropolis-Hasting algorithm

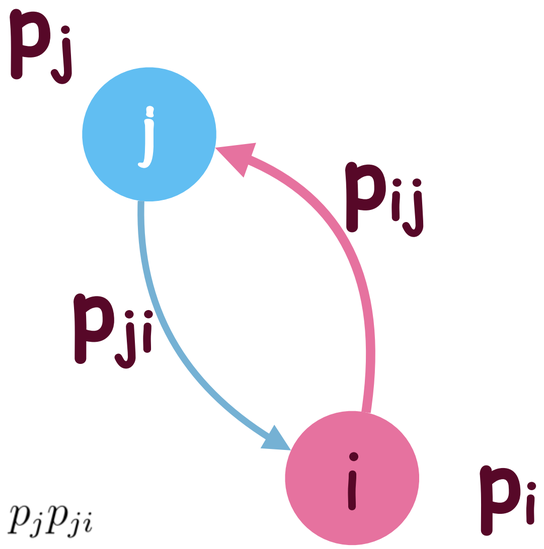
- The transitions of the Markov chain are defined as follows:
- At state i select neighbour j with probability $\frac{1}{r}$.
- Since the degree of i may be less than r, with some probability no edge (transition) is selected and the walk remains at i.
- If a neighbour j is selected and $p_j \geq p_i$, go to j.
- If $p_j < p_i$, then

  go to j with probability $\frac{p_j}{p_i}$ and

  stay at i with probability $(1 - \frac{p_j}{p_i})$

$$p_{ij} = \frac{1}{r} \min\left(1, \frac{p_j}{p_i}\right)$$

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}$$

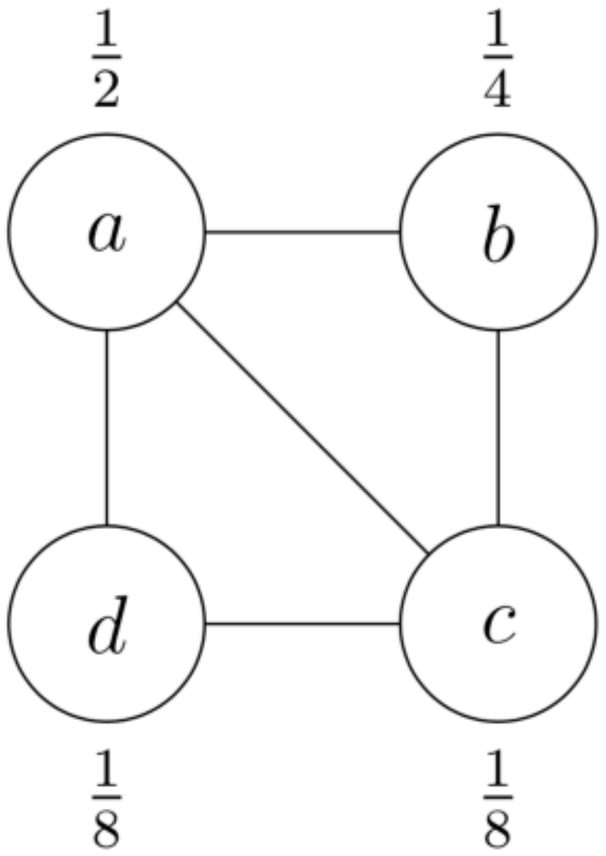


$$p_i p_{ij} = \frac{p_i}{r} \min\left(1, \frac{p_j}{p_i}\right) = \frac{1}{r} \min(p_i, p_j) = \frac{p_j}{r} \min\left(1, \frac{p_i}{p_j}\right) = p_j p_{ji}$$

Example

- **Goals:**
  - Construct the transition matrix using M-H
  - Check the detailed balance property

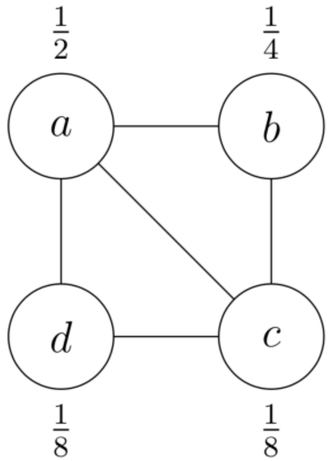
$$p(a) = \frac{1}{2}$$
$$p(b) = \frac{1}{4}$$
$$p(c) = \frac{1}{8}$$
$$p(d) = \frac{1}{8}$$



Example

- **r=3**

$p(a) = \frac{1}{2}$
$p(b) = \frac{1}{4}$
$p(c) = \frac{1}{8}$
$p(d) = \frac{1}{8}$



$a \to b \quad \frac{1}{3}\ \frac{1}{4}\ \frac{2}{1} = \frac{1}{6}$
$a \to c \quad \frac{1}{3}\ \frac{1}{8}\ \frac{2}{1} = \frac{1}{12}$
$a \to d \quad \frac{1}{3}\ \frac{1}{8}\ \frac{2}{1} = \frac{1}{12}$
$a \to a \quad 1-\frac{1}{6}-\frac{1}{12}-\frac{1}{12}=\frac{2}{3}$

$b \to a \quad \frac{1}{3}$
$b \to c \quad \frac{1}{3}\ \frac{1}{8}\ \frac{4}{1} = \frac{1}{6}$
$b \to b \quad 1-\frac{1}{3}-\frac{1}{6} = \frac{1}{2}$

$c \to a \quad \frac{1}{3}$
$c \to b \quad \frac{1}{3}$
$c \to d \quad \frac{1}{3}$
$c \to c \quad 1-\frac{1}{3}-\frac{1}{3}-\frac{1}{3} = 0$

$d \to a \quad \frac{1}{3}$
$d \to c \quad \frac{1}{3}$
$d \to d \quad 1-\frac{1}{3}-\frac{1}{3} = \frac{1}{3}$

$$p(a) = p(a)p(a \to a) + p(b)p(b \to a) + p(c)p(c \to a) + p(d)p(d \to a)$$
$$= \frac{1}{2}\ \frac{2}{3} + \frac{1}{4}\ \frac{1}{3} + \frac{1}{8}\ \frac{1}{3} + \frac{1}{8}\ \frac{1}{3} = \frac{1}{2}$$

Check the property

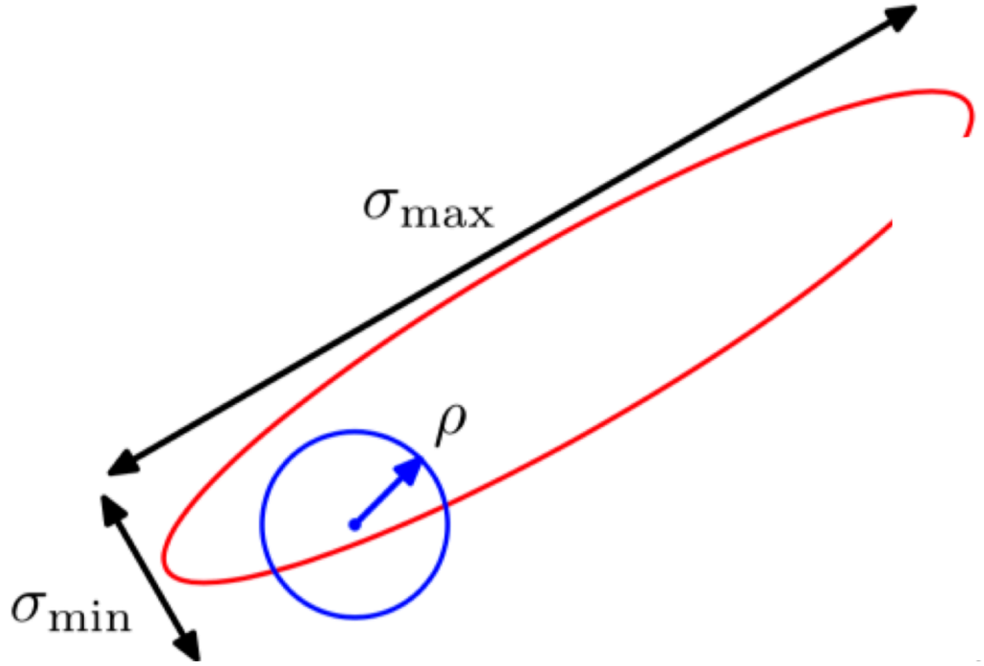
$$p(a) = p(a)p(a \to a) + p(b)p(b \to a) + p(c)p(c \to a) + p(d)p(d \to a)$$
$$= \tfrac{1}{2}\ \tfrac{2}{3} + \tfrac{1}{4}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{2}$$

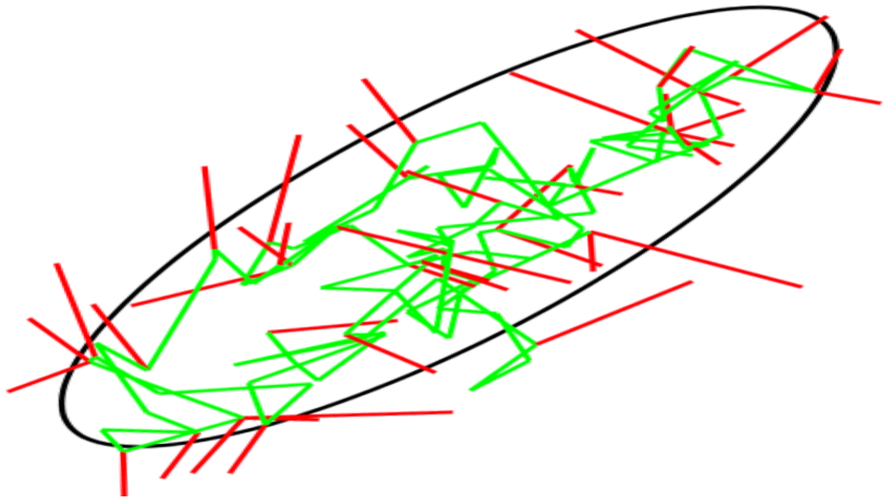
$$p(b) = p(a)p(a \to b) + p(b)p(b \to b) + p(c)p(c \to b)$$
$$= \tfrac{1}{2}\ \tfrac{1}{6} + \tfrac{1}{4}\ \tfrac{1}{2} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{4}$$

$$p(c) = p(a)p(a \to c) + p(b)p(b \to c) + p(c)p(c \to c) + p(d)p(d \to c)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{4}\ \tfrac{1}{6} + \tfrac{1}{8}\ 0 + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

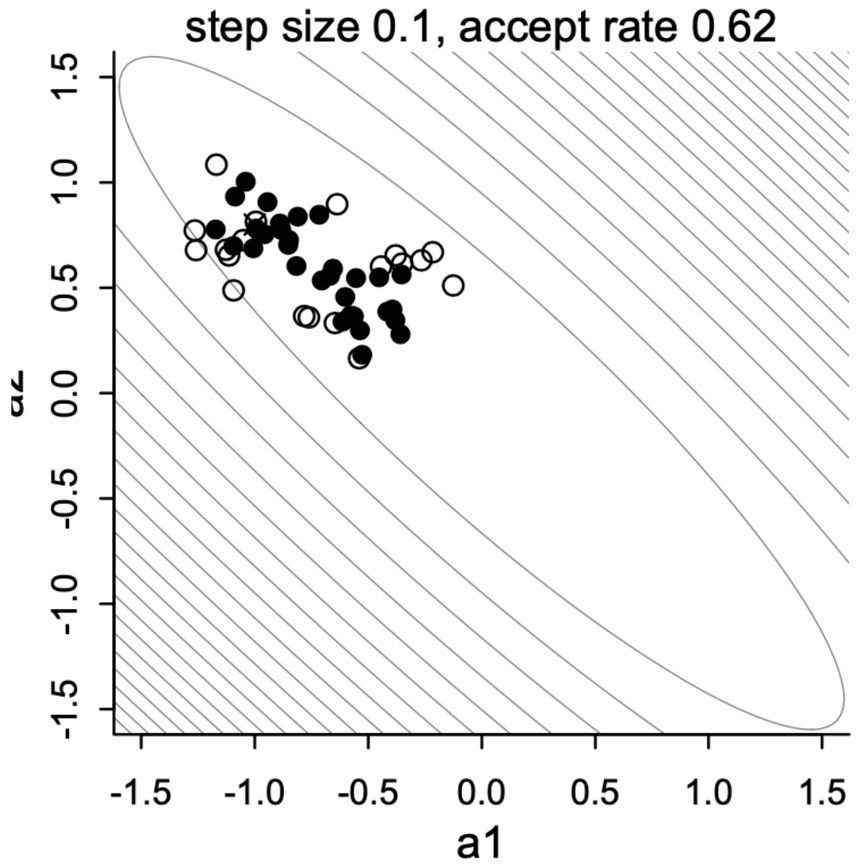
$$p(d) = p(a)p(a \to d) + p(c)p(c \to d) + p(d)p(d \to d)$$
$$= \tfrac{1}{2}\ \tfrac{1}{12} + \tfrac{1}{8}\ \tfrac{1}{3} + \tfrac{1}{8}\ \tfrac{1}{3} = \tfrac{1}{8}$$

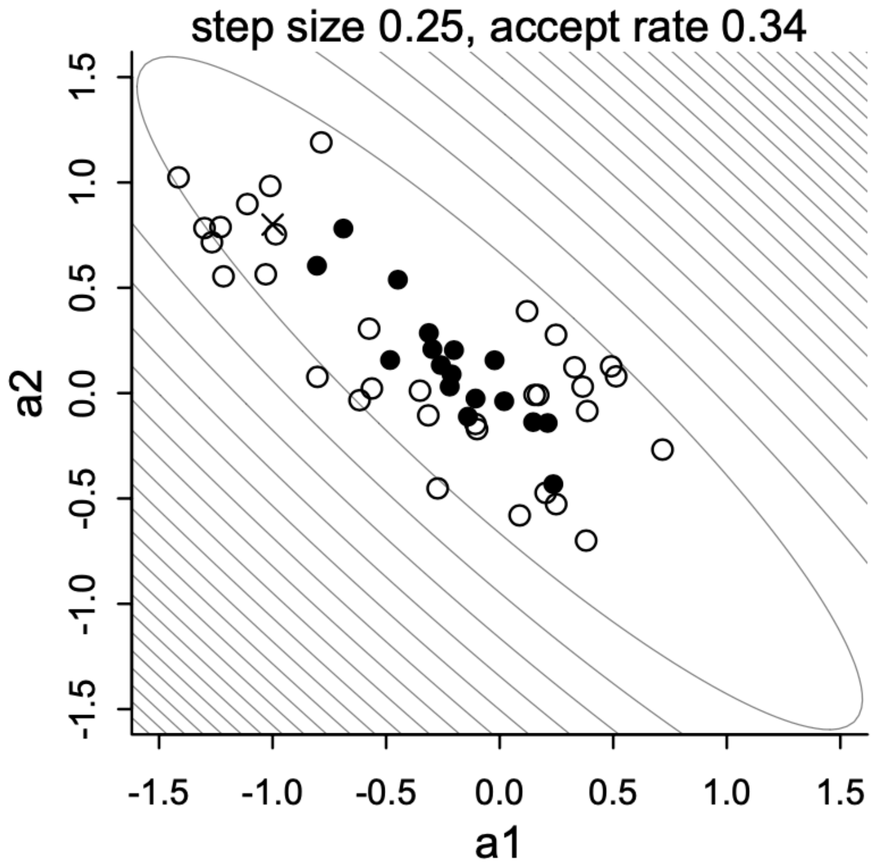
Visualization of MH: continuous case





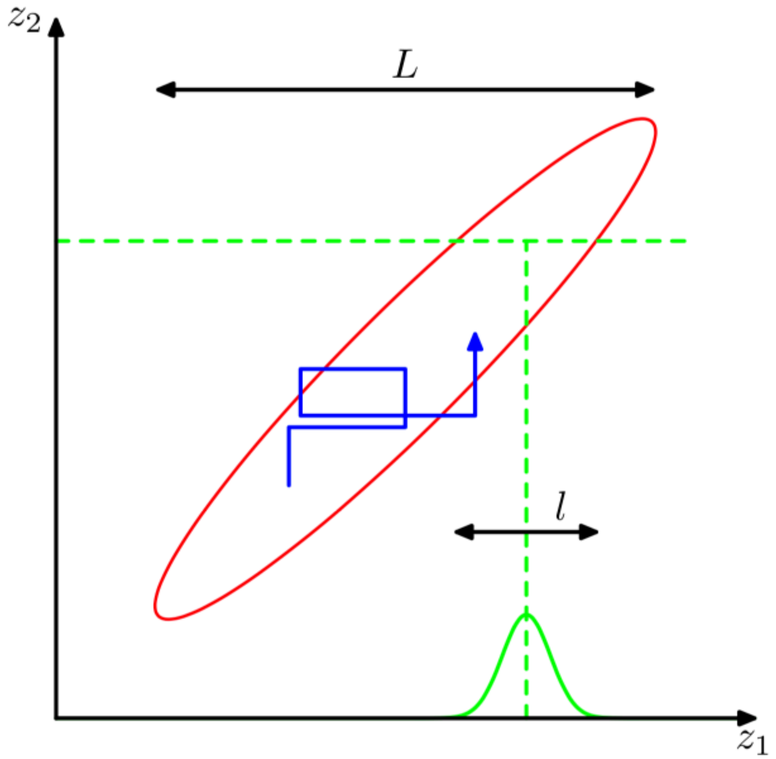
Metropolis chains under high correlation

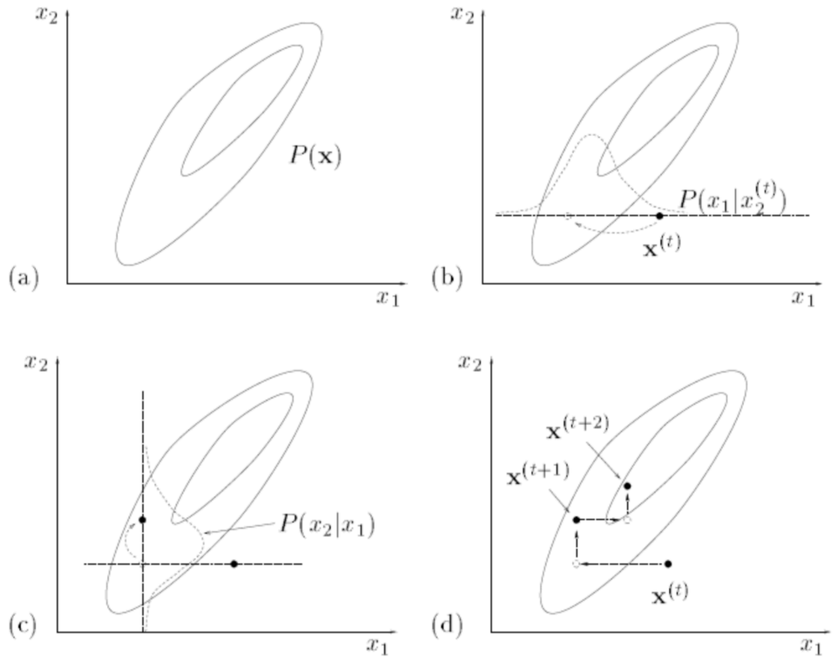




Gibbs Sampling

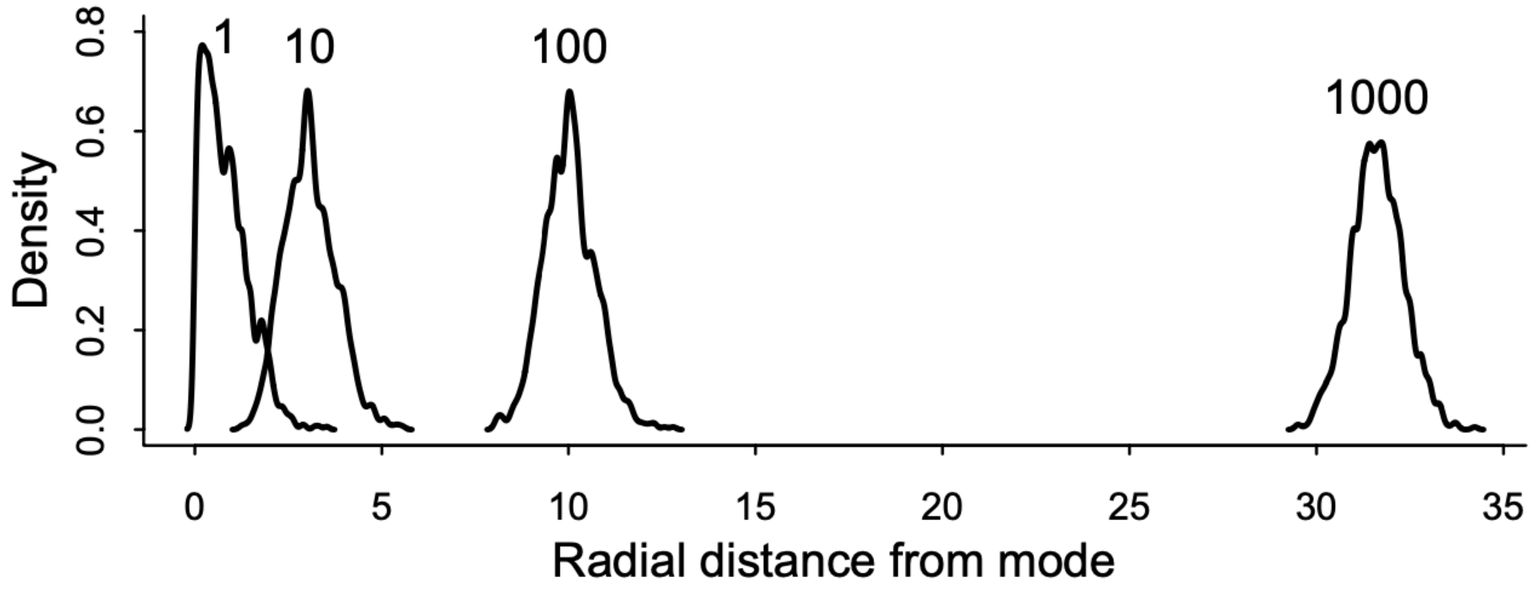
- **The idea in Gibbs sampling is that, rather than probabilistically picking the next state all at once, you make a separate probabilistic choice for one of the d dimensions, where each choice depends on the other d − 1 dimensions.**

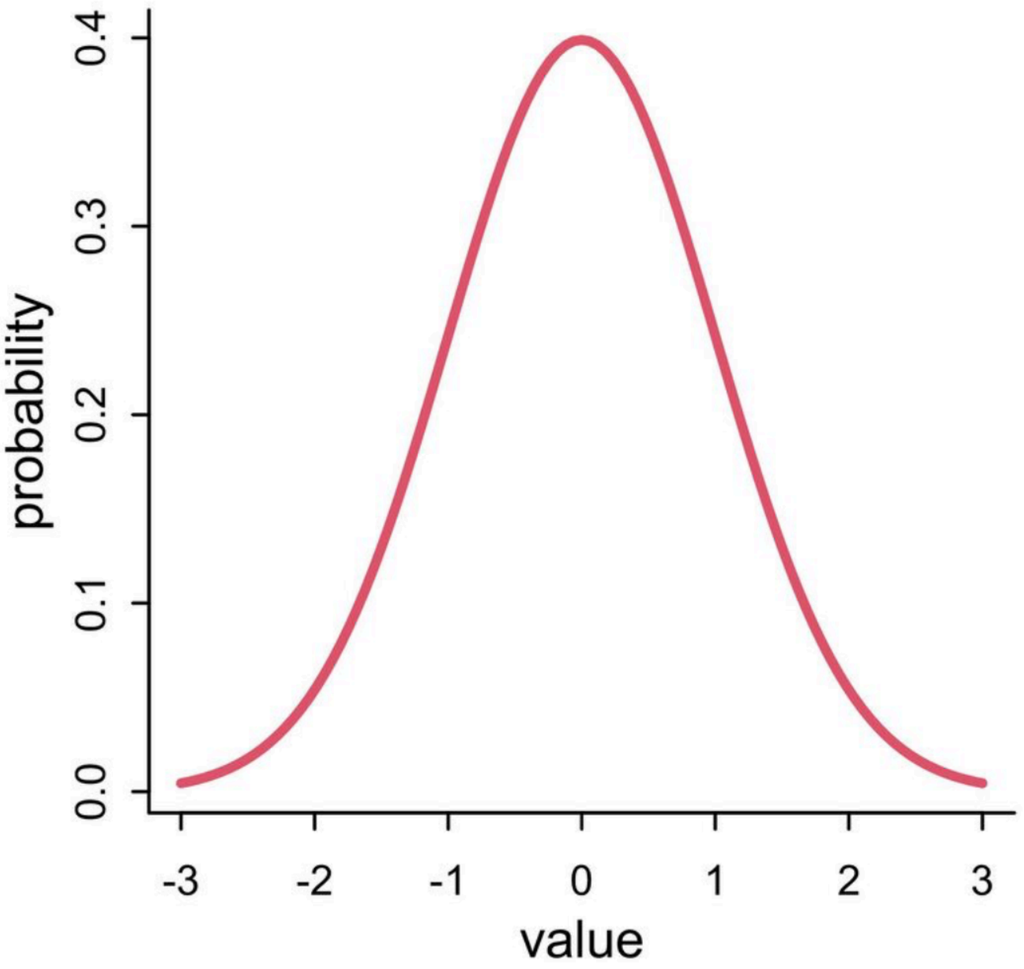


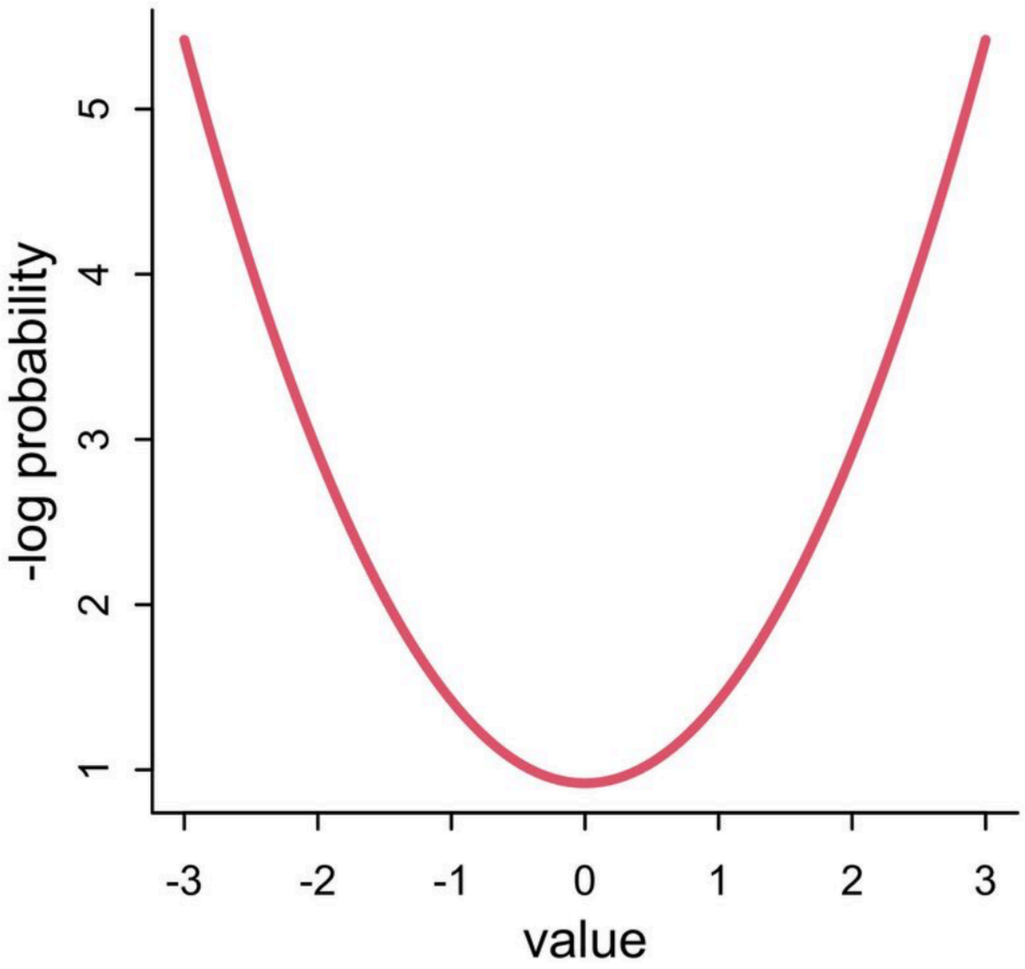


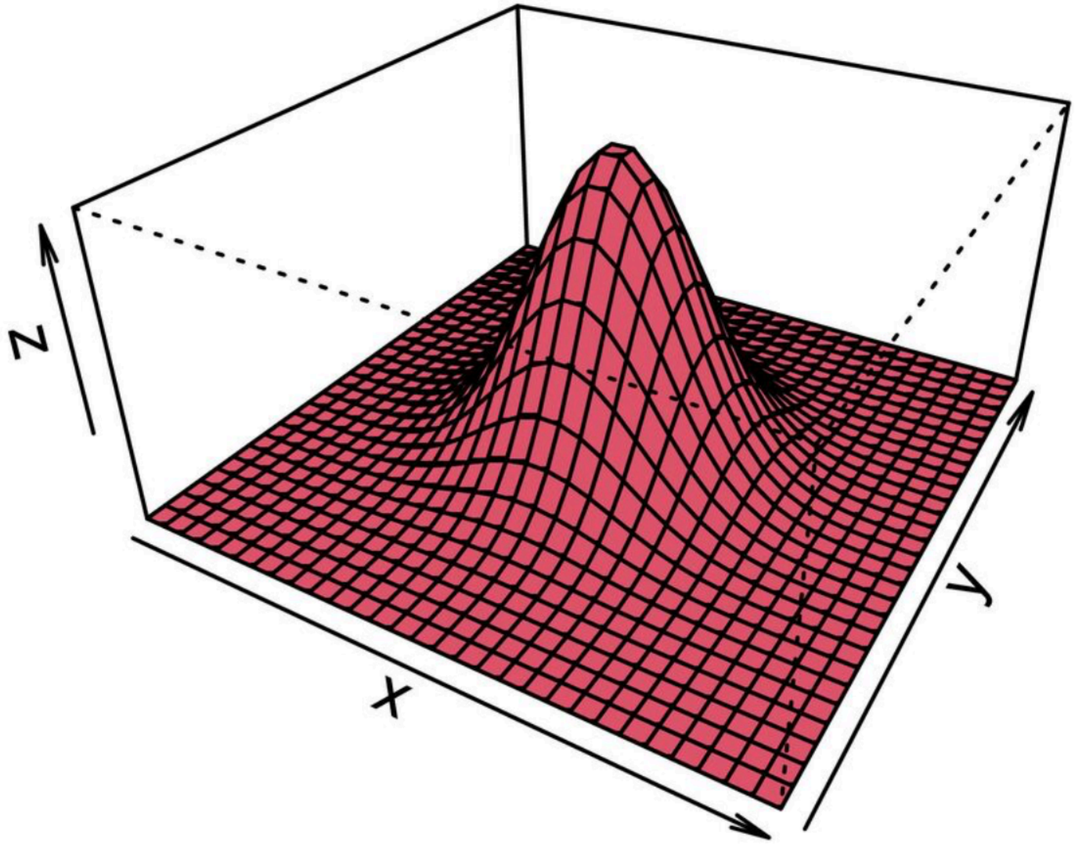
High-dimensional problems. Concentration of measure

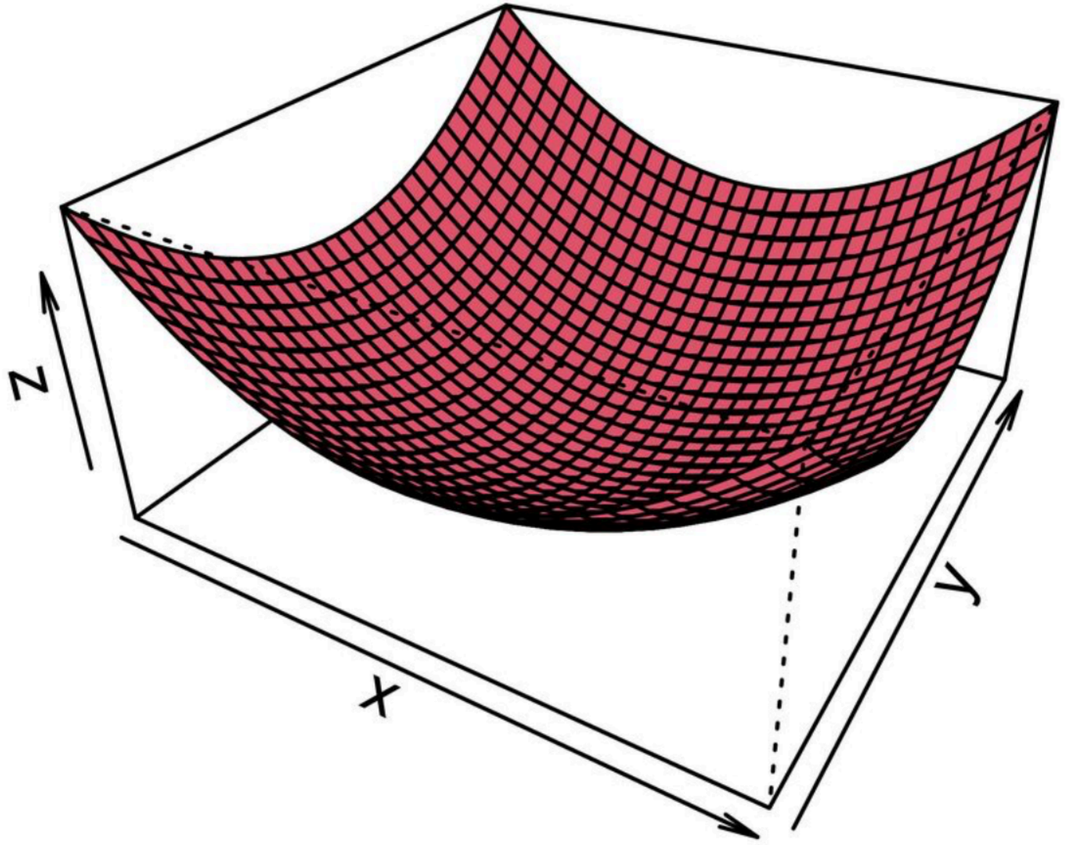
- **The most of the probability mass of a high-dimension distribution is always very far from the mode of the distribution.**
- **=> The combination of parameter values that maximizes posterior probability, the mode, is not actually in a region of parameter values that are highly plausible**




probability
0.0
0.1
0.2
0.3
0.4
-3
-2
-1
0
1
2
3
value


-log probability
1
2
3
4
5
-3
-2
-1
0
1
2
3
value

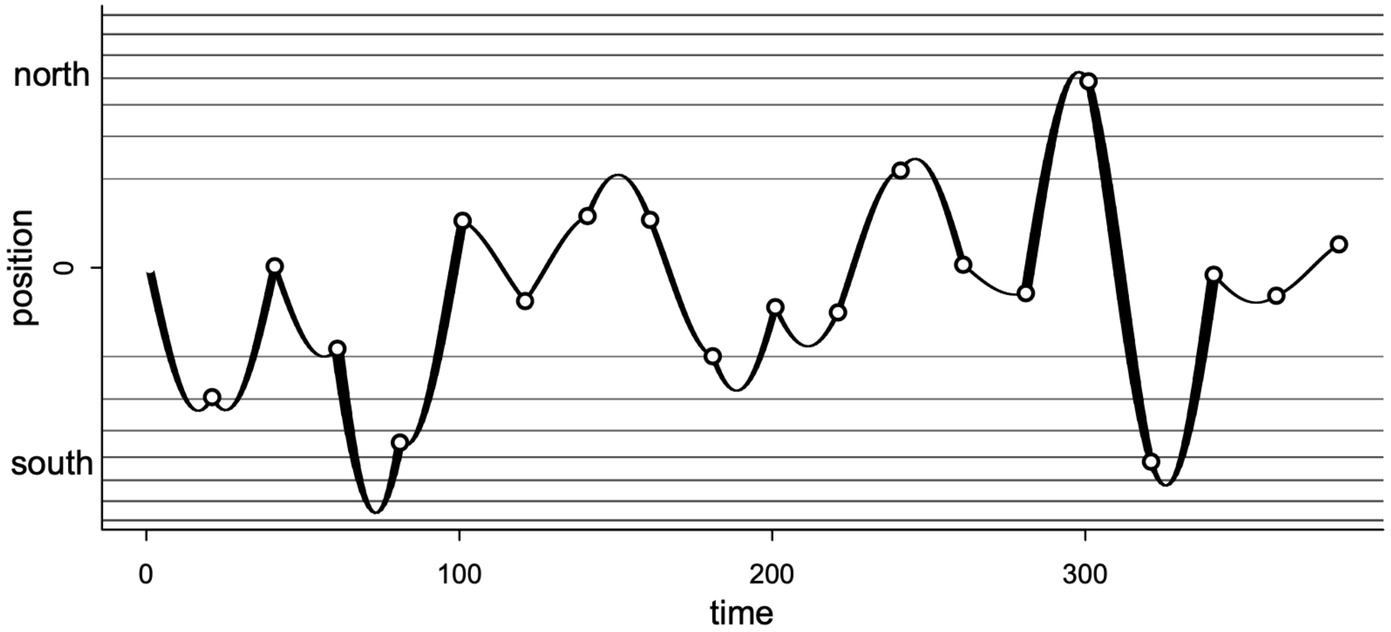

z
x
y


z
x
y


Low probability
High probability

Hamiltonian Monte Carlo (Hybrid Monte Carlo, HMC)

- HMC really does run a physics simulation, pretending the vector of parameters gives the position of a little frictionless particle.
- The log-posterior provides a surface for this particle to glide across.



Calculus is a superpower

Hamiltonian Monte Carlo needs gradients

How does it get them? Write them yourself or…

Auto-diff: Automatic differentiation

Symbolic derivatives of your model code

Used in many machine learning approaches;
“Backpropagation” is special case

$$J = \begin{bmatrix} \frac{\partial B_x}{\partial x} & \frac{\partial B_y}{\partial x} & \frac{\partial B_z}{\partial x} \\ \frac{\partial B_x}{\partial y} & \frac{\partial B_y}{\partial y} & \frac{\partial B_z}{\partial y} \\ \frac{\partial B_x}{\partial z} & \frac{\partial B_y}{\partial z} & \frac{\partial B_z}{\partial z} \end{bmatrix}$$

More calculus)

- Model:

$$x_i \sim \text{Normal}(\mu_x, 1)$$
$$y_i \sim \text{Normal}(\mu_y, 1)$$
$$\mu_x \sim \text{Normal}(0, 0.5)$$
$$\mu_y \sim \text{Normal}(0, 0.5)$$

- Where the following formula came from?

$$\sum_i \log p(y_i|\mu_y, 1) + \sum_i \log p(x_i|\mu_x, 1) + \log p(\mu_y|0, 0.5) + \log p(\mu_x, 0, 0.5)$$

Gradients, 2D Gaussian example

Log posterior:

$$\sum_i \log p(y_i|\mu_y, 1) + \sum_i \log p(x_i|\mu_x, 1) + \log p(\mu_y|0, 0.5) + \log p(\mu_x, 0, 0.5)$$

- Given we have a Gaussian density, N(y|a,b), then

$$\frac{\partial \log \mathrm{N}(y|a,b)}{\partial a} = \frac{y-a}{b^2}$$

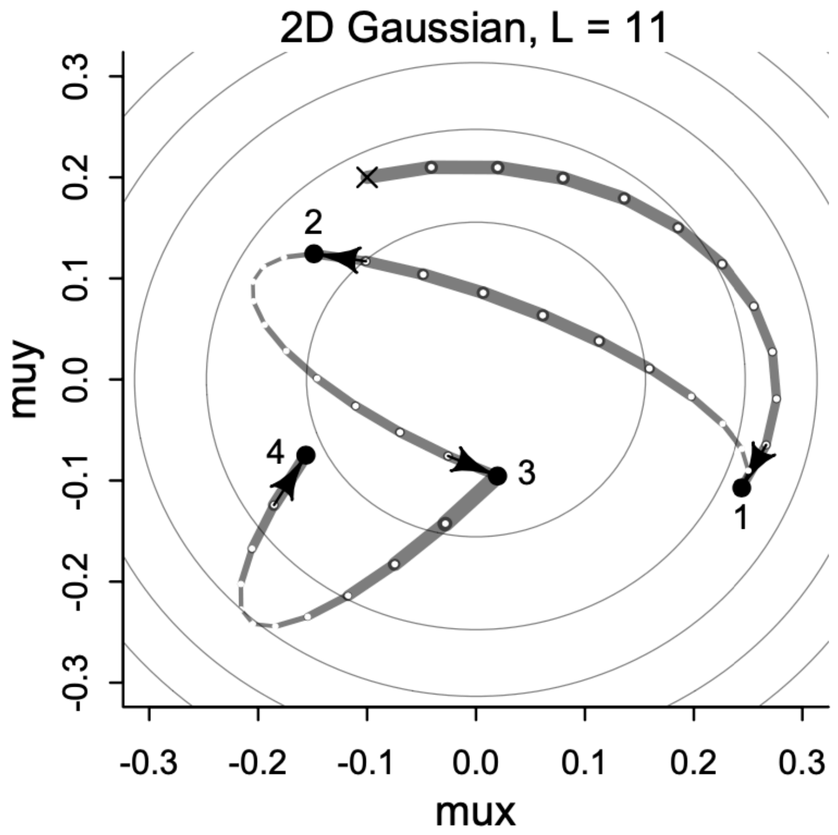
- for first dim.:

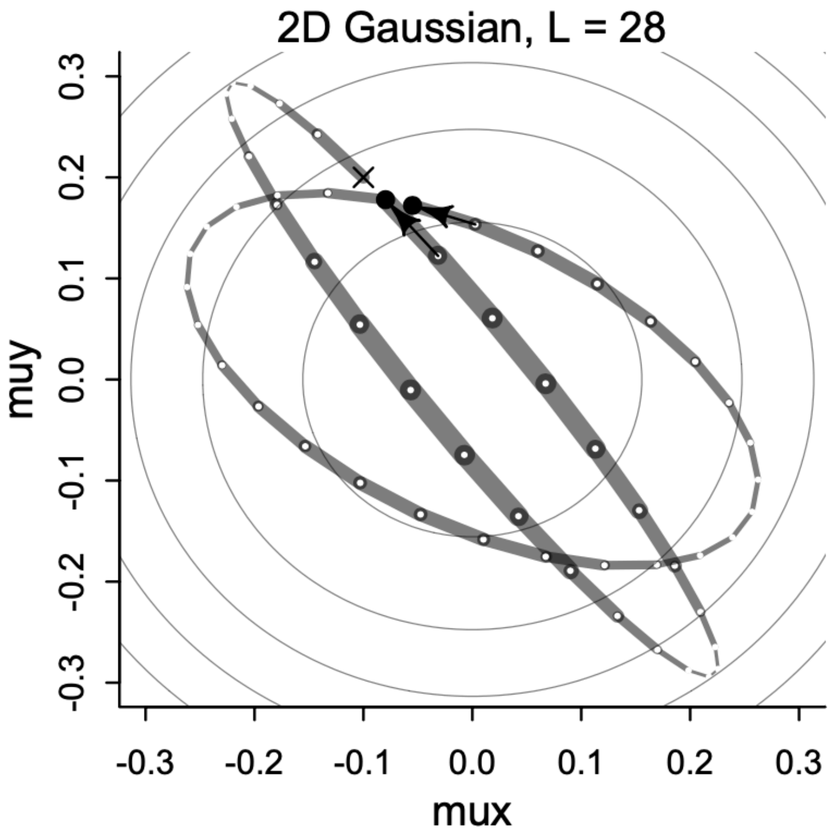
$$\frac{\partial U}{\partial \mu_x} = \frac{\partial \log \mathrm{N}(x|\mu_x, 1)}{\partial \mu_x} + \frac{\partial \log \mathrm{N}(\mu_x|0, 0.5)}{\partial \mu_x} = \sum_i \frac{x_i - \mu_x}{1^2} + \frac{0-\mu_x}{0.5^2}$$

R Code

```
# gradient function
# need vector of partial derivatives of U with respect to vector q
U_gradient <- function( q , a=0 , b=1 , k=0 , d=1 ) {
    muy <- q[1]
    mux <- q[2]
    G1 <- sum( y - muy ) + (a - muy)/b^2 #dU/dmuy
    G2 <- sum( x - mux ) + (k - mux)/d^2 #dU/dmux
    return( c( -G1 , -G2 ) ) # negative bc energy is neg-log-prob
}
# test data
set.seed(7)
y <- rnorm(50)
x <- rnorm(50)
x <- as.numeric(scale(x))
y <- as.numeric(scale(y))
```

U-turns


2D Gaussian, L = 11
muy
mux
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
1
2
3
4


2D Gaussian, L = 28
muy
mux
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3

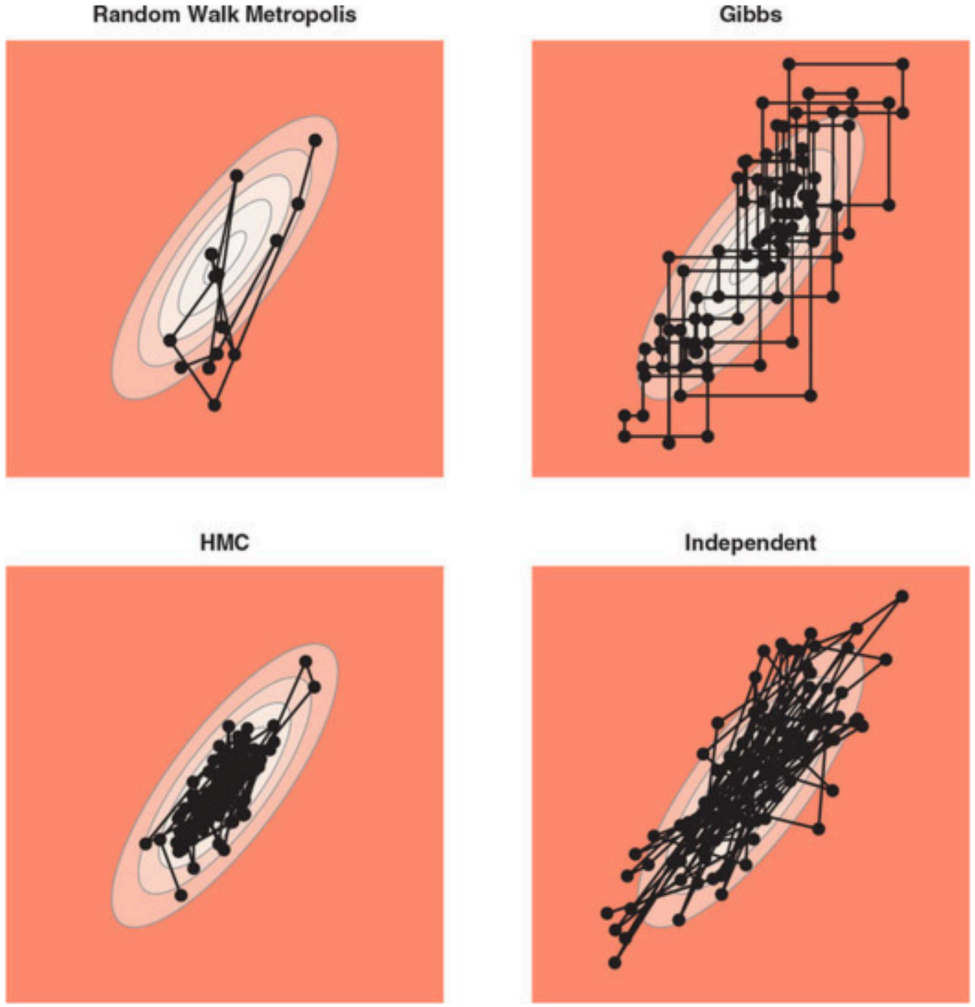
mc-stan.org

Stan

About Stan

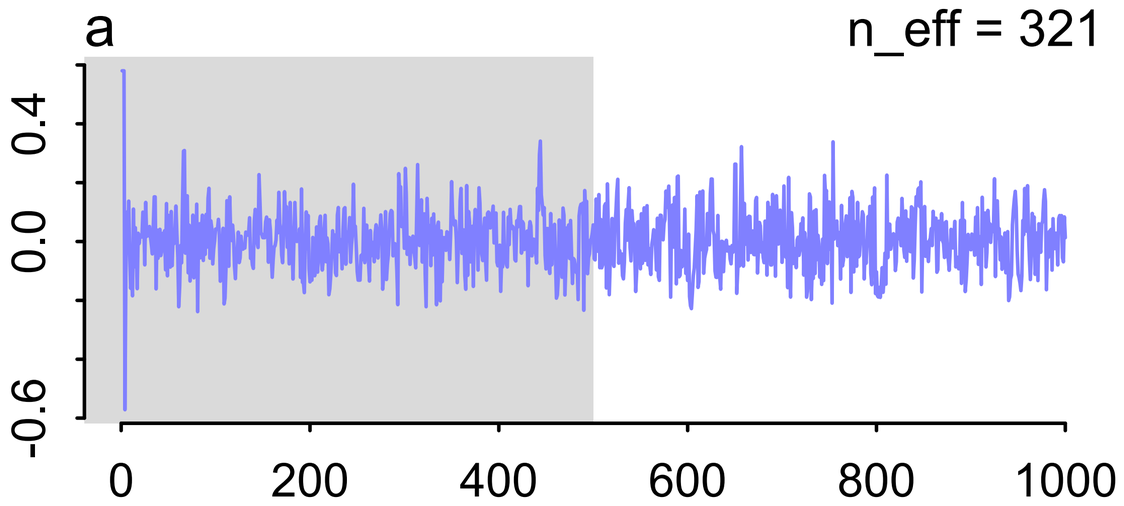
Stan is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

Different Sampling methods

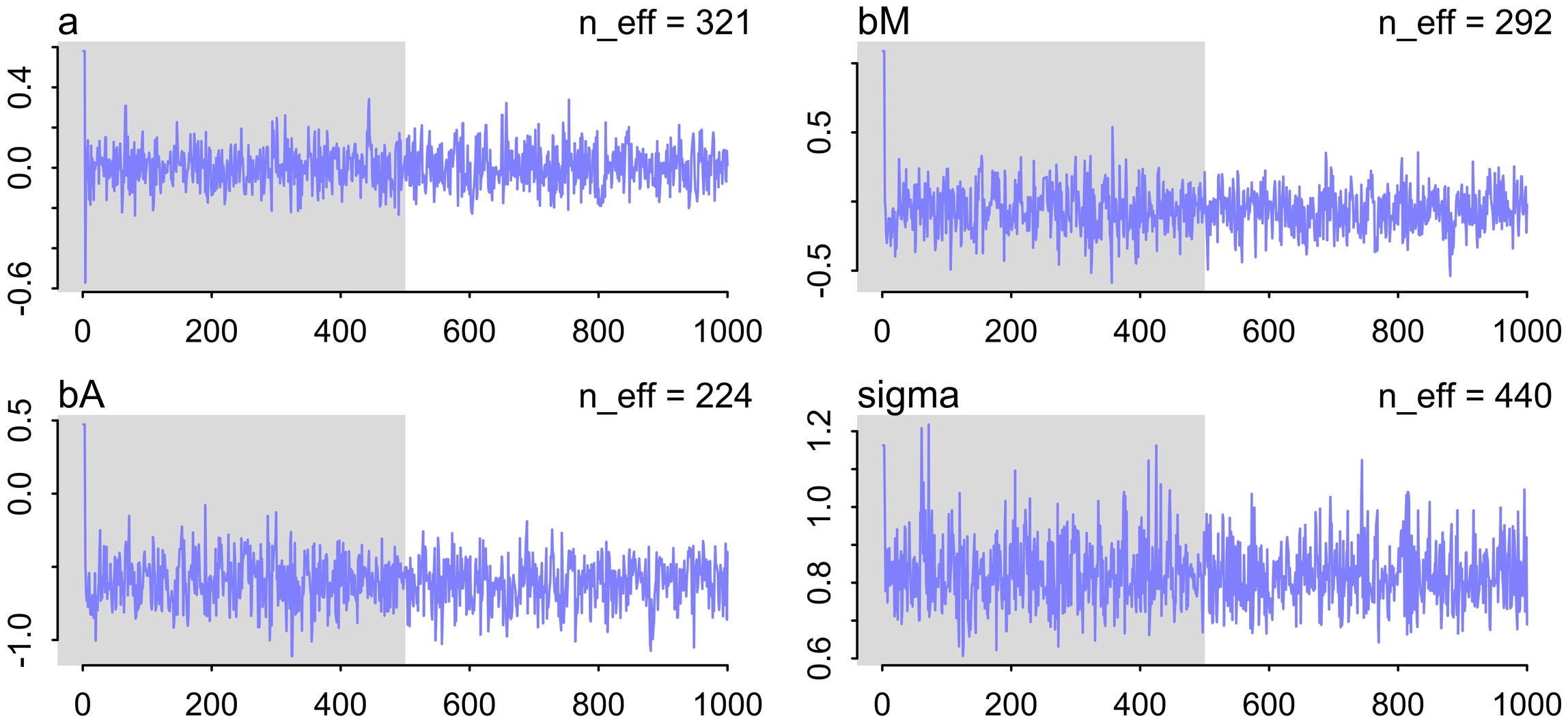


- Lambert, Ben. A Students Guide to Bayesian Statistics SAGE Publications. 2018

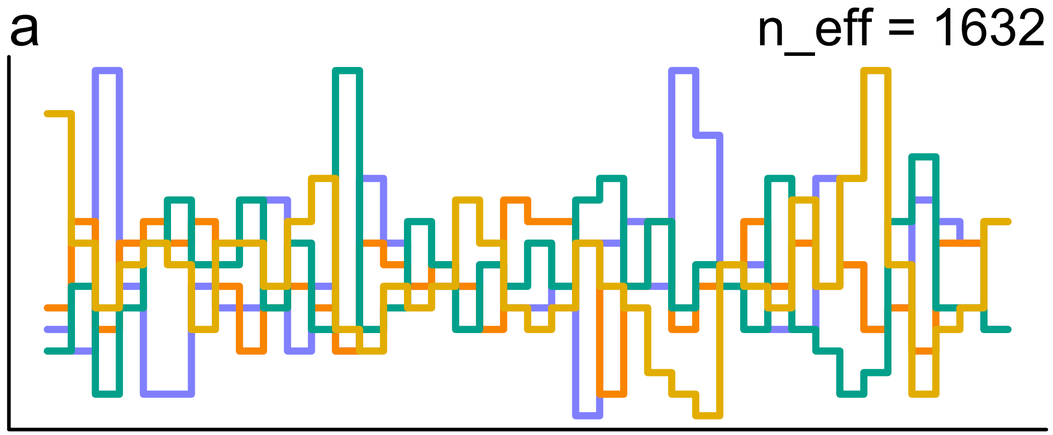
Trace plots



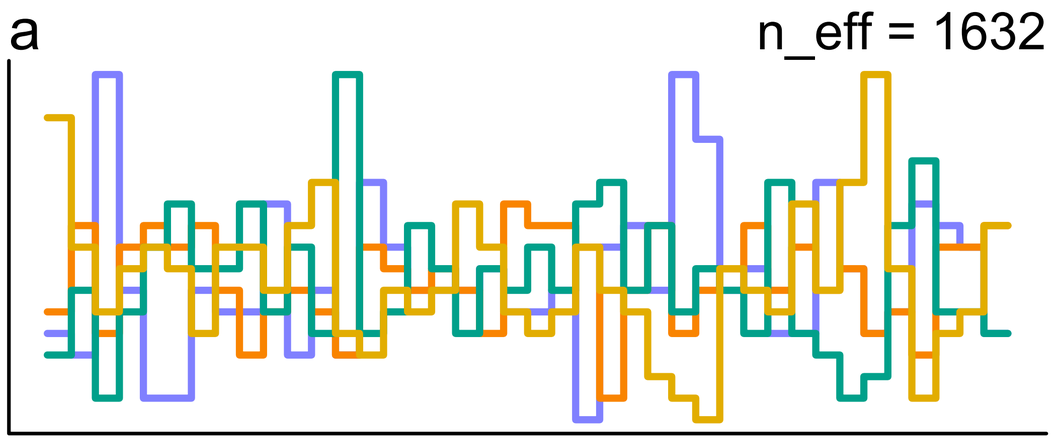
Trace plots

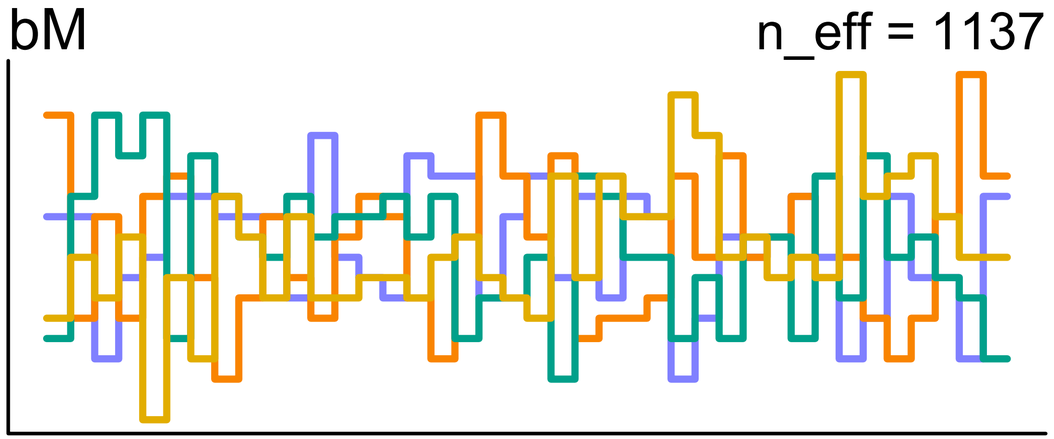

a
n_eff = 321
0.4
0.0
-0.6
0
200
400
600
800
1000
bM
n_eff = 292
0.5
-0.5
0
200
400
600
800
1000
bA
n_eff = 224
0.5
0.0
-1.0
0
200
400
600
800
1000
sigma
n_eff = 440
1.2
1.0
0.8
0.6
0
200
400
600
800
1000

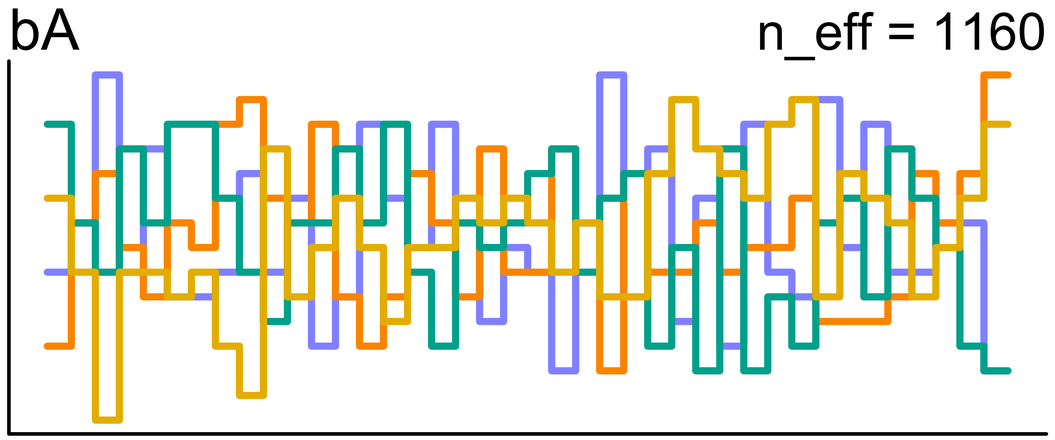
Trace rank (Trank) plots

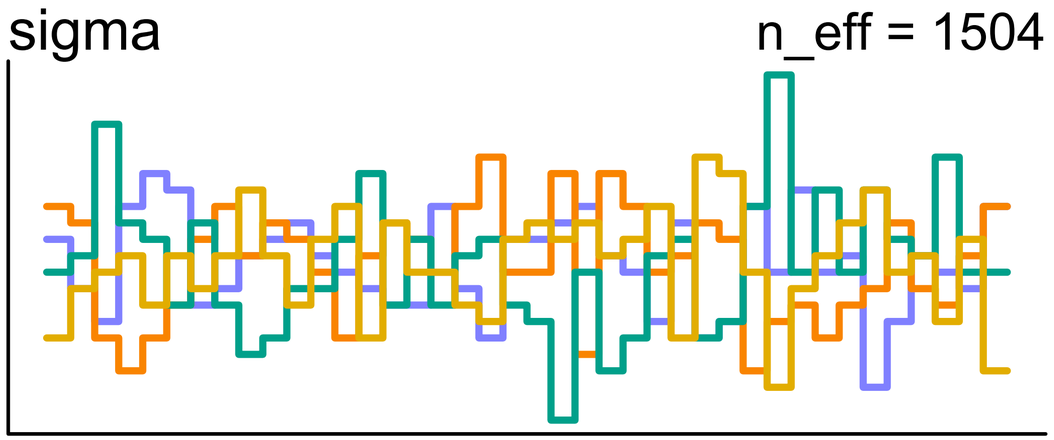


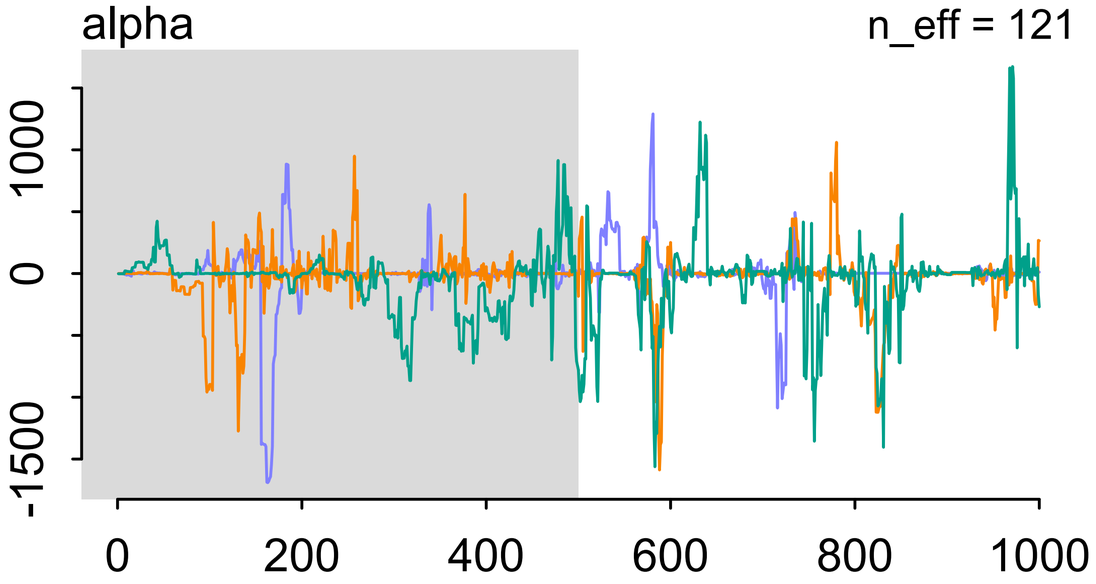
Trace rank (Trank) plots

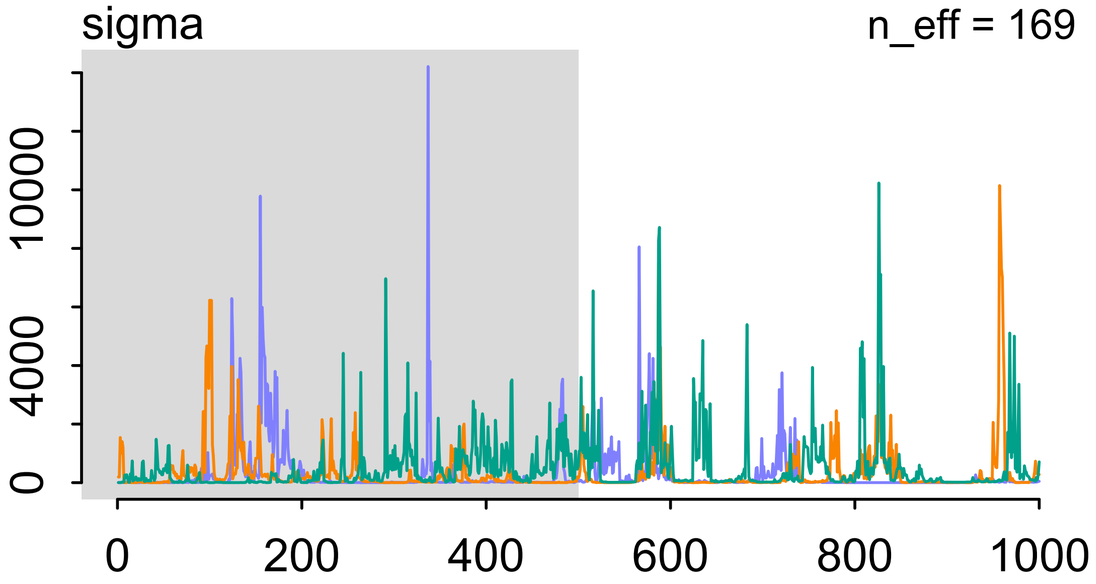


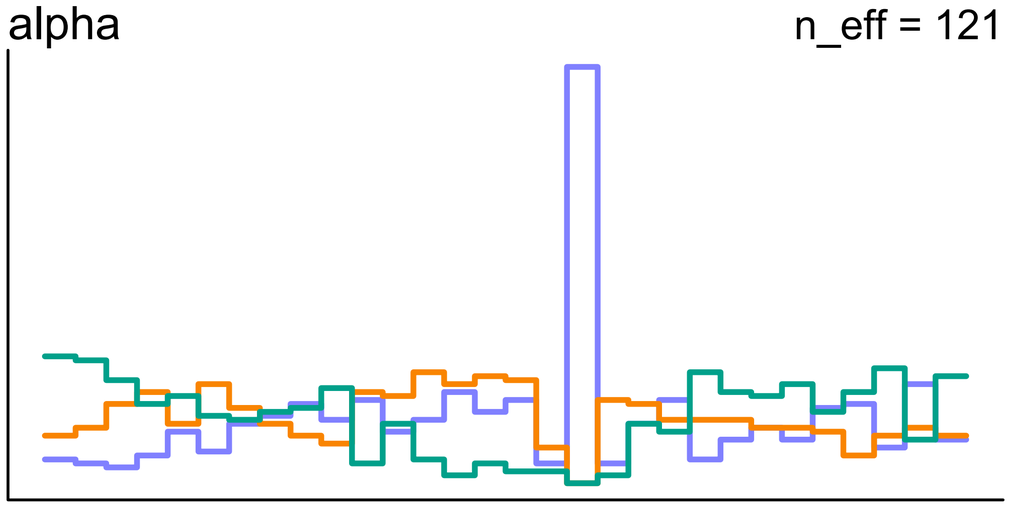


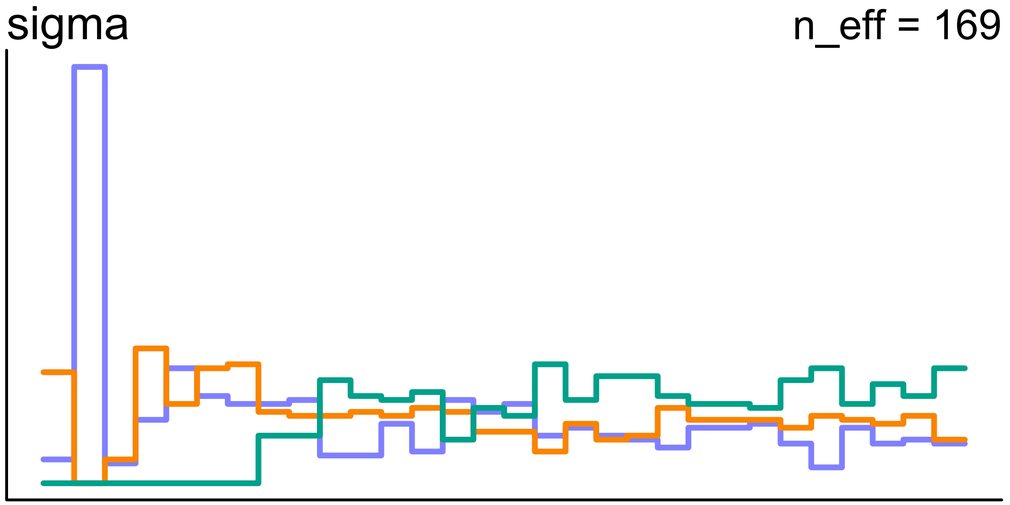





alpha
n_eff = 121
1000
0
-1500
0
200
400
600
800
1000


sigma
n_eff = 169
10000
4000
0
0
200
400
600
800
1000


alpha
n_eff = 121


sigma
n_eff = 169

R-hat

When chains converge:

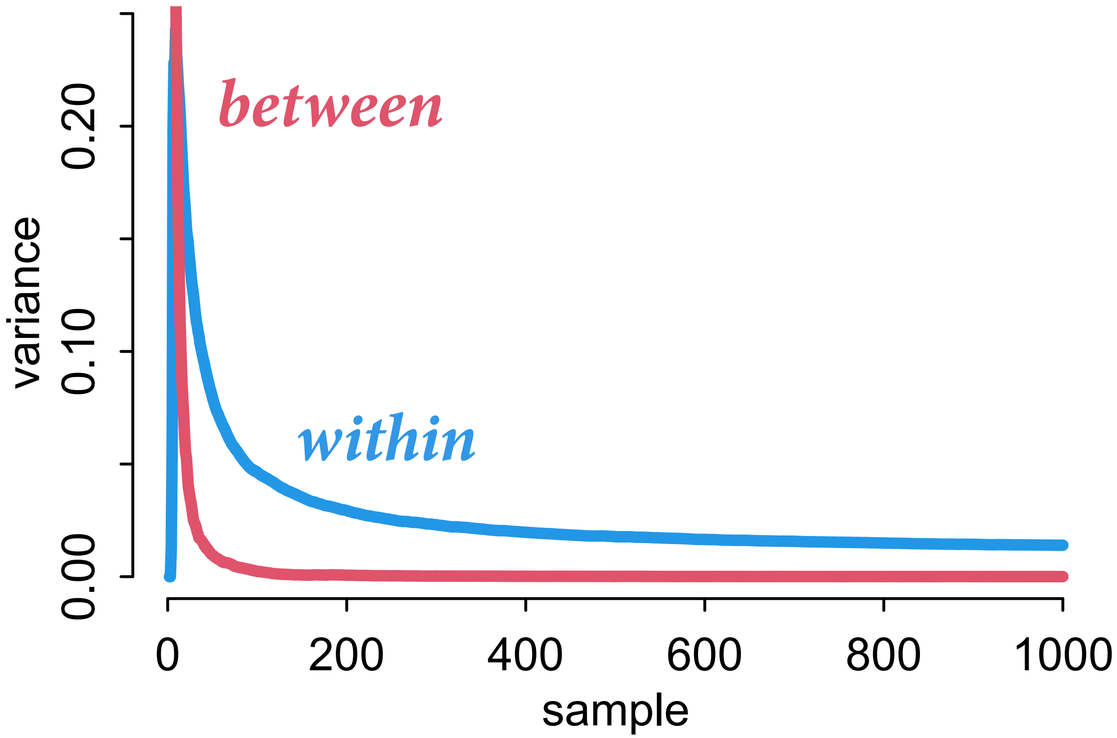
(1) Start and end of each chain explores same region

(2) Independent chains explore same region

R-hat is a ratio of variances:
As total variance shrinks to average variance within chains, R-hat approaches 1

NO GUARANTEES; NOT A TEST

```
> precis(mHMC)
        mean   sd  5.5% 94.5% n_eff Rhat4
a       0.00 0.10 -0.16  0.16  1632     1
bM     -0.06 0.17 -0.32  0.21  1137     1
bA     -0.61 0.17 -0.86 -0.34  1160     1
sigma   0.83 0.09  0.70  0.99  1504     1
>
```



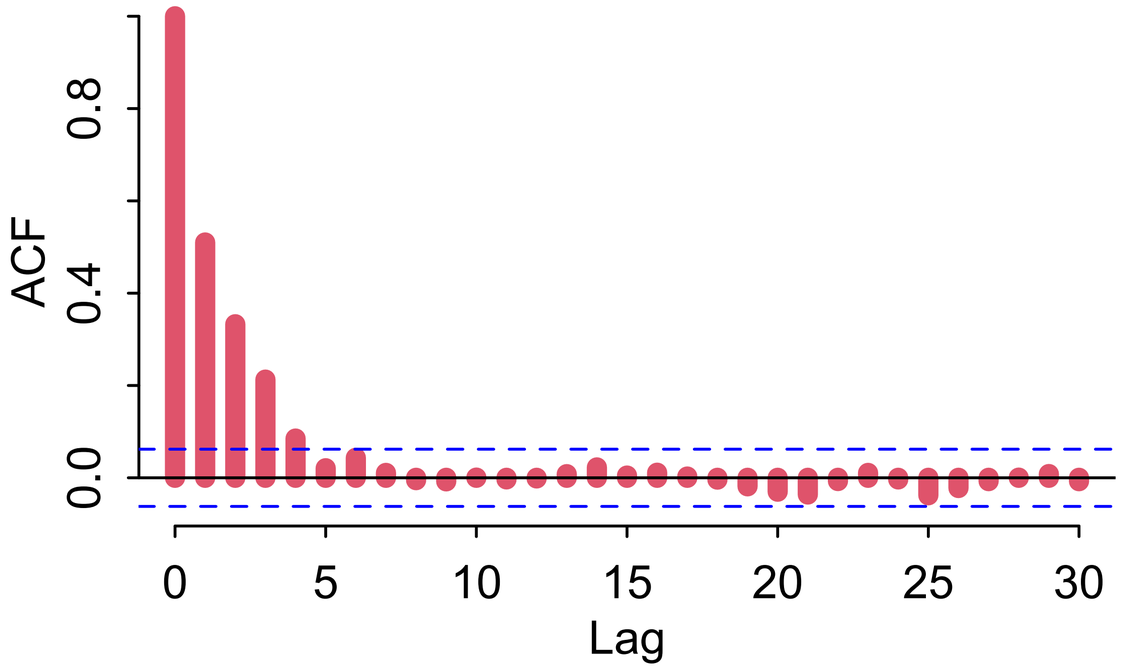
n_eff

Estimate of number of effective samples

"How long would the chain be, if each sample was independent of the one before it?"

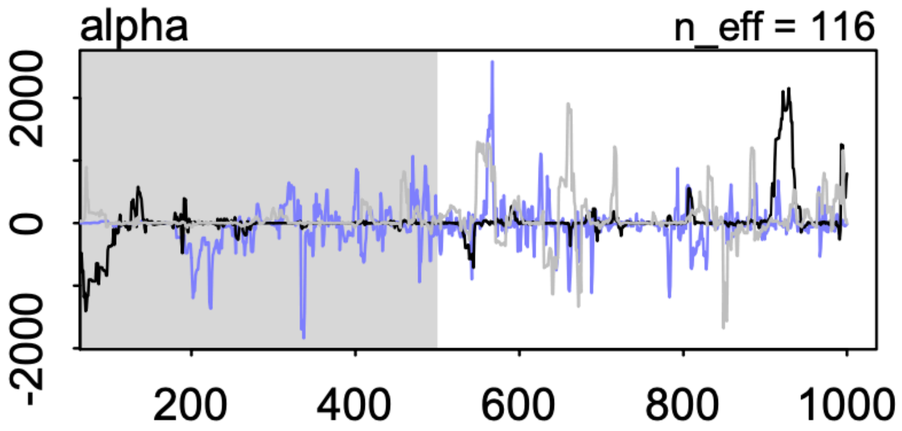
When samples are autocorrelated, you have fewer *effective* samples

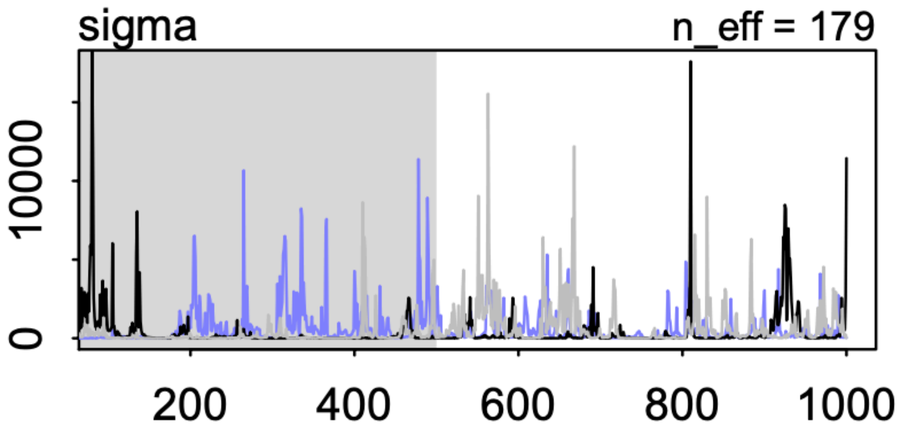
```
> precis(mHMC)
        mean   sd  5.5% 94.5% n_eff Rhat4
a       0.00 0.10 -0.16  0.16  1632     1
bM     -0.06 0.17 -0.32  0.21  1137     1
bA     -0.61 0.17 -0.86 -0.34  1160     1
sigma   0.83 0.09  0.70  0.99  1504     1
>
```

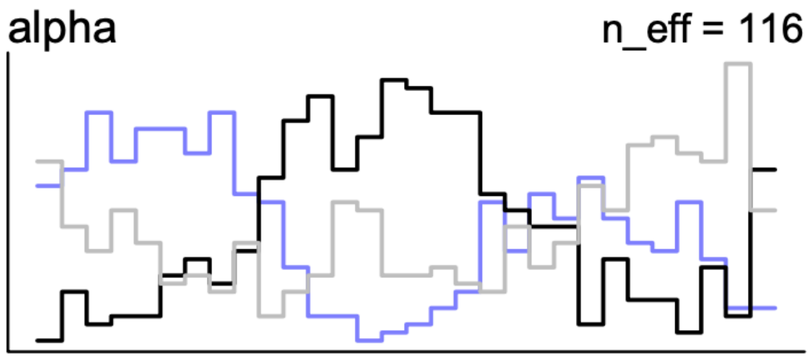


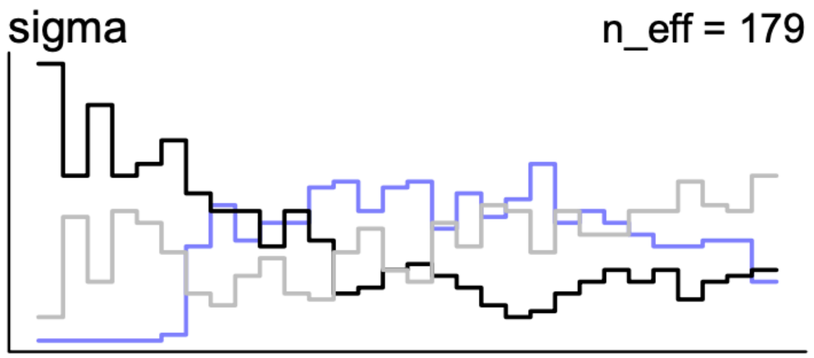
MCMC. Diagnosis of trace and trank plots

- Not healthy ?



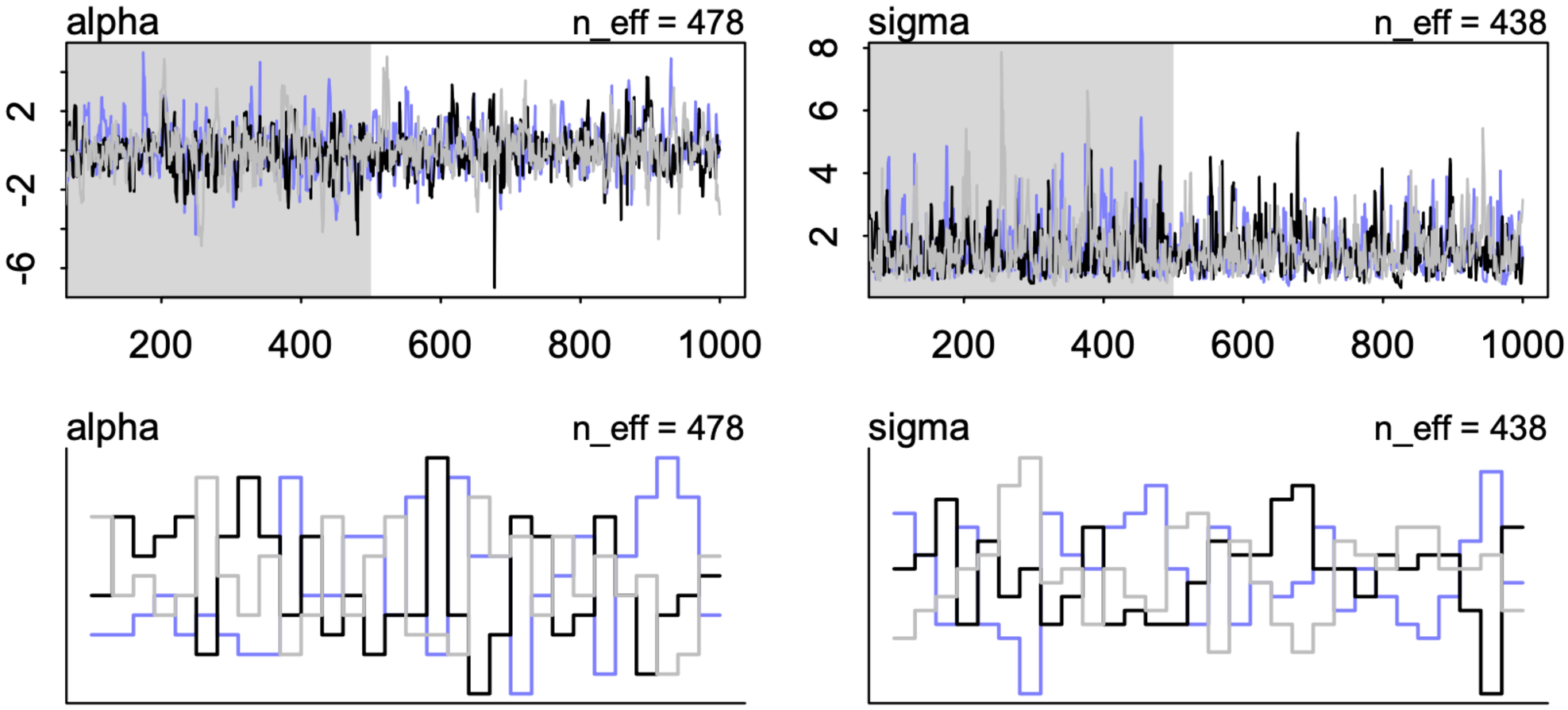






MCMC. Diagnosis of trace and trank plots

- Healthy ?



Summary

- MCMC has been a revolution in scientific computing
- Custom scientific modeling
- High-dimension
- Do not “pipette by mouth”