



UPPSALA  
UNIVERSITET

• Introduction

# Bayesian Statistics and Data Analysis

## Lecture 5

Måns Magnusson

Department of Statistics, Uppsala University  
Thanks to Aki Vehtari, Aalto University



UPPSALA  
UNIVERSITET

● Introduction

## Section 1

### Introduction



# It's all about expectations

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta)p(\theta|y)d\theta,$$

$$\text{where } p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$$

We can easily evaluate  $p(y|\theta)p(\theta)$  for any  $\theta$ , but the integral  $\int p(y|\theta)p(\theta)d\theta$  is usually difficult.

We can use the unnormalized posterior  $q(\theta|y) = p(y|\theta)p(\theta)$ , for example, in

- Grid (equal spacing) evaluation with self-normalization

$$E_{p(\theta|y)}[f(\theta)] \approx \frac{\sum_{s=1}^S [f(\theta^{(s)})q(\theta^{(s)}|y)]}{\sum_{s=1}^S q(\theta^{(s)}|y)}$$

- Monte Carlo methods which can sample from  $p(\theta^{(s)}|y)$  using only  $q(\theta^{(s)}|y)$

$$E_{p(\theta|y)}[f(\theta)] \approx \frac{1}{S} \sum_{s=1}^S f(\theta^{(s)})$$



• Introduction

- Monte Carlo methods we have discussed so far
  - Inverse CDF works for 1D
  - Analytic transformations work for only certain distributions
  - Factorization works only for certain joint distributions
  - Grid evaluation and sampling works in less than a few dimensions
  - Rejection sampling works mostly in 1D (truncation is a special case)
  - Importance sampling is reliable only in special cases
- What to do in high dimensions?
  - Markov chain Monte Carlo (Ch 11-12)
  - Laplace, Variational\*, EP\* (Ch 4,13\*)



- Introduction

- Andrey Markov proved weak law of large numbers and central limit theorem for certain dependent-random sequences, which were later named Markov chains
- The probability of each event depends only on the state attained in the previous event (or finite number of previous events)
- Markov's one example was the sequence of letters in Pushkin's novel "Yevgeniy Onegin"



UPPSALA  
UNIVERSITET

# Markov chain

---

- Introduction

- Example of a simple Markov chain on black board



• Introduction

- Produce draws  $\theta^{(t)}$  given  $\theta^{(t-1)}$  from a Markov chain, which has been constructed so that its equilibrium distribution is  $p(\theta|y)$ 
  - + generic
  - + combine sequence of easier Monte Carlo draws to form a Markov chain
  - + chain goes where most of the posterior mass is
  - + asymptotically chain spends the  $\alpha\%$  of time where  $\alpha\%$  posterior mass is
  - + central limit theorem holds for expectations
    - draws are dependent
    - construction of efficient Markov chains is not always easy



- Introduction

- Set of random variables  $\theta^1, \theta^2, \dots$ , so that with all values of  $t$ ,  $\theta^t$  depends only on the previous  $\theta^{(t-1)}$

$$p(\theta^t | \theta^1, \dots, \theta^{(t-1)}) = p(\theta^t | \theta^{(t-1)})$$

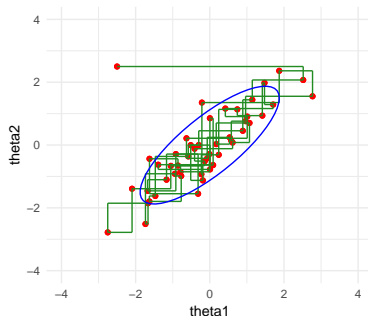
- Chain has to be initialized with some starting point  $\theta^0$
- Transition distribution  $T_t(\theta^t | \theta^{t-1})$  (may depend on  $t$ )
- By choosing a suitable transition distribution, the stationary distribution of Markov chain is  $p(\theta | y)$





# Gibbs sampling

- Alternate sampling from 1D conditional distributions
  - 1D is easy even if no conjugate prior and analytic posterior
- demo11\_1



• Draws — Steps of the sampler — 90% HPD

- Basic algorithm

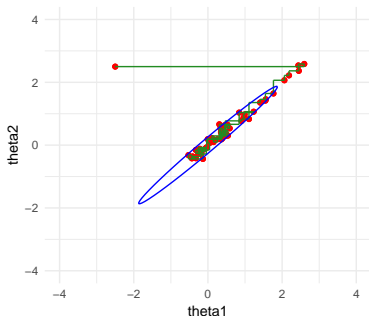
sample  $\theta_j^t$  from  $p(\theta_j | \theta_{-j}^{t-1}, y)$ ,

where  $\theta_{-j}^{t-1} = (\theta_1^t, \dots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \dots, \theta_d^{t-1})$



# Gibbs sampling

- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
  - BUGS/WinBUGS/OpenBUGS/JAGS
- No algorithm parameters to tune (cf. proposal distribution in Metropolis algorithm)
- For not so easy conditionals, use e.g. inverse-CDF
- Several parameters can be updated in blocks (*blocking*)
- Slow if parameters are highly dependent in the posterior
  - demo11\_1 continues



• Draws — Steps of the sampler — 90% HPD



UPPSALA  
UNIVERSITET

# Conditional vs joint

---

- Introduction

- How about sampling  $\theta$  jointly?
  - e.g. it is easy to sample from multivariate normal
- Can we use that to form a Markov chain?  
<http://eleventh.org/blog/2017/11/28/build-a-better-markov-chain/>



# Metropolis algorithm

- Algorithm

- starting point  $\theta^0$
- $t = 1, 2, \dots$ 
  - pick a proposal  $\theta^*$  from the proposal distribution  $J_t(\theta^*|\theta^{t-1})$ .  
Proposal distribution has to be symmetric, i.e.  
 $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$ , for all  $\theta_a, \theta_b$
  - calculate acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$

(c) set

$$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

- ie, if  $p(\theta^*|y) > p(\theta^{t-1}|y)$  accept the proposal always and otherwise accept the proposal with probability  $r$
- rejection of a proposal increments the time  $t$  also by one ie, the new state is the same as previous
- step c is executed by generating a random number from  $\mathcal{U}(0, 1)$
- $p(\theta^*|y)$  and  $p(\theta^{t-1}|y)$  have the same normalization terms, and thus instead of  $p(\cdot|y)$ , unnormalized  $q(\cdot|y)$  can be used, as the normalization terms cancel out!



- Introduction

- Example: one bivariate observation  $(y_1, y_2)$ 
  - bivariate normal distribution with unknown mean and known covariance

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \middle| y \sim \mathcal{N} \left( \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

- proposal distribution  $J_t(\theta^* | \theta^{t-1}) = \mathcal{N}(\theta^* | \theta^{t-1}, \sigma_p^2)$
- Demo <http://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/>



# Why Metropolis algorithm works

---

- Introduction

- Intuitively more draws from the higher density areas as jumps to higher density are always accepted and only some of the jumps to the lower density are accepted
- Theoretically
  1. Prove that simulated series is a Markov chain which has unique stationary distribution
  2. Prove that this stationary distribution is the desired target distribution



# Why Metropolis algorithm works

---

1. Prove that simulated series is a Markov chain which has unique stationary distribution
  - a) irreducible
    - = positive probability of eventually reaching any state from any other state
  - b) aperiodic
    - = aperiodic (return times are not periodic)
    - holds for a random walk on any proper distribution (except for trivial exceptions)
  - c) recurrent / not transient
    - = probability to return to a state  $i$  is 1
    - holds for a random walk on any proper distribution (except for trivial exceptions)



## Why Metropolis algorithm works

### 2. Prove that this stationary distribution is the desired target distribution $p(\theta|y)$

- consider starting algorithm at time  $t - 1$  with a draw  $\theta^{t-1} \sim p(\theta|y)$
- consider any two such points  $\theta_a$  and  $\theta_b$  drawn from  $p(\theta|y)$  and labeled so that  $p(\theta_b|y) \geq p(\theta_a|y)$
- the unconditional probability density of a transition from  $\theta_a$  to  $\theta_b$  is

$$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b) = p(\theta_a|y)J_t(\theta_b|\theta_a),$$

- the unconditional probability density of a transition from  $\theta_b$  to  $\theta_a$  is

$$\begin{aligned} p(\theta^t = \theta_a, \theta^{t-1} = \theta_b) &= p(\theta_b|y)J_t(\theta_a|\theta_b) \left( \frac{p(\theta_a|y)}{p(\theta_b|y)} \right) \\ &= p(\theta_a|y)J_t(\theta_a|\theta_b), \end{aligned}$$

which is the same as the probability of transition from  $\theta_a$  to  $\theta_b$ , since we have required that  $J_t(\cdot|\cdot)$  is symmetric

- since their joint distribution is symmetric,  $\theta^t$  and  $\theta^{t-1}$  have the same marginal distributions, and so  $p(\theta|y)$  is the stationary distribution of the Markov chain of  $\theta$





- Introduction

- Generalization of Metropolis algorithm for non-symmetric proposal distributions
  - acceptance ratio includes ratio of proposal distributions

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{t-1})}{p(\theta^{t-1}|y)/J_t(\theta^{t-1}|\theta^*)} = \frac{p(\theta^*|y)J_t(\theta^{t-1}|\theta^*)}{p(\theta^{t-1}|y)J_t(\theta^*|\theta^{t-1})}$$



# Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$  for all  $\theta$
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or  $t$  distribution is used
- After the shape has been selected, it is important to select the scale
  - small scale
    - many steps accepted, but the chain moves slowly due to small steps
  - big scale
    - long steps proposed, but many of those rejected and again chain moves slowly
- Generic rule for rejection rate is 60-90% (but depends on dimensionality and a specific algorithm variation)



UPPSALA  
UNIVERSITET

# Gibbs sampling

---

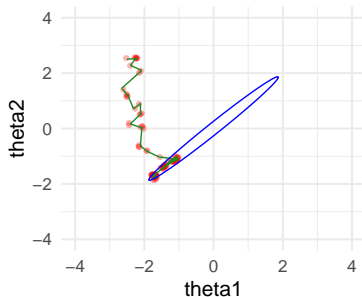
- Introduction

- Specific case of Metropolis-Hastings algorithm
  - single updated (or blocked)
  - proposal distribution is the conditional distribution
    - proposal and target distributions are same
    - acceptance probability is 1

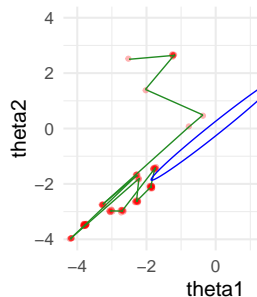


# Metropolis

- Usually doesn't scale well to high dimensions
  - if the shape doesn't match the whole distribution, the efficiency drops
  - demo11\_2



• Draws — Steps of the sampler — 90% HPD



• Draws — Steps of the sampler — 90% HPD



# Dynamic Hamiltonian Monte Carlo and NUTS

---

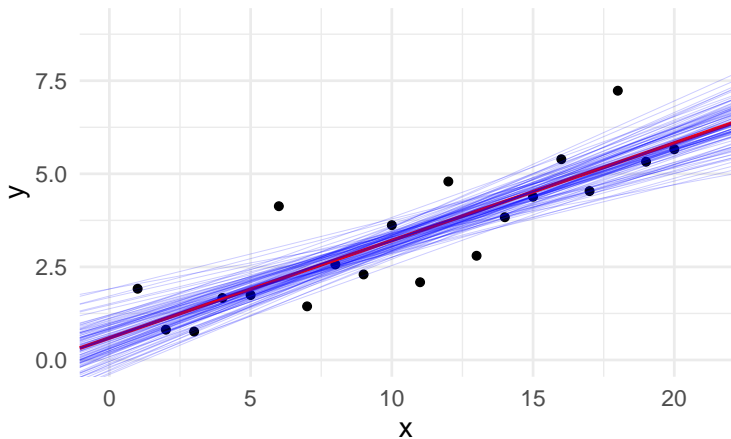
- Introduction

- Chapter 12 presents some more advanced methods
  - Chapter 12 includes Hamiltonian Monte Carlo and NUTS, which is one of the most efficient methods
    - uses gradient information
    - Hamiltonian dynamic simulation reduces random walk
    - Demo <http://eleventh.org/blog/2017/11/28/build-a-better-markov-chain/>



# Example of uncertainty in modeling

## Posterior draws

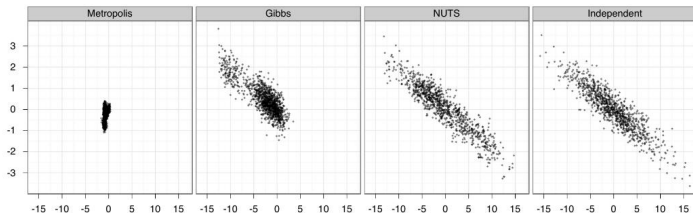




● Introduction

## Comparison of algorithms on **highly correlated** 250-dimensional Gaussian distribution

- Do **1,000,000** draws with both Random Walk Metropolis and Gibbs, thinning by 1000
- Do **1,000** draws using Stan's NUTS algorithm (no thinning)
- Do 1,000 independent draws (we can do this for multivariate normal)

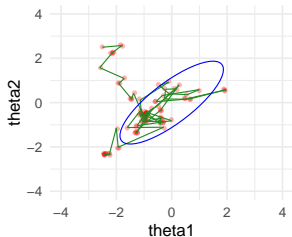


Source: Jonah Gabry



## Warm-up and convergence diagnostics

- Asymptotically chain spends the  $\alpha\%$  of time where  $\alpha\%$  posterior mass is
  - but in finite time the initial part of the chain may be non-representative and lower error of the estimate can be obtained by throwing it away



• Draws — Steps of the sampler — 90% HPD

- Warm-up = remove draws from the beginning of the chain
  - warm-up may include also phase for adapting algorithm parameters
- Convergence diagnostics
  - Do we get samples from the target distribution?





- Introduction

- Monte Carlo estimates still valid (central limit theorem holds)

$$E_{p(\theta|y)}[f(\theta)] \approx \frac{1}{S} \sum_{s=1}^S f(\theta^{(s)})$$

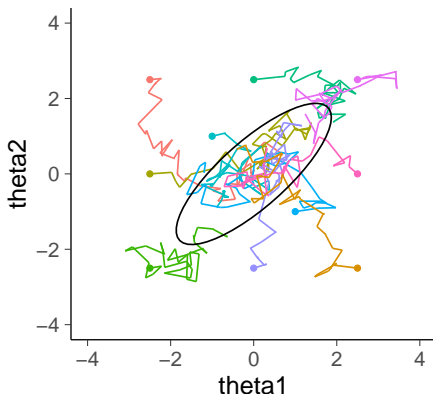
- Estimation of Monte Carlo error is more difficult
  - evaluation of *effective* sample size



## Several chains

- Use of several chains make convergence diagnostics easier
- Start chains from different starting points – preferably overdispersed

### No convergence

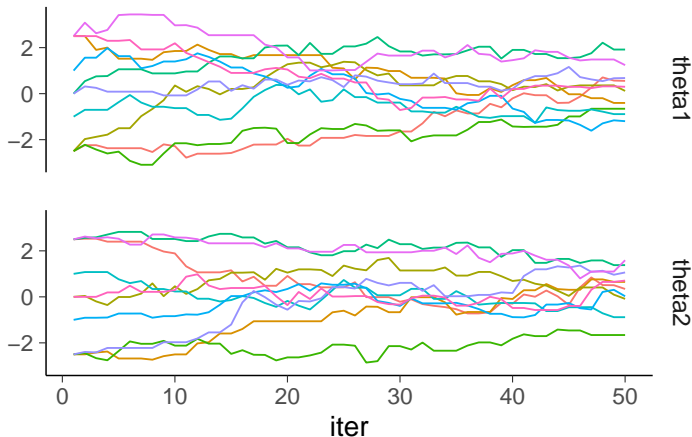


- Remove draws from the beginning of the chains and run chains long enough so that it is not possible to distinguish where each chain started and the chains are well mixed



## Several chains

Not converged



Visually converged





## $\hat{R}$ : comparison of within and between variances of the chains

---

- BDA3:  $\hat{R}$  aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains  
W = within chain variance estimate  
var\_hat\_plus = total variance estimate



- $M$  chains, each having  $N$  draws (with new  $\hat{R}$ -hat notation)
- Within chains variance  $W$

$$W = \frac{1}{M} \sum_{m=1}^M s_m^2, \text{ where } s_m^2 = \frac{1}{N-1} \sum_{n=1}^N (\theta_{nm} - \bar{\theta}_{.m})^2$$

- Between chains variance  $B$

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

$$\text{where } \bar{\theta}_{.m} = \frac{1}{N} \sum_{n=1}^N \theta_{nm}, \bar{\theta}_{..} = \frac{1}{M} \sum_{m=1}^M \bar{\theta}_{.m}$$

- $B/N$  is variance of the means of the chains
- Estimate total variance  $\text{var}(\theta|y)$  as a weighted mean of  $W$  and  $B$

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



- Estimate total variance  $\text{var}(\theta|y)$  as a weighted mean of  $W$  and  $B$

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

- this *overestimates* marginal posterior variance if the starting points are overdispersed
- Given finite  $N$ ,  $W$  *underestimates* marginal posterior variance
  - single chains have not yet visited all points in the distribution
  - when  $N \rightarrow \infty$ ,  $E(W) \rightarrow \text{var}(\theta|y)$
- As  $\widehat{\text{var}}^+(\theta|y)$  overestimates and  $W$  underestimates, compute

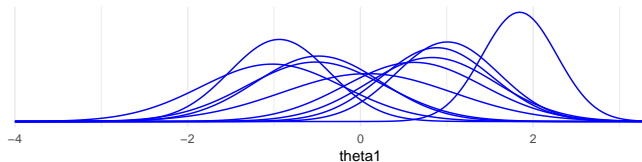
$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^+}{W}}$$



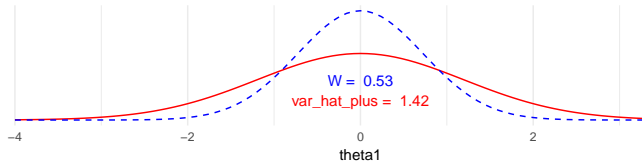
$\hat{R}$

- BDA3:  $\hat{R}$  aka *potential scale reduction factor* (PSRF)
- Compare means and variances of the chains  
W = within chain variance estimate  
var\_hat\_plus = total variance estimate

50 warmup, 50 post warmup iterations



Rhat = 1.64



500 warmup, 500 post warmup iterations





$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^+}{W}}$$

- Estimates how much the scale of  $\psi$  could reduce if  $N \rightarrow \infty$
- $\hat{R} \rightarrow 1$ , when  $N \rightarrow \infty$
- if  $\hat{R}$  is big (e.g.,  $R > 1.01$ ), keep sampling
- If  $\hat{R}$  close to 1, it is still possible that chains have not converged
  - if starting points were not overdispersed
  - distribution far from normal (especially if infinite variance)
  - just by chance when  $N$  is finite





- Introduction

- BDA3: split- $\hat{R}$
- Examines *mixing* and *stationarity* of chains
- To examine stationarity chains are split to two parts
  - after splitting, we have  $M$  chains, each having  $N$  draws
  - scalar draws  $\theta_{nm}$  ( $n = 1, \dots, N; m = 1, \dots, M$ )
  - compare means and variances of the split chains



# Rank normalized $\hat{R}$

---

- Original  $\hat{R}$  requires that the target distribution has finite mean and variance
- Rank normalization fixes this and is also more robust given finite but high variance
- Folding improves detecting scale differences between chains
- Paper proposes also local convergence diagnostics and practical MCSE estimates for quantiles
- Notation updated compared to BDA3

Vehtari, Gelman, Simpson, Carpenter, Bürkner (2020).  
Rank-normalization, folding, and localization: An improved  
R-hat for assessing convergence of MCMC. Bayesian Analysis,  
doi:10.1214/20-BA1221.

<https://projecteuclid.org/euclid.ba/1593828229>.



UPPSALA  
UNIVERSITET

# Time series analysis

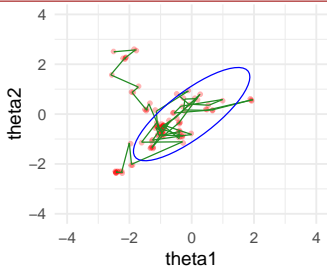
---

- Introduction

- Auto correlation function
  - describes the correlation given a certain lag
  - can be used to compare efficiency of MCMC algorithms and parameterizations

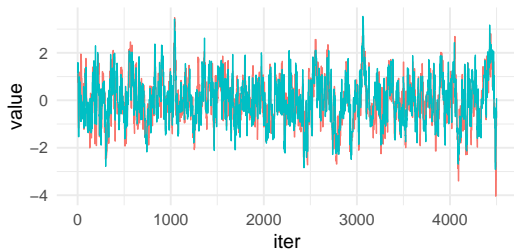


# Auto correlation



● Draws — Steps of the sampler — 90% HPI

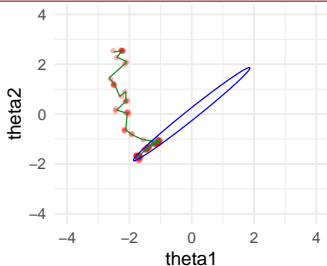
## Trends



—  $\theta_1$  —  $\theta_2$

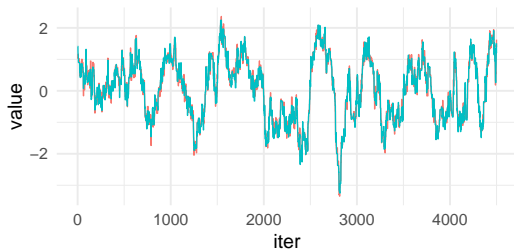


# Auto correlation



• Draws — Steps of the sampler — 90% HPI

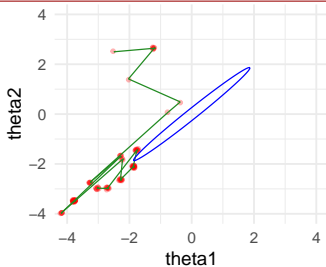
## Trends



—  $\theta_1$  —  $\theta_2$

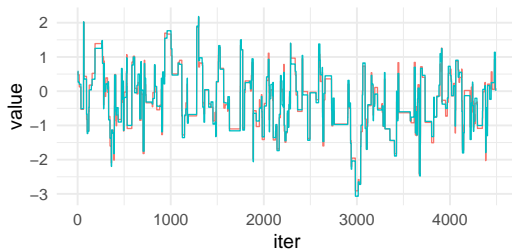


# Auto correlation



• Draws — Steps of the sampler — 90% HPI

## Trends



—  $\theta_1$  —  $\theta_2$



# Time series analysis

---

- Time series analysis can be used to estimate Monte Carlo error in case of MCMC
- For expectation  $\bar{\theta}$

$$\text{Var}[\bar{\theta}] = \frac{\sigma_{\theta}^2}{S_{E_{\max}}}$$

where  $S_{E_{\max}} = S/\tau$ , and  $\tau$  is sum of autocorrelations

- $\tau$  describes how many dependent draws correspond to one independent sample
- new R-hat paper  $S = NM$  (in BDA3  $N = nm$  and  $n_{E_{\max}} = N/\tau$ )
- BDA3 focuses on  $S_{E_{\max}}$  and not the Monte Carlo error directly  
new R-hat paper discusses more about MCSEs for different quantities



# Time series analysis

- Estimation of the autocorrelation using several chains

$$\hat{\rho}_n = 1 - \frac{W - \frac{1}{M} \sum_{m=1}^M \hat{\rho}_{n,m}}{2\widehat{\text{var}}^+}$$

where  $\hat{\rho}_{n,m}$  is autocorrelation at lag  $n$  for chain  $m$

- This combines  $\hat{R}$  and autocorrelation estimates
  - takes into account if the chains are not mixing (the chains have not converged)
- BDA3 has slightly different and less accurate equation. The above equation is used in Stan 2.18+
- Compared to a method which computes the autocorrelation from a single chain, the multi-chain estimate has smaller variance





- Estimation of  $\tau$

$$\tau = 1 + 2 \sum_{t=1}^{\infty} \hat{\rho}_t$$

where  $\hat{\rho}_t$  is empirical autocorrelation

- empirical autocorrelation function is noisy and thus estimate of  $\tau$  is noisy
- noise is larger for longer lags (less observations)
- less noisy estimate is obtained by truncating

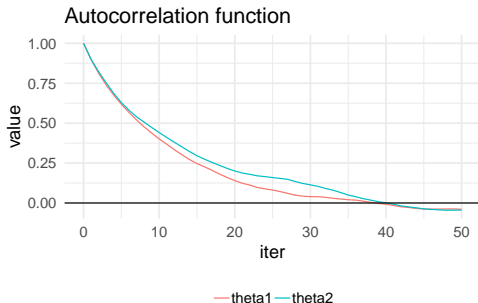
$$\hat{\tau} = 1 + 2 \sum_{t=1}^T \hat{\rho}_t$$

- As  $\tau$  is estimated from a finite number of draws, it's expectation is overoptimistic
  - if  $\hat{\tau} > MN/20$  then the estimate is unreliable



# Geyer's adaptive window estimator

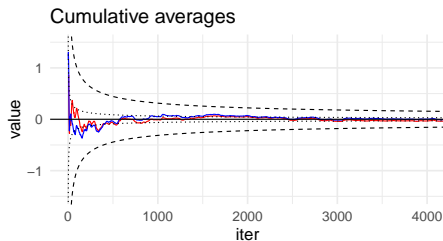
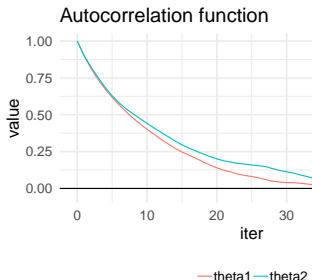
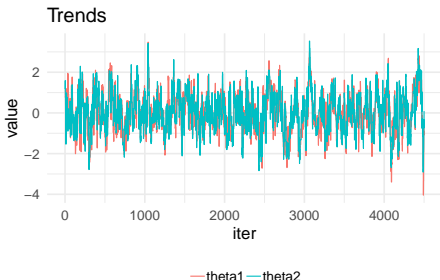
- Truncation can be decided adaptively
  - for stationary, irreducible, recurrent Markov chain
  - let  $\Gamma_m = \rho_{2m} + \rho_{2m+1}$ , which is sum of two consequent autocorrelations
  - $\Gamma_m$  is positive, decreasing and convex function of  $m$
- Initial positive sequence estimator (Geyer's IPSE)
  - Choose the largest  $m$  so, that all values of the sequence  $\hat{\Gamma}_1, \dots, \hat{\Gamma}_m$  are positive





# Effective sample size

$$\text{Effective sample size ESS} = S_{E_{\max}} \approx S/\hat{\tau}$$



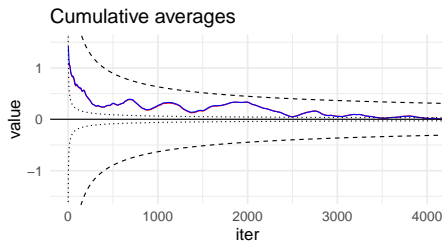
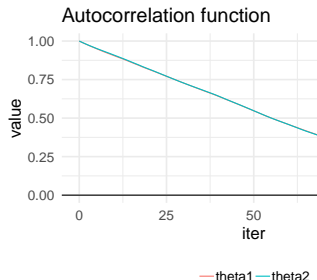
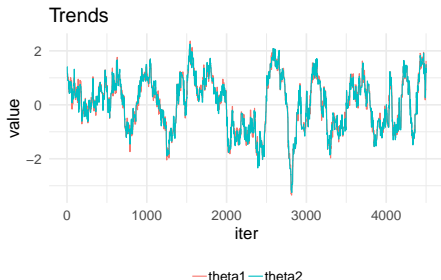
$$\hat{\tau} = 1 + 2 \sum_{t=1}^T \hat{\rho}_t$$

$$\approx 24$$



# Effective sample size

$$\text{Effective sample size ESS} = S_{E_{\max}} \approx S/\hat{\tau}$$



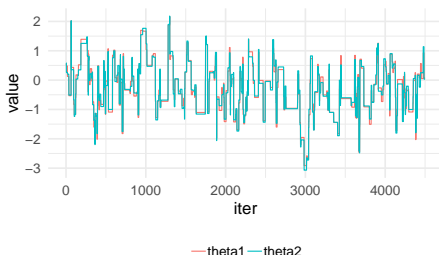
$$\hat{\tau} = 1 + 2 \sum_{t=1}^T \hat{\rho}_t$$
$$\approx 104$$



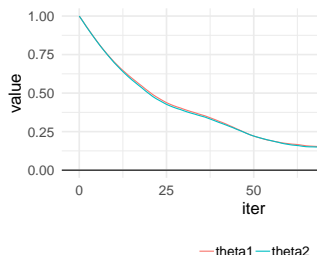
# Effective sample size

$$\text{Effective sample size ESS} = S_{E_{\max}} \approx S / \hat{\tau}$$

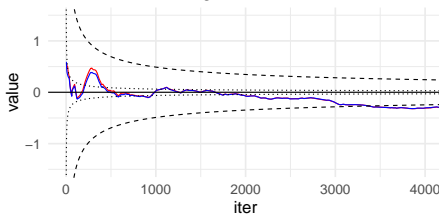
Trends



Autocorrelation function



Cumulative averages



- theta1 - theta2 -- 95% interval for MCMC error ..... 95% interval for indepen

$$\hat{\tau} = 1 + 2 \sum_{t=1}^T \hat{\rho}_t$$
$$\approx 63$$



- Introduction

- Nonlinear dependencies
  - optimal proposal depends on location
- Funnels
  - optimal proposal depends on location
- Multimodal
  - difficult to move from one mode to another
- Long-tailed with non-finite variance and mean
  - central limit theorem for expectations does not hold