BST 228: Applied Bayesian Analysis

Assessing MCMC Convergence

Convergence Remarks

- As we've seen already, we can use the detailed balance equation to build a Markov chain based on a known stationary distribution
- All Markov chains built are finite, so while theory may guarantee eventual convergence in many cases, that might not be sufficient for our needs
- As discussed in past lectures, we use diagnostics to assess how quickly the chain converges and whether it adequately represents a sample from the target (posterior) distribution

Metropolis-Hastings

- $f_p(y|x) =$ probability of proposing a move from x to y
- $R_{x \to y} =$ probability of accepting move from x to y

$$\frac{g(x)}{g(y)} = \frac{f_p(x|y)\ R_{y \to x}}{f_p(y|x)\ R_{x \to y}}$$

$$R_{x \to y} = \frac{g(y)}{g(x)} \frac{f_p(x|y)}{f_p(y|x)} R_{y \to x} \quad \text{or} \quad R_{y \to x} = \frac{g(x)}{g(y)} \frac{f_p(y|x)}{f_p(x|y)} R_{x \to y}$$

- By definition, neither $R_{x \to y}$ nor $R_{y \to x}$ can be greater than one
- To maximize the acceptance probabilities, set the larger of the two acceptance probabilities to 1, then the other is fixed:

$$R_{x \to y} = \min \left\{ 1, \frac{g(y)}{g(x)} \frac{f_p(x|y)}{f_p(y|x)} \right\}$$

Gibbs sampling

Sample situation

- Unable to sample from a posterior, $g(x, y)$
- But it is easy (easier) to sample x from $g(x|y)$ and y from $g(y|x)$

General case

- In Gibbs sampling, we move around in a state space in one (or more) variables at a time, sampling from conditional distributions

Gibbs sampling is a special case of Metropolis-Hastings

Gibbs example – initialize

```
#===> Data <===#
y <- 34 + 25*rt(20, 25)

#===> Summaries <===#
yn <- length(y)
yM <- mean(y)
yV <- var(y)

#===> Algorithm <===#
N  <- 1e4
mu <- rep(40, N)
s2 <- rep(9, N)

#===> Prior Info <===#
pM <- 40
w2 <- 1/9
pA <- 16.22
pB <- 172.2
```

Gibbs example – iterate

```
#===> Iterate <===#
for(i in 2:N){
  w1     <- yn / s2[i-1]
  normP1 <- (w1*yM + w2*pM)/(w1+w2)
  normP2 <- 1/(w1+w2)
  mu[i]  <- rnorm(1, normP1, sqrt(normP2))

  gamP1 <- pA + yn/2
  temp  <- yV*(yn-1) + yn*(yM - mu[i])^2
  gamP2 <- pB + temp/2
  s2[i] <- 1/rgamma(1, gamP1, gamP2)
}
mu <- mu[-(1:100)]
s2 <- s2[-(1:100)]
```

Specially note that `s2[i-1]` is used to obtain `mu[i]`, but `mu[i]` is used to obtain `s2[i]`

Metropolis-Hastings and Gibbs

Convergence considerations for M-H (and Gibbs)

- What decisions do we have to make?
- What problems do we look for?
- How do we assess sampling performance?

Available options for MH and Gibbs

There are several options we can use when creating an MCMC algorithm

- proposal function
- variable grouping
- variable order in chain
- starting values
- length of the chain

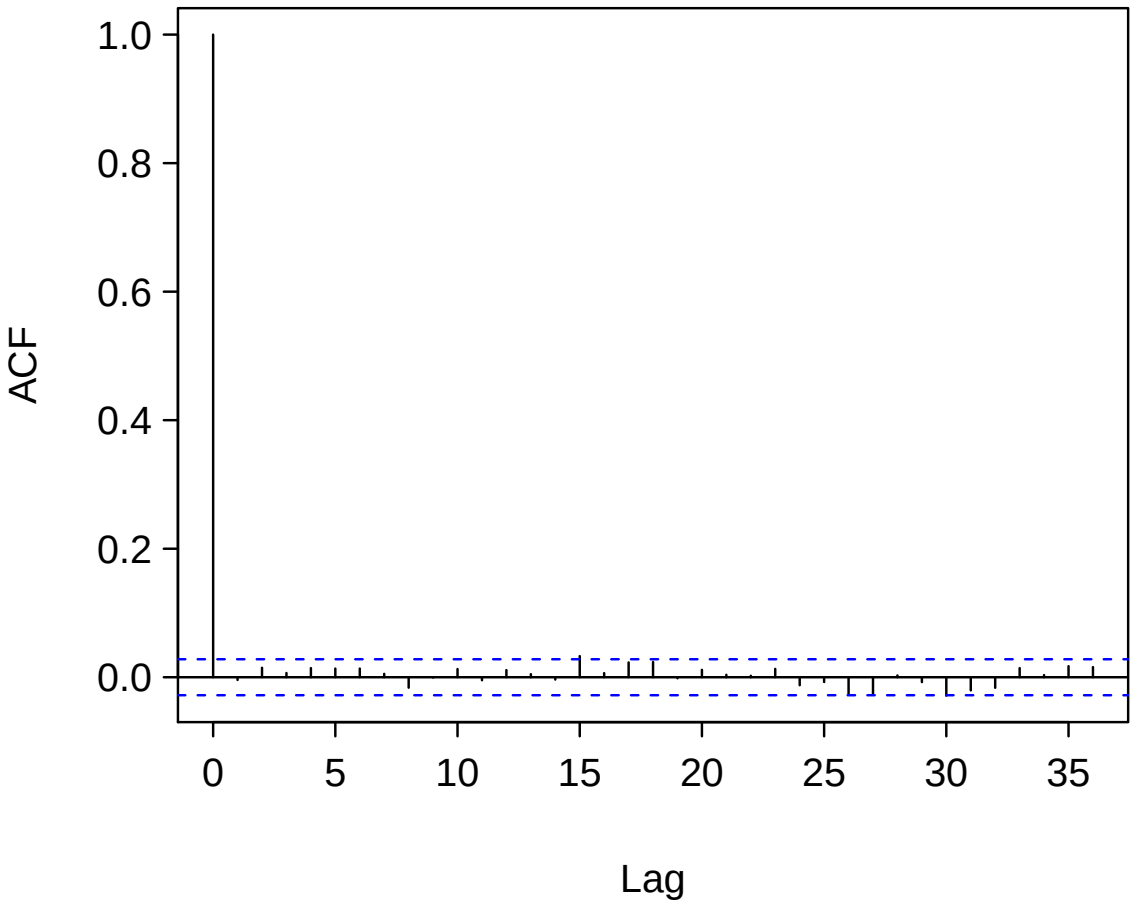
We will think back to these options as we examine convergence assessment tools

How we define "best"

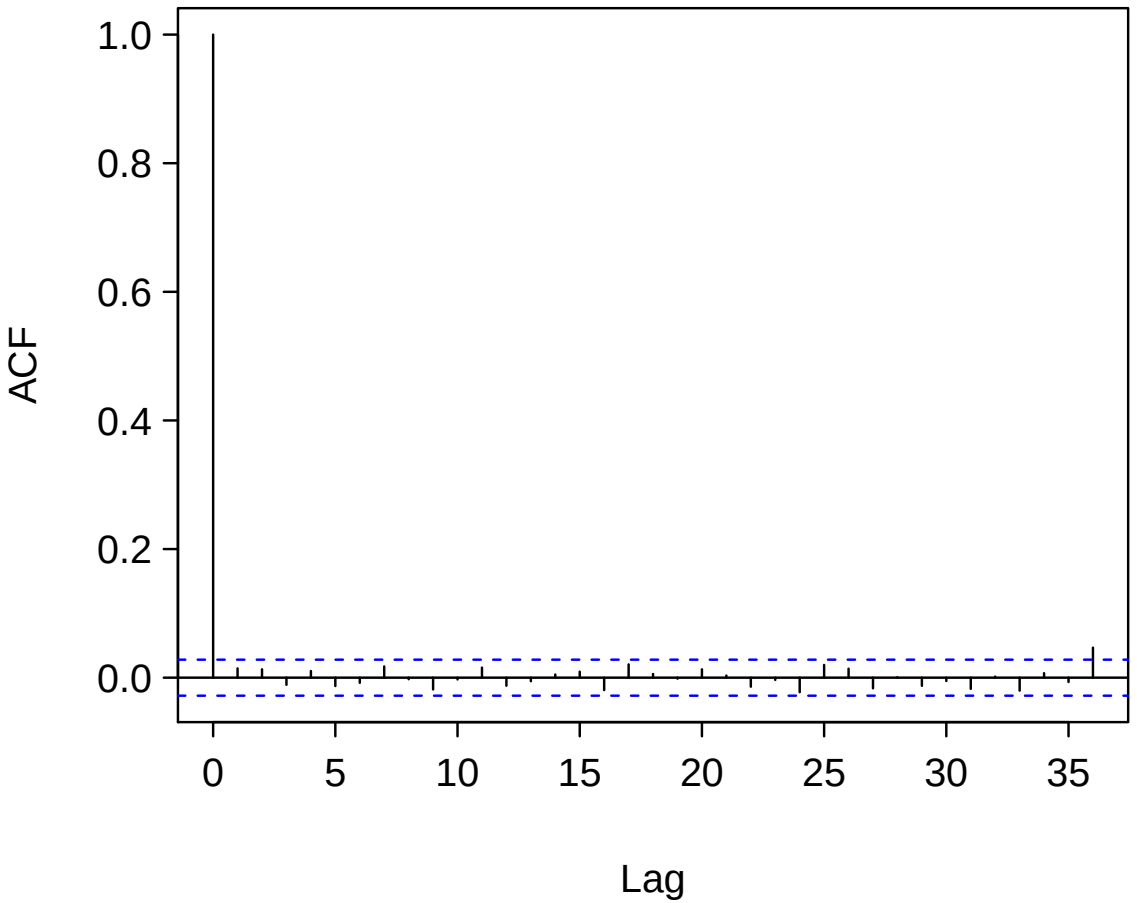
Properties of the Markov chain

- The observations $x_1, x_2, ...$ are *not* independent
- Observations close in the sequence are generally correlated
- A good MCMC sample tends to keep this correlation low (there are also other criteria)
- We can measure these correlations to provide a helpful guide to the mixing of the the Markov chain

Gibbs example – ACF of `mu`



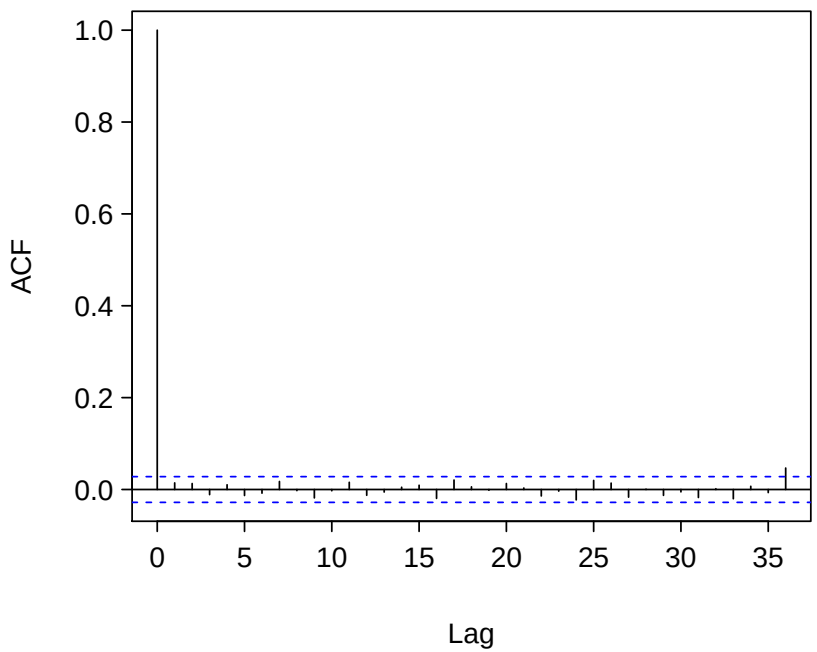
Gibbs example – ACF of s2



ACF uncertainty

There are error bounds on the ACF

- These tend to rely mostly on the length of the chain
- If your chains are even modestly long, then the error in the ACF estimate will have little impact on convergence assessment



Thinning

- High ACF may indicate poor choices in constructing the algorithm, or it might just indicate that there isn't much you can do (for MH, check acceptance rate)
- When ACF remains large for large lags, consider thinning the chain
- Thinning is the process of only saving a fraction of the chain, typically every k^{th} sample where k is chosen by the user and depends on the setting
- For example, if the ACF goes to 0.8 after 50 lags, consider saving only 1 out of every 50 samples
- Saving an entire chain with high autocorrelation will be a waste of hard drive space

Effective sample size

This formula will vary slightly depending on the source:

$$N_{ESS} = N \left(1 + 2 \sum_{i=1}^{\infty} c_i \right)^{-1}$$

- Consider the sample mean for large N:

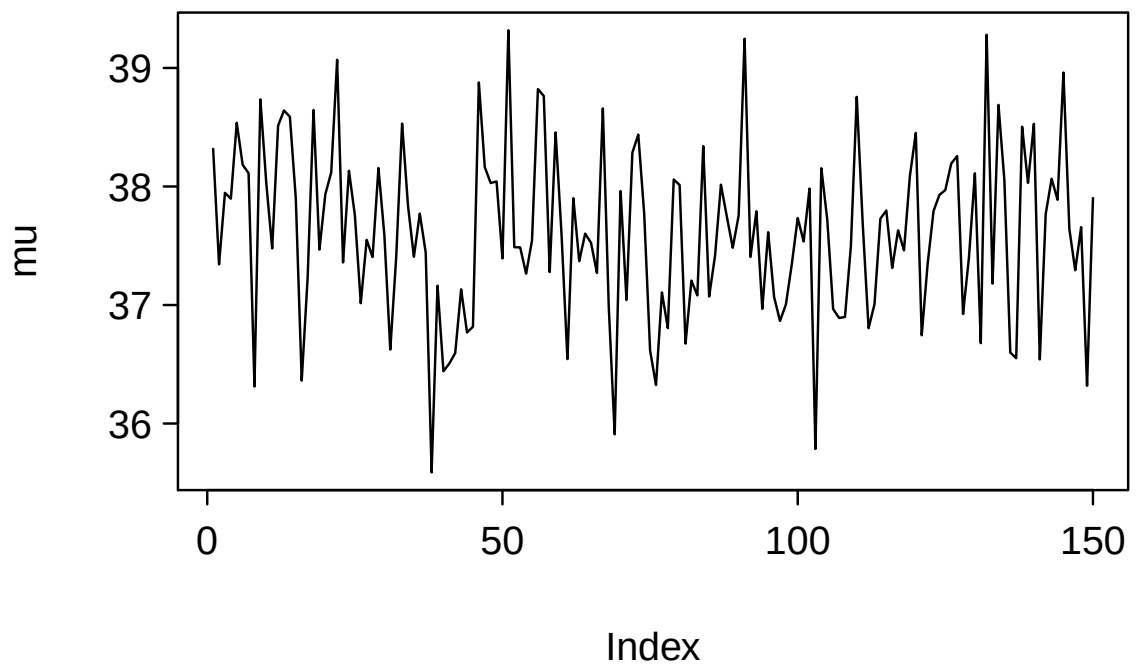
$$\bar{x} = \frac{x_1 + \cdots + x_N}{N}$$

- Prove that if the x_i are correlated where c_i represents the autocorrelation with lag i, that the variance of $\bar{x}$ is about the same as if we had a sample of N_{ESS} independent x_i
- To make this more reasonable, let's assume $c_k = 0$ for all $k > K$ where $K << N$

Trace plots

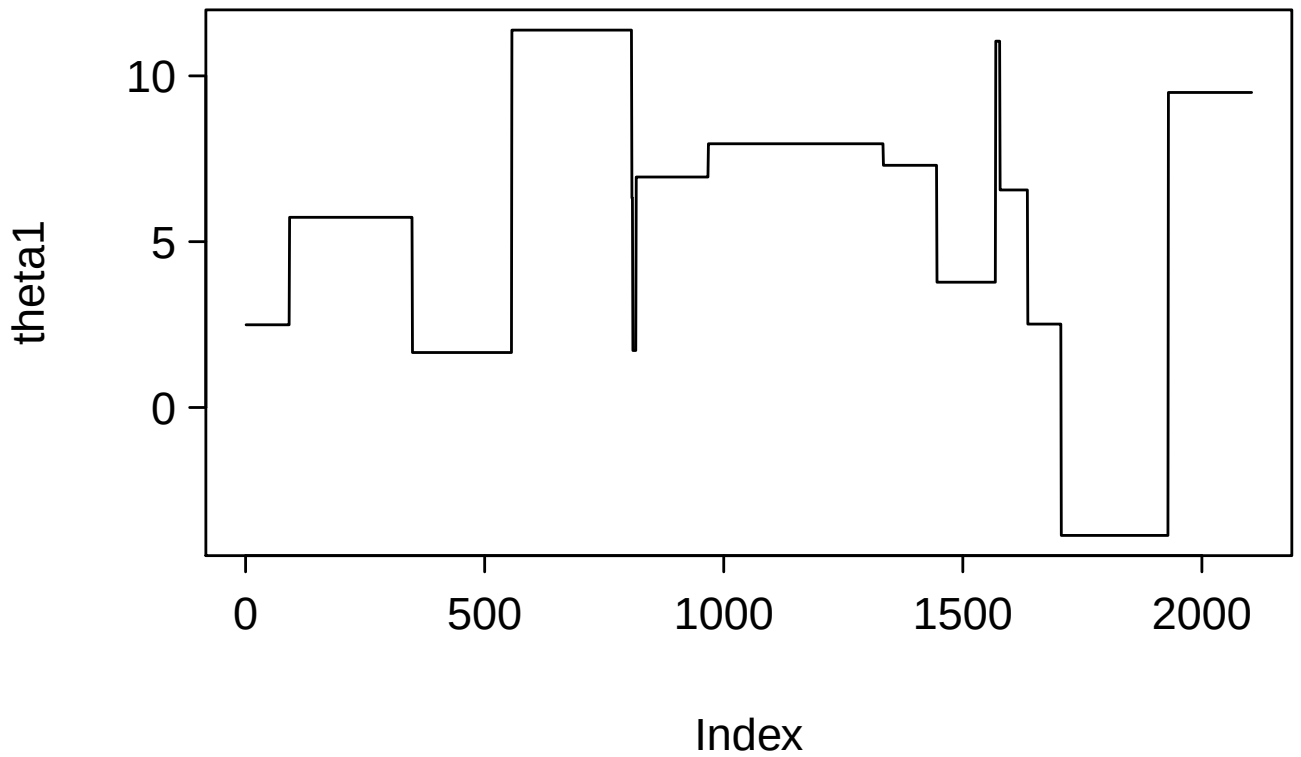
Plot the path of a chain

- Plot the value of each parameter over time
- There should be convergence to a region, but not to a single value



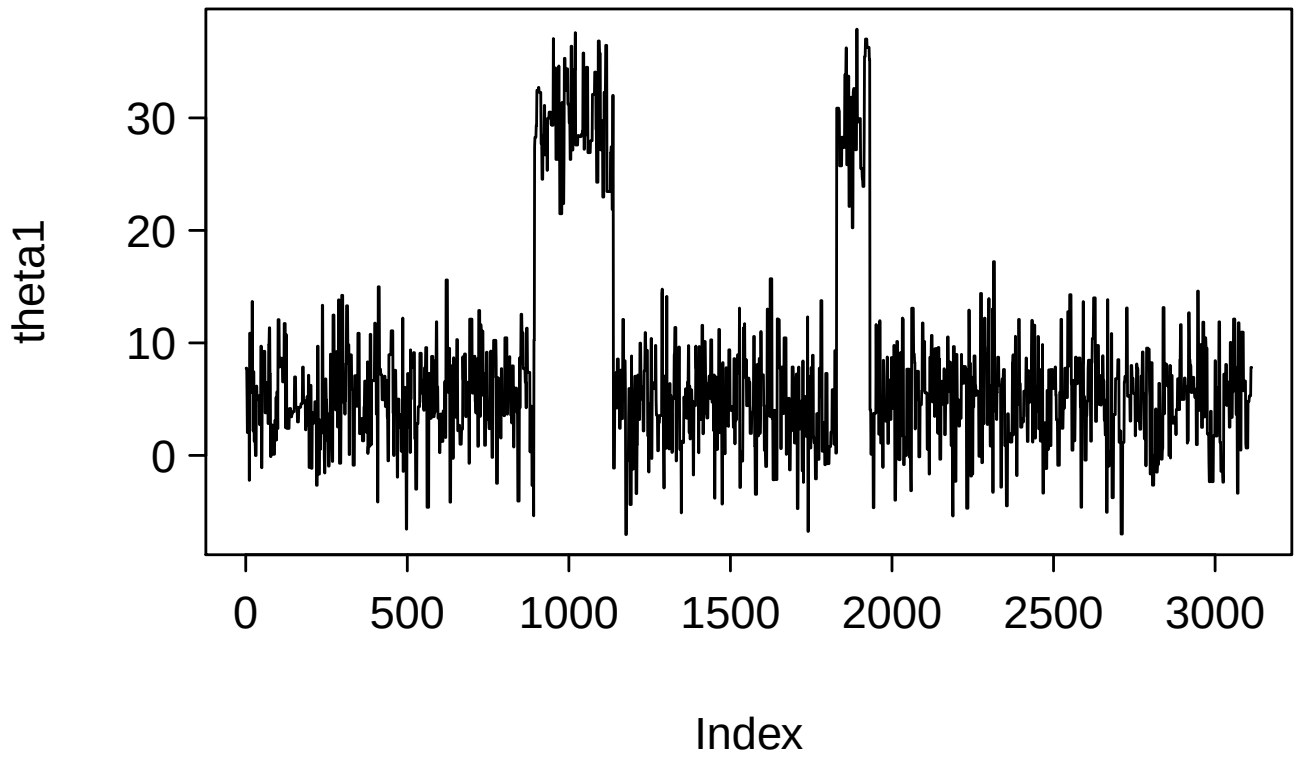
Trace plots

Why would a chain pause on values for long periods of iterations?



Trace plots

What would the plot below indicate?



Running mean – computation

Use `cumsum` to get the running total, then divide by the number of included values

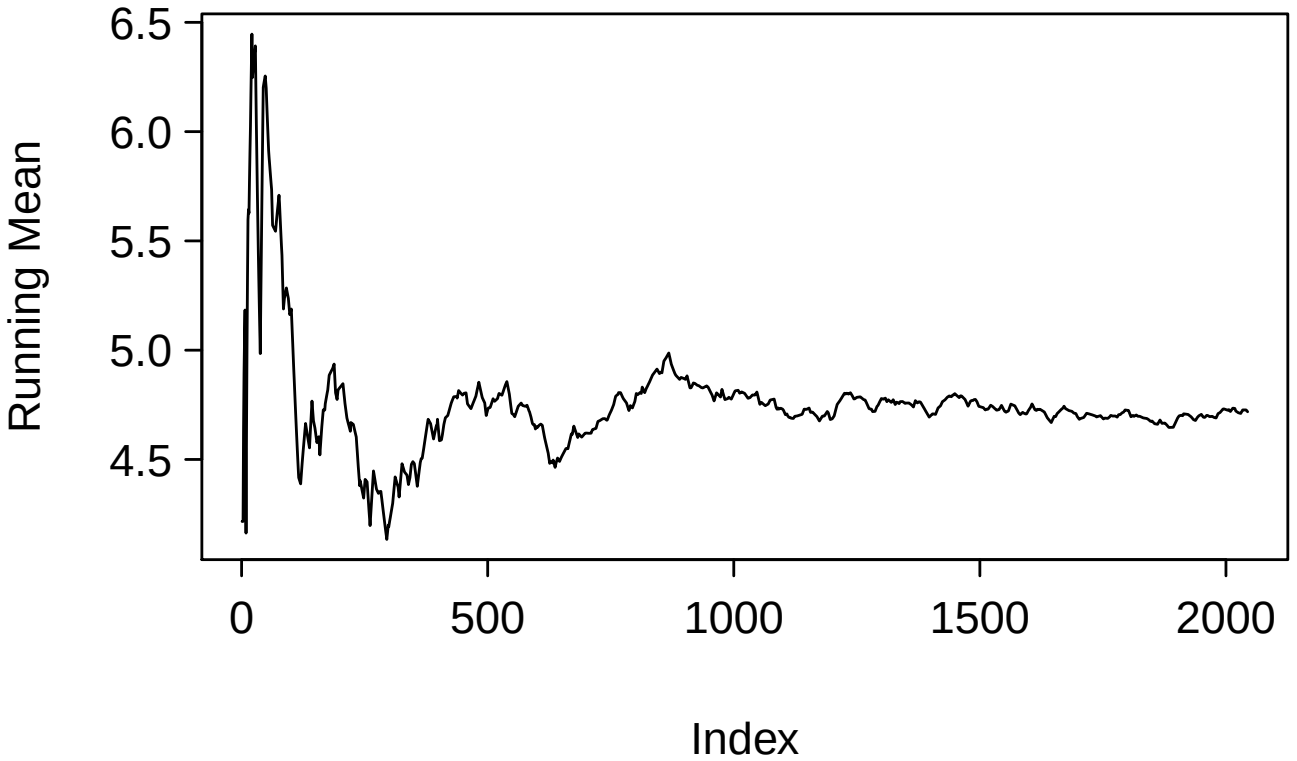
```
s     <- cumsum(theta1)
n     <- 1:length(theta1)
rMeam <- s/n
plot(rMean)
```

Running mean

From the chain on the previous slide... how does this look?

Running mean

When convergence of the running mean looks more stable



Burn in

The starting value is important

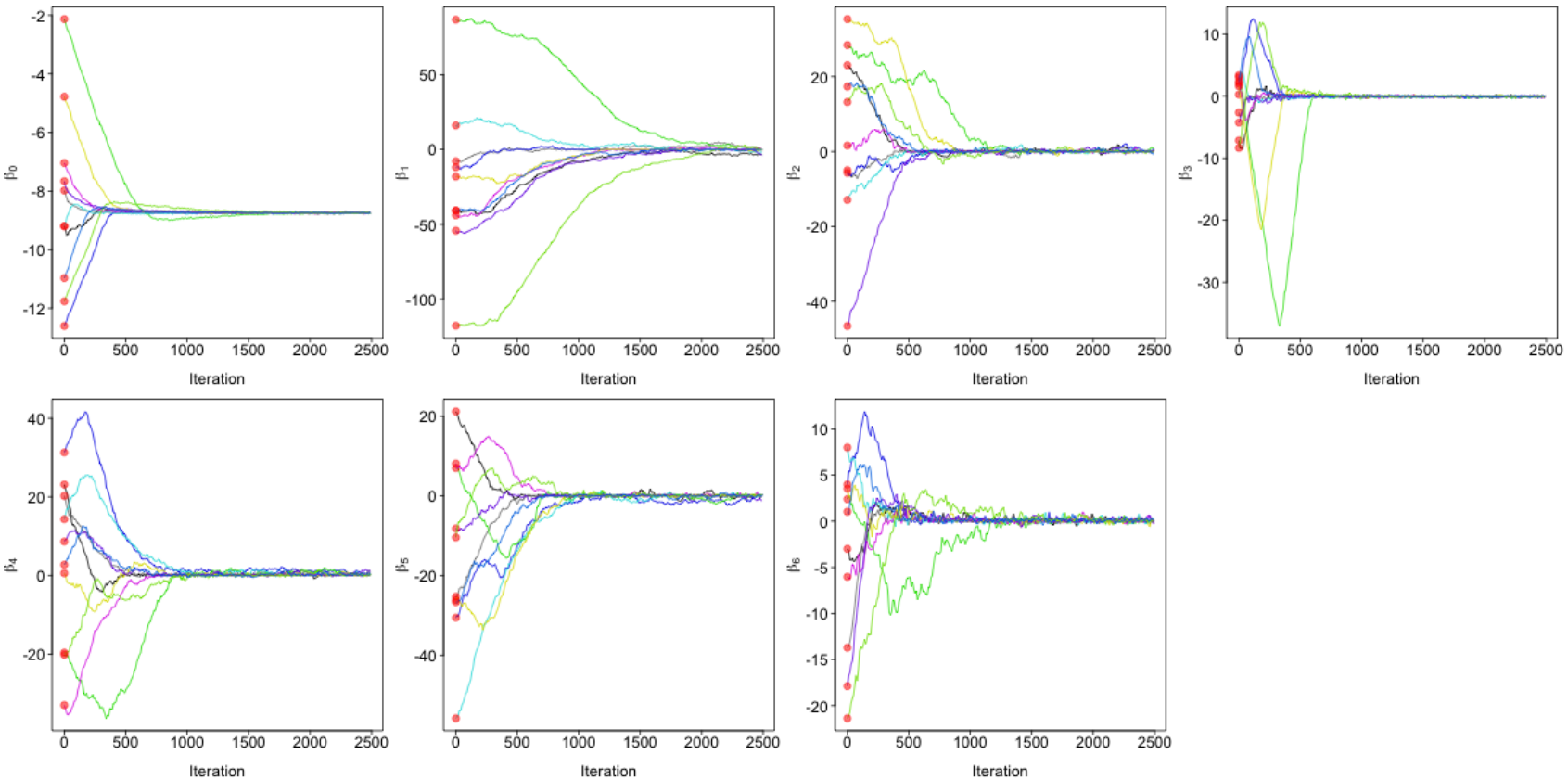
- As the chain gets large, it will forget its starting location
- We want to work with parts of the chain that forget this non-random starting location

Removing a burn-in

- We use the ACF and trace plots to decide when we are confident the chain has forgotten its starting location
- Drop all values before this cutoff as a burnin

Starting values

Try many starting values, and put all of the chains on a single plot



Try starting values that are grossly different than values in your longer chains

Gelman and Rubin's scale reduction factor

Running m chains of length n (after burn-in)

- These chains $\{\{x_{j,t}\}_{t=1}^{n}\}_{j=1}^{m}$ offer m inferences
- Does inference differ significantly in each chain?

The topic of interest

- Parameter of interest: has mean μ with variance σ^2 in the posterior
- Use the average of all x_{jt} for estimate $\hat{\mu}$
- Differ much from the individual chain estimates, $\bar{x}_{j.}$?

Constructing the variance estimate

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{x}_{j.} - \bar{x}_{..})^2 \qquad W = \frac{1}{m(n-1)}\sum_{j=1}^{m}\sum_{t=1}^{n}(x_{jt} - \bar{x}_{j.})^2$$

$$\hat{\sigma}_{+}^{2} = \frac{n-1}{n}W + \frac{1}{n}B \qquad \hat{R} = \sqrt{\frac{\hat{\sigma}_{+}^{2}}{W}}$$

Gelman and Rubin's scale reduction factor

Potential scale reduction factor ($\hat{R}$)

≈ 1 Each of the m chains appear to follows the same distribution, so pool the chains

$>\approx 1.2$ Increase the length of the chains!

Can also use for other statistics than the mean

References

- Gelman & Rubin (1992): *Inference from Iterative Simulation Using Multiple Sequences*
- Brooks & Gelman (1998): *General Methods for Monitoring Convergence of Iterative Simulations*