Markov Chain Monte Carlo: A Practical Introduction

Posterior Distribution

- Posterior distribution for $\boldsymbol{\theta}$ given $\mathbf{x}$ is

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{x})} \propto p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

- $\boldsymbol{\theta}$ may a be a collection
- Example: Inference about a mean and variance of a $N(\mu, \sigma^2)$
- $\boldsymbol{\theta} = (\mu, \sigma^2)$

$$p(\mu, \sigma^2 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mu, \sigma^2) p(\mu, \sigma^2)}{p(\mathbf{x})} \propto p(\mathbf{x} \mid \mu, \sigma^2) p(\mu, \sigma^2)$$

Estimation in Bayesian Modeling

- Our "answer" is a posterior distribution
  - All parameters treated as random, not fixed
  - *"The Bayesian ~~estimate~~ is…"* ~~is at best ambiguous~~

- Contrasts with frequentist approaches to inference, estimation
  - Parameters are fixed, so estimation comes to finding the single best value
  - "Best" here in terms of a criterion (ML, LS, etc.)

- Peak of a mountain vs. mapping the entire terrain of peaks, valleys, and plateaus (of a landscape)

Strategies for Obtaining Posterior Distributions

Evaluating Posterior Distributions: Do The Math

1. ***Analytically***. Do the math. This requires the product of the prior and the likelihood taking on a recognizable form and the evaluation of the denominator. Illuminated via conjugacy: the prior distribution takes on a particular form such that, in combination with the form of the likelihood yields a posterior that is of the same form as the prior

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{x})}$$

Examples:

$x \sim \text{Binomial}(\theta, J)$, $\theta \sim \text{Beta} \rightarrow \theta|x \sim \text{Beta}$

$x \sim N(\mu, \sigma^2)$ with σ^2 known, $\mu \sim N \rightarrow \mu|x \sim N$

Evaluating Posterior Distributions: Approximate The Math

2. ***Approximate the analytical solution***. Ok, can't do the math. So approximate it some way.

   Examples:

   Normal approximations to marginal, posterior densities

   Taylor series analytic approximations

Evaluating Posterior Distributions: Avoid the Problem

3. ***Evaluate features of the distribution.*** Don't try to evaluate the posterior distribution, just try to find its features/summaries, such as the posterior mean, or standard deviation. Also the posterior mode, which is akin to ML: Find the most likely value, but in the posterior, not the likelihood.

   May be straightforward in conjugate situations, obtained numerically under conjugacy when mode does not have closed form, obtained numerically in general cases if derivatives are amenable

   Example: Posterior modal estimation of IRT models in BILOG

Evaluating Posterior Distributions: Simulation

4. ***Simulation-based estimation***. Construct a sampling algorithm to ***simulate*** or ***draw from*** the posterior. Collect many such draws, which serve to empirically approximate the posterior distribution, and can be used to empirical approximate summary statistics.

Monte Carlo Principle:

> Anything we want to know about a random variable θ can be learned by sampling many times from $f(\theta)$, the density of θ.
>
> -- Jackman (2009, p. 133)

Evaluating Posterior Distributions: Simulation

4. ***Simulation-based estimation***. Construct a sampling algorithm to ***simulate*** or ***draw from*** the posterior.

   If you know the form of the posterior (*N*, *Gamma*, *t*, χ^2, etc.) simulate or draw from that distribution

   Many different algorithms

   inverse-CDF, importance sampling, accept-reject sampling, adaptive-rejection sampling

Evaluating Posterior Distributions: Simulation With Markov Chains

5. ***Simulation-based estimation***. Construct a sampling algorithm to simulate or draw from the posterior***…without requiring we know the form of the distribution***. Collect many such draws, which serve to empirically approximate the posterior distribution, and can be used to empirical approximate summary statistics.

   Monte Carlo Principle

   Do this via Markov chain Monte Carlo estimation

   Example: Stan, JAGS, BUGS, and similar programs

Markov Chain Monte Carlo

What's In a Name?

Markov chain ***Monte Carlo***

- Construct a sampling algorithm to ***simulate*** or ***draw from*** the posterior.
- Collect many such draws, which serve to empirically approximate the posterior distribution, and can be used to empirical approximate summary statistics.

Monte Carlo Principle:

> Anything we want to know about a random variable θ can be learned by sampling many times from $f(\theta)$, the density of θ.
>
> -- Jackman (2009, p. 133)

What's In a Name?

Markov ***chain*** Monte Carlo

- Values really generated as a sequence or chain
- t denotes the step in the chain
- $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(t)}, \ldots, \theta^{(T)}$
- Also thought of as a time indicator

Markov chain Monte Carlo

- Follows the Markov property…

The Markov Property

- Current state depends on previous position
  - Examples: weather, checkers, baseball counts & scoring

- Next state conditionally independent of past, given the present
  - Akin to a full mediation model

- $p(\theta^{(t+1)} \mid \theta^{(t)}, \theta^{(t-1)}, \theta^{(t-2)}, \ldots) = p(\theta^{(t+1)} \mid \theta^{(t)})$

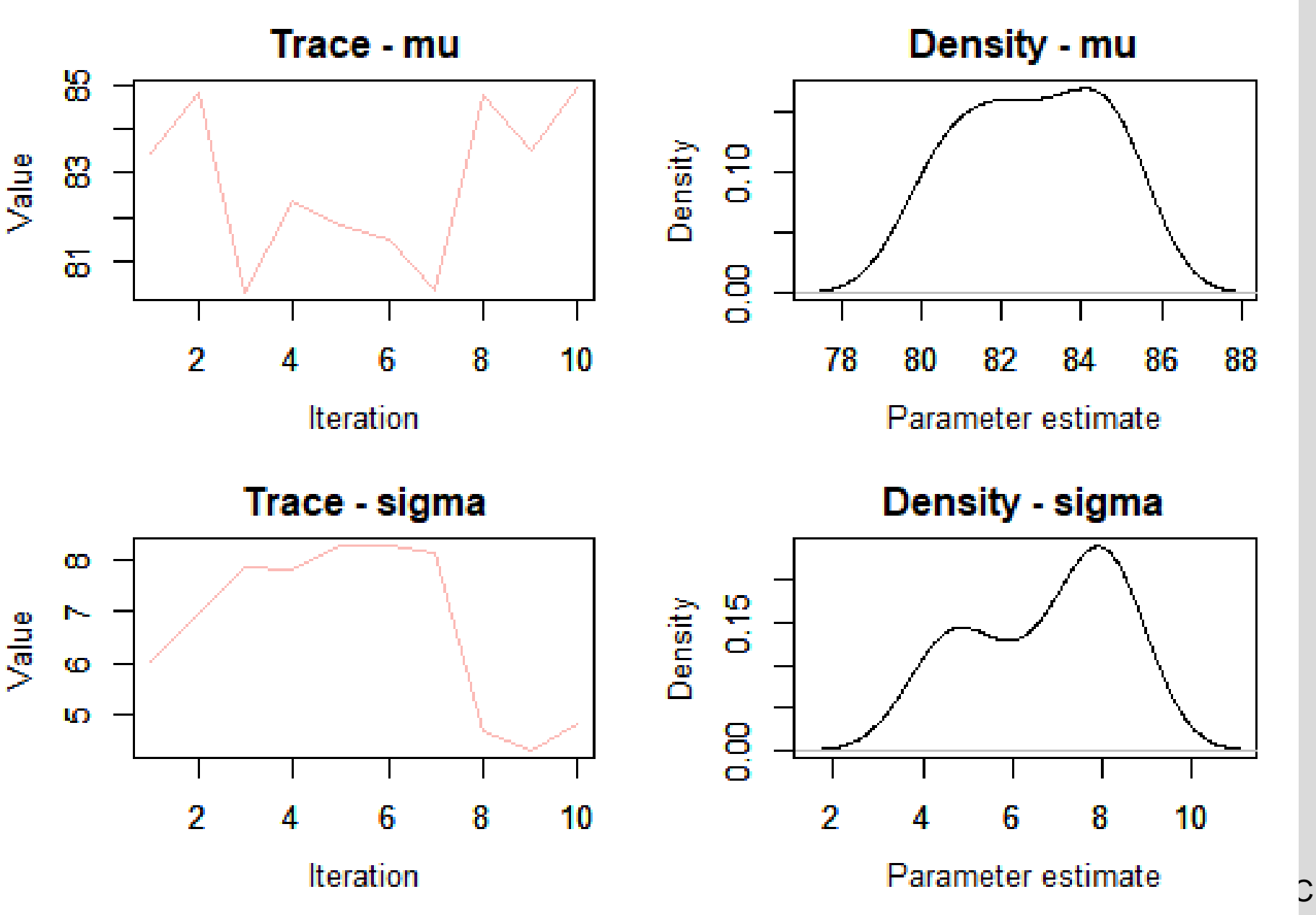
Running Stan via R

In R code in *Normal model (mu and sigma.squared unknown) in Stan via rstanarm.R*

```
# Choose features of MCMC --------
#   the number of chains
#   the number of iterations to warmup
#   the total number of iterations
n.chains = 1
n.warmup = 1000
n.iters.per.chain.after.warmup = 10
n.iters.total.per.chain =
n.iters.per.chain.after.war
```

Requesting 10 iterations: number of simulations from the distribution

Visualizing the Chain: Trace Plot



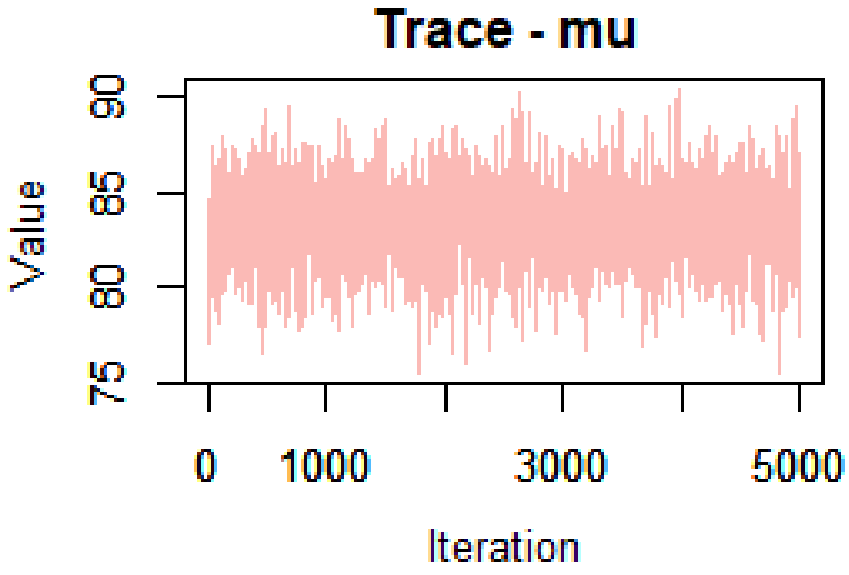
Running Stan via R

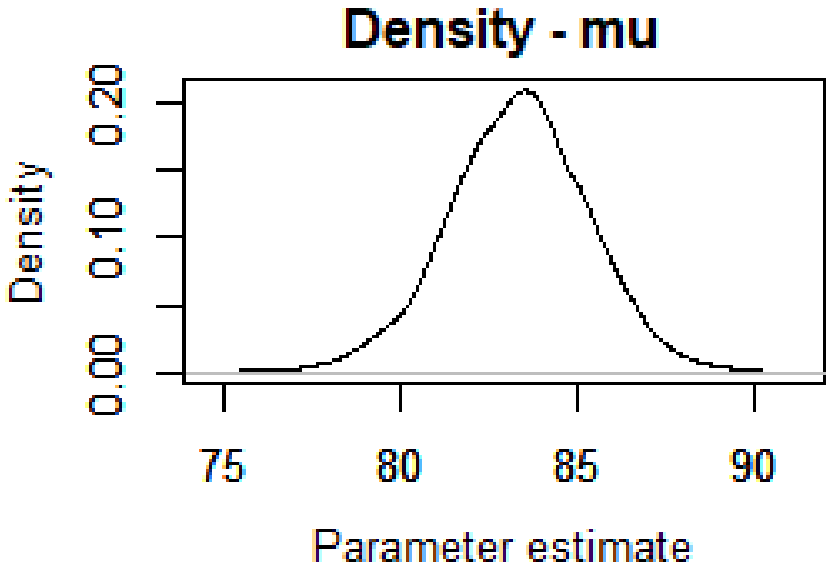
In R code in *Normal model (mu and sigma.squared unknown) in Stan via rstanarm.R*

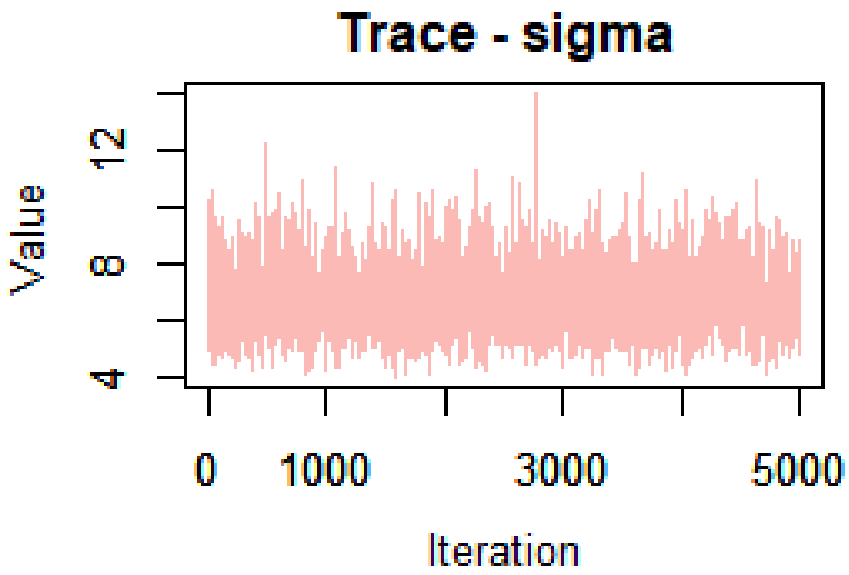
```
# Choose features of MCMC --------
#   the number of chains
#   the number of iterations to warmup
#   the total number of iterations
n.chains = 1
n.warmup = 1000
n.iters.per.chain.after.warmup = 5000
n.iters.total.per.chain =
n.iters.per.chain.after.war
```

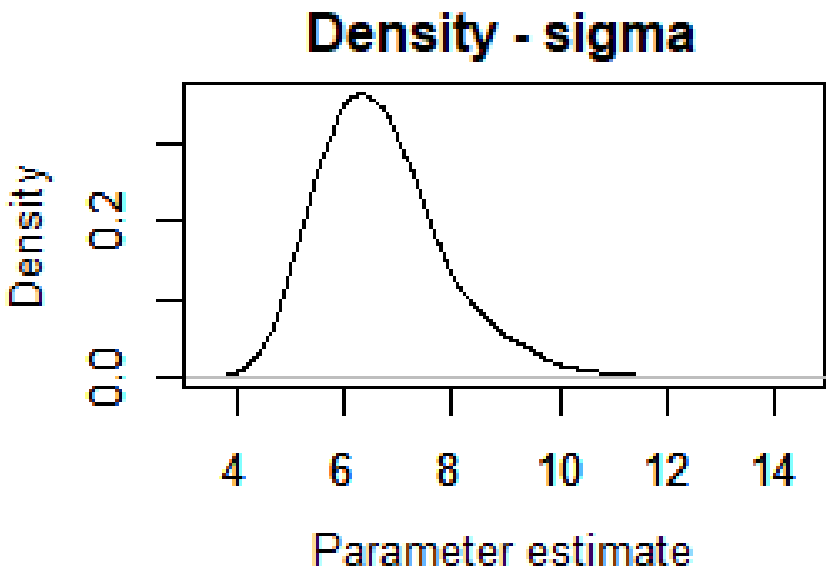
Requesting 5000 iterations: number of simulations from the distribution

Visualizing the Chain: Trace Plot









Markov Chain Monte Carlo

- Markov chains are ***sequences of numbers*** that have the Markov property
  - Draws in cycle $t+1$ depend on values from cycle t, but given those not on previous cycles (Markov property)
- Under certain assumptions Markov chains reach ***stationarity***
- The collection of values converges to a distribution, referred to as a stationary distribution
  - Memoryless: It will "forget" where it starts
  - Start anywhere, will reach stationarity if regularity conditions hold
  - For Bayes, set it up so that this is the posterior distribution
- Upon convergence, samples from the chain approximate the stationary (posterior) distribution

Sidebar: Illustrating Convergence to a Distribution

Distribution of Weather (Simplified)

- Let $\theta^{(t)}$ be a binary variable corresponding to whether it is *sunny* or *rainy* on day t
- The probability that it rains on a given day depends on the previous day (day $t - 1$)

Weather on a day	p(Sunny next day)	p(Rainy next day)
Sunny	.9	.1
Rainy	.7	.3

Distribution of Weather (Simplified)

- Suppose it is sunny on day 0
- What is the probability distribution for day 1?

$$p(\theta^{(1)}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix} = \begin{bmatrix} .9 & .1 \end{bmatrix}$$

Vector of (sunny, rainy)

Transition probability matrix
(1st row sunny, 2nd row rainy)

Distribution of Weather (Simplified)

$$p(\theta^{(2)}) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}^2 = \begin{bmatrix}.88 & .12\end{bmatrix}$$

$$p(\theta^{(3)}) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}^3 = \begin{bmatrix}.876 & .124\end{bmatrix}$$

$$p(\theta^{(4)}) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}^4 = \begin{bmatrix}.8752 & .1248\end{bmatrix}$$

$$p(\theta^{(5)}) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}^5 = \begin{bmatrix}.87504 & .12496\end{bmatrix}$$

$$p(\theta^{(100)}) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}.9 & .1\\ .7 & .3\end{bmatrix}^{100} = \begin{bmatrix}.875 & .125\end{bmatrix}$$

Distribution of Weather (Simplified)

- Weather as a Markov property
  - Probability of weather today governed by yesterday
- Start with a sunny day, $p(\theta)$ in the limit = [.875, .125]

Distribution of Weather (Simplified)

- Suppose it is rainy on day 0
- What is the probability distribution for day 1?

$$p(\theta^{(1)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix} = \begin{bmatrix} .7 & .3 \end{bmatrix}$$

Vector of (sunny, rainy)

Transition probability matrix
(1st row sunny, 2nd row rainy)

Distribution of Weather (Simplified)

$$p(\theta^{(2)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix}^2 = \begin{bmatrix} .84 & .16 \end{bmatrix}$$

$$p(\theta^{(3)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix}^3 = \begin{bmatrix} .868 & .132 \end{bmatrix}$$

$$p(\theta^{(4)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix}^4 = \begin{bmatrix} .8736 & .1264 \end{bmatrix}$$

$$p(\theta^{(5)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix}^5 = \begin{bmatrix} .87472 & .12528 \end{bmatrix}$$

$$p(\theta^{(100)}) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix}^{100} = \begin{bmatrix} .875 & .125 \end{bmatrix}$$

Distribution of Weather (Simplified)

- Weather as a Markov property
  - Probability of weather today governed by yesterday
- Start with a sunny day, $p(\theta)$ in the limit = [.875, .125]
- Start with a rainy day, $p(\theta)$ in the limit = [.875, .125]
- Doesn't matter where you start!
- Will always arrive at the ***stationary distribution***
- Note that we are discussing arriving at a ***distribution***, not a point
  - Not saying that it in the limit it will be sunny, but that in the limit it will be sunny with probability .875
  - $p(\text{sunny}) = .875$, $p(\text{rainy}) = .125$

Our Responsibilities

Our Responsibilities

Markov chain Monte Carlo is a highly technical and usually automated procedure. You might write your own MCMC code, for the sake of learning. But it is very easy to introduce subtle biases. A package like Stan, in contrast, is continuously tested against expected output. Most people who use Stan don't really understand what it is doing, under the hood. That's okay. Science requires division of labor, and if every one of us had to write our own Markov chains from scratch, a lot less research would get done in the aggregate.

-- McElreath (2020, p. 287)

Our Responsibilities

Markov chain Monte Carlo is a highly technical and usually automated procedure. You might write your own MCMC code, for the sake of learning. But it is very easy to introduce subtle biases. A package like Stan, in contrast, is continuously tested against expected output. Most people who use Stan don't really understand what it is doing, under the hood. That's okay. Science requires division of labor, and if every one of us had to write our own Markov chains from scratch, a lot less research would get done in the aggregate.

But as with many technical and powerful procedures, it's natural to feel uneasy about MCMC and maybe even a little superstitious. Something magical is happening inside the computer, and unless we make the right sacrifices and say the right words, an ancient evil might awake. So we do need to understand enough to know when the evil stirs.

-- McElreath (2020, p. 287)

Assessing Convergence

Assessing Convergence

- With MCMC, convergence to a ***distribution***, not a point

- ML:
  - Convergence is when we've reached the highest point in the likelihood,
  - The highest peak of the mountain

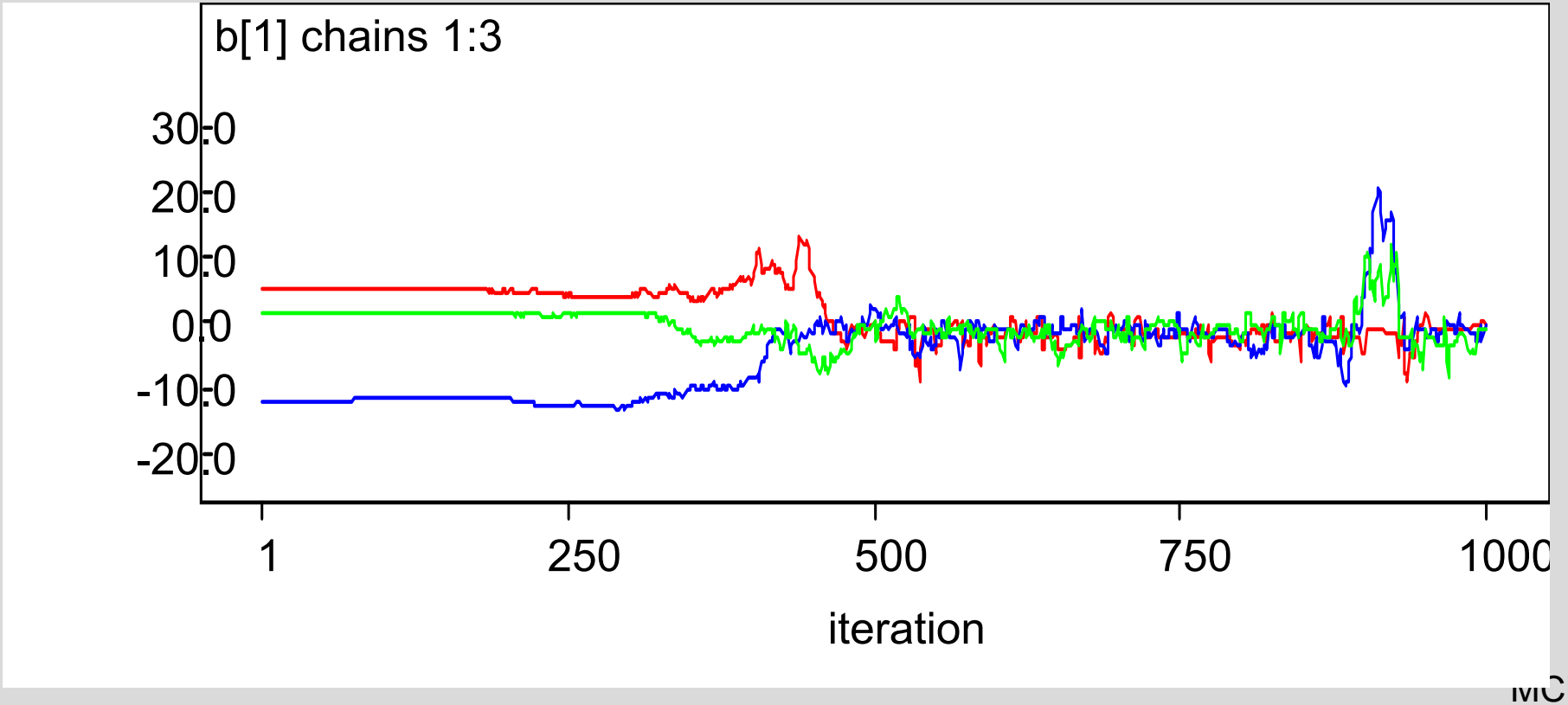
- MCMC:
  - Convergence when we're sampling values from the correct distribution,
  - We are mapping the entire terrain accurately

Assessing Convergence

- A properly constructed Markov chain is guaranteed to converge to the stationary (posterior) distribution…eventually
- Upon convergence, it will sample over the full support of the stationary (posterior) distribution…over an ∞ number of draws
- In a finite chain, no guarantee that the chain has converged or is sampling through the full support of the stationary (posterior) distribution
- Many ways to diagnose convergence
- Whole software packages dedicated to just assessing convergence of chains (e.g., R packages ‘coda’ and ‘boa’)

Graphical Check on Multiple Chains

- Run ***multiple*** chains from dispersed starting points
- Suggest convergence when the chains come together
- If they all go to the same place, it's probably the stationary distribution

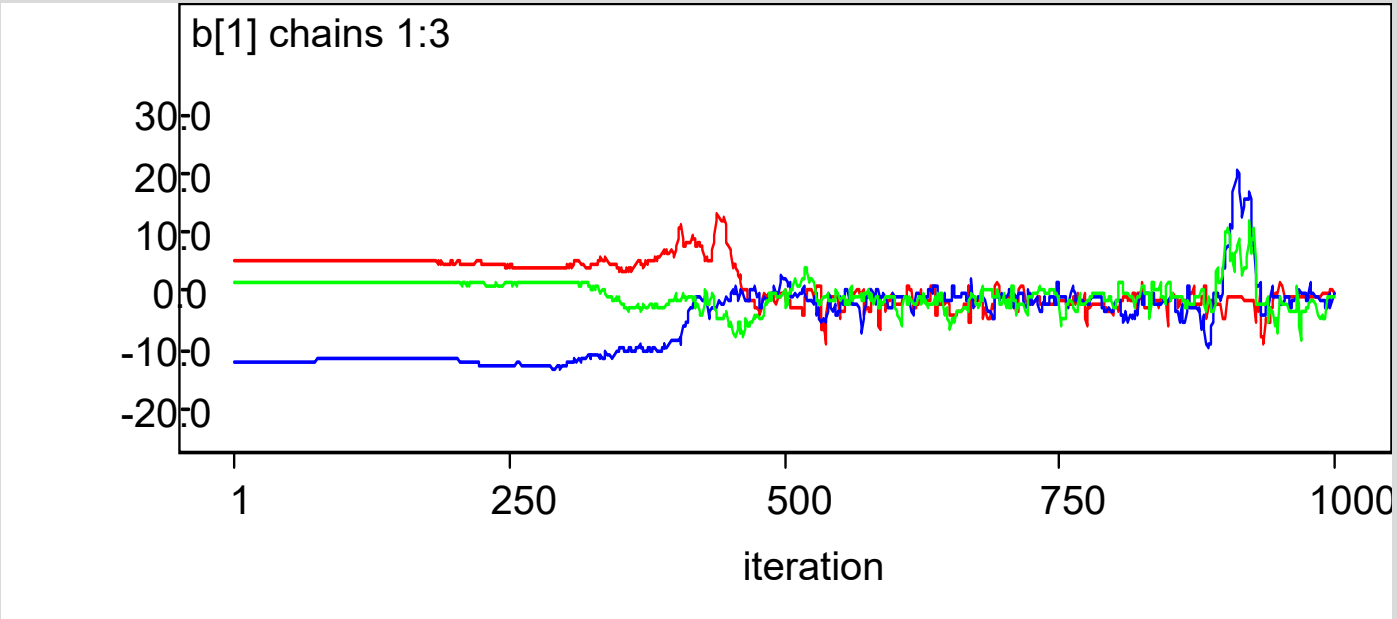


Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- An analysis of variance type argument
- $PSRF$ or $\hat{R} =$

$$\frac{\text{Total Variance}}{\text{Within Chain Variance}} = \frac{\text{Between Chain Variance} + \text{Within Chain Variance}}{\text{Within Chain Variance}}$$

- If there is substantial between-chain variance, will be >> 1



Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- Run ***multiple*** chains from dispersed starting points
- Suggest convergence when the chains come together
- Operationalized in terms of partitioning variability
- Run multiple chains for $2T$ iterations, discard first half
- Examine between and within chain variability
- Various versions, modifications suggested over time

Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- For any θ, for any chain c the within-chain variance is

$$W_c = \frac{1}{T-1}\sum_{t=1}^{T}(\theta_{(c)}^{(t)} - \bar{\theta}_{(c)})^2$$

- For all chains, the pooled within-chain variance is

$$W = \frac{1}{C}\sum_{c=1}^{C} W_c = \frac{1}{C(T-1)}\sum_{c=1}^{C}\sum_{t=1}^{T}(\theta_{(c)}^{(t)} - \bar{\theta}_{(c)})^2$$

Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- The between-chain variance is

$$B = \frac{T}{C-1}\sum_{c=1}^{C}(\bar{\theta}_{(c)} - \bar{\theta})^2$$

- The estimated variance is

$$\hat{Var}(\theta) = (T - 1/T)W + (1/T)B$$

Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

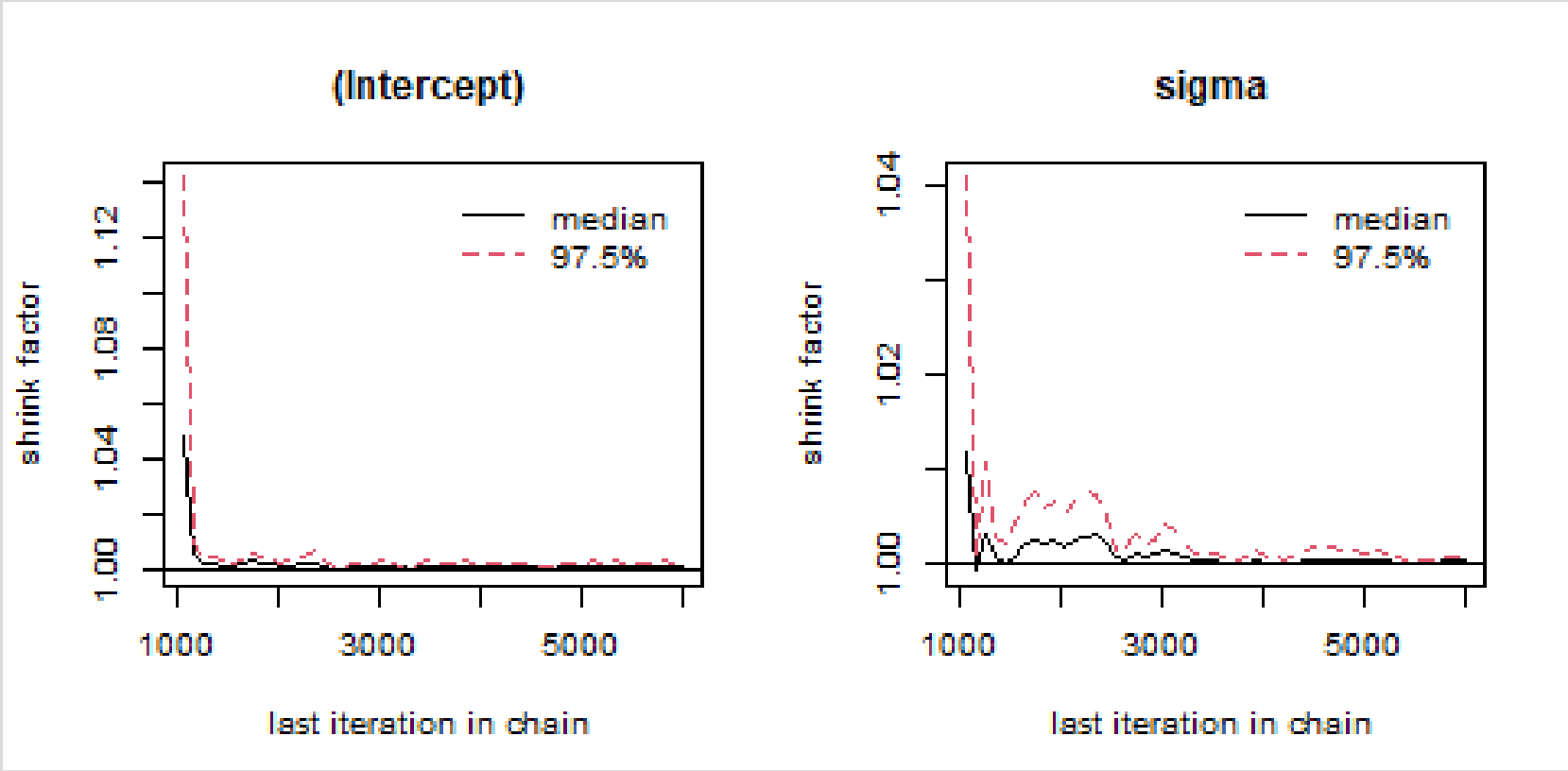
- The potential scale reduction factor is

$$\hat{R} = \sqrt{\frac{\hat{V}ar(\theta)}{W}}$$

- If close to 1 (e.g., < 1.1) for all parameters, can conclude convergence

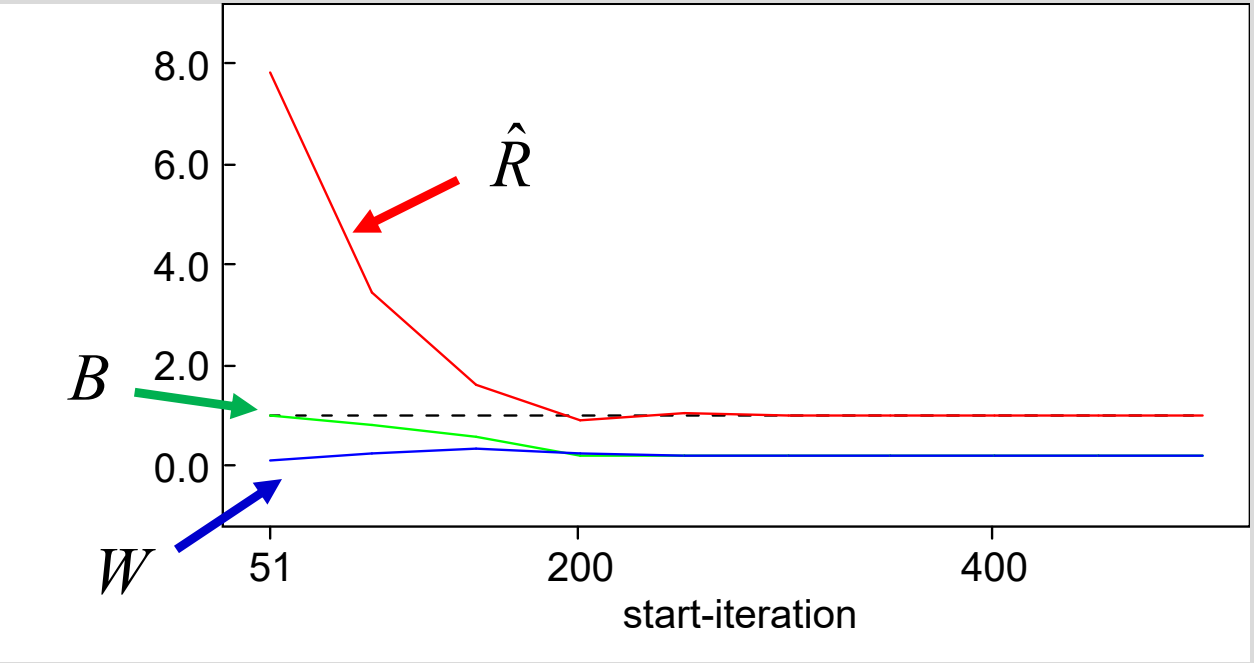
Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- Examine it over "time", look for $\hat{R} \rightarrow 1$, stability of B and W
- If close to 1 (e.g., < 1.2, or < 1.1) can conclude convergence
- `gelman.plot()` in R



Gelman & Rubin's (1992) Potential Scale Reduction Factor (PSRF, $\hat{R}$)

- Examine it over "time", look for $\hat{R} \rightarrow 1$, stability of B and W
- If close to 1 (e.g., < 1.2, or < 1.1) can conclude convergence



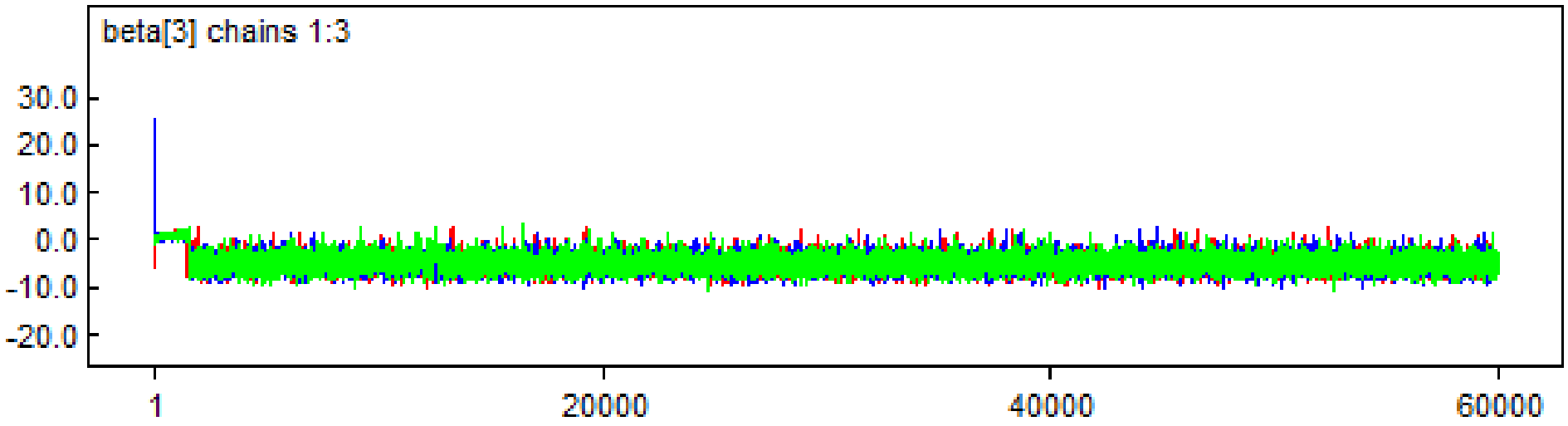
Running Stan via R For 10 Iterations

```
Warning messages:
1: The largest R-hat is 1.27, indicating chains have not mixed.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#r-hat
2: Bulk Effective Samples Size (ESS) is too low, indicating
posterior means and medians may be unreliable.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#bulk-ess
3: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
Running the chains for more iterations may help. See
http://mc-stan.org/misc/warnings.html#tail-ess
```

Whoa! Will need to run more iterations…

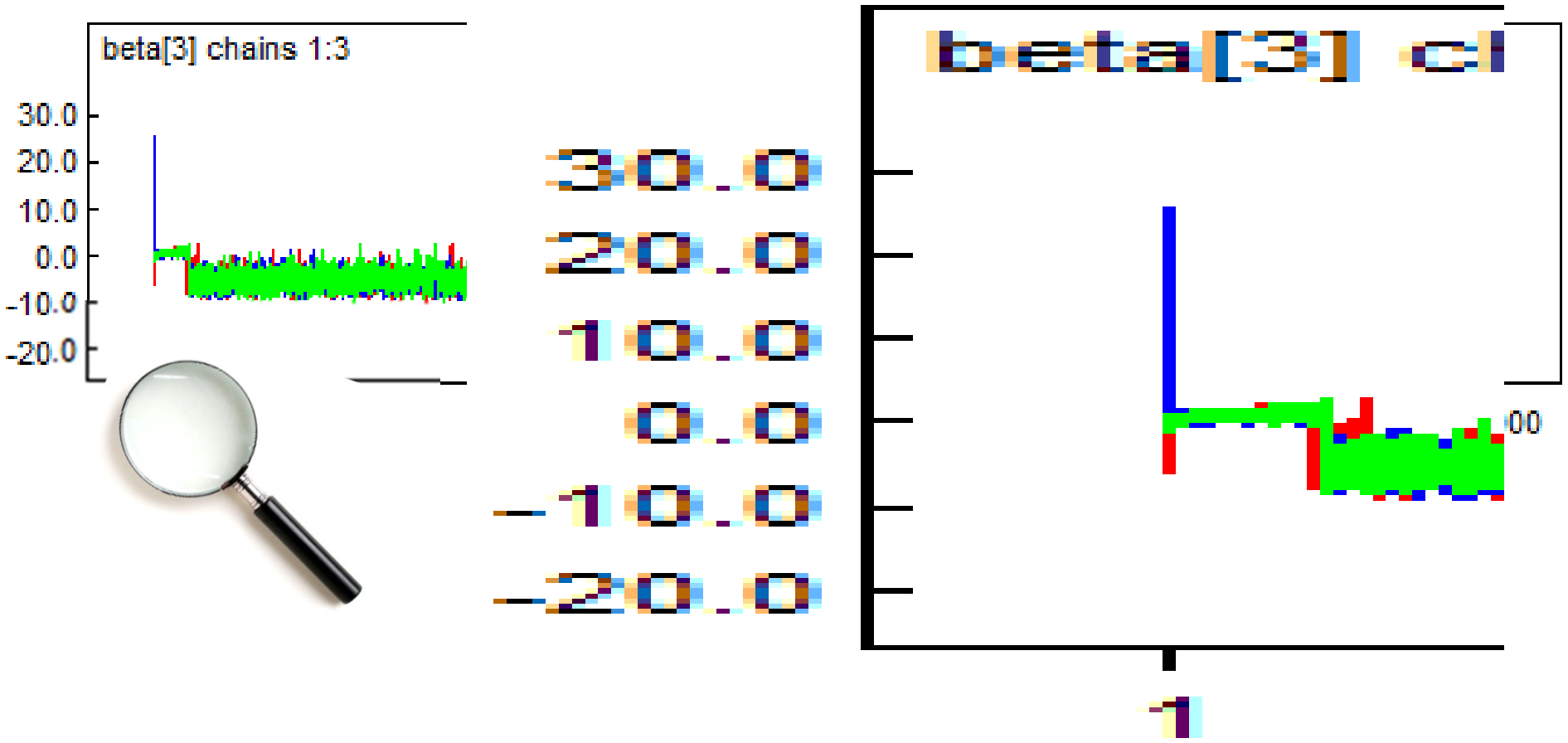
Assessing Convergence: No Guarantees

Multiple chains coming together does not guarantee they have converged



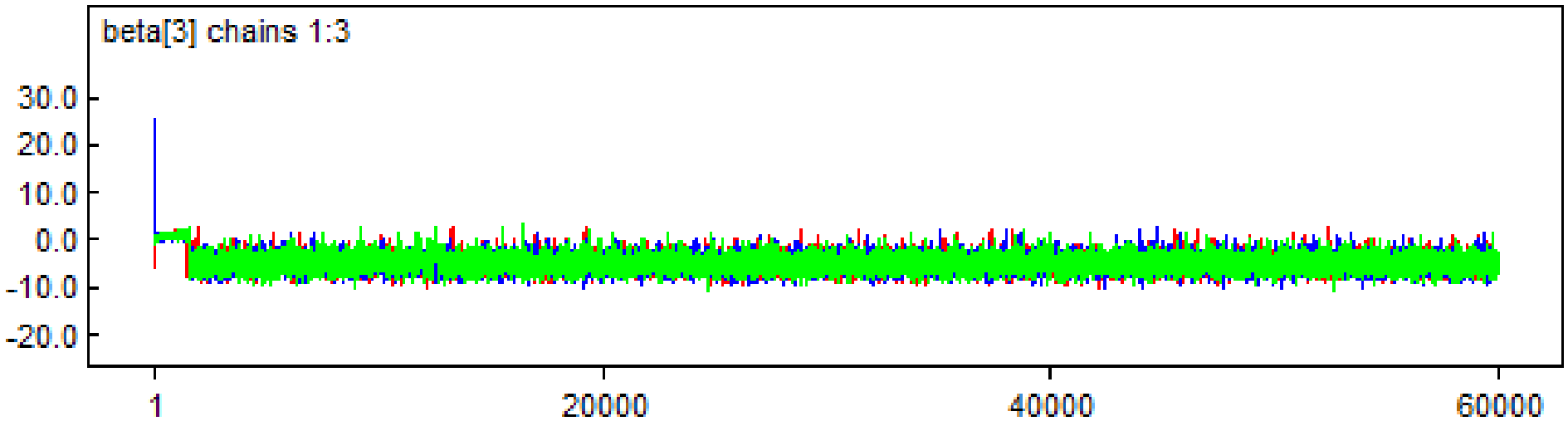
Assessing Convergence: No Guarantees

Multiple chains come together does not guarantee they have converged



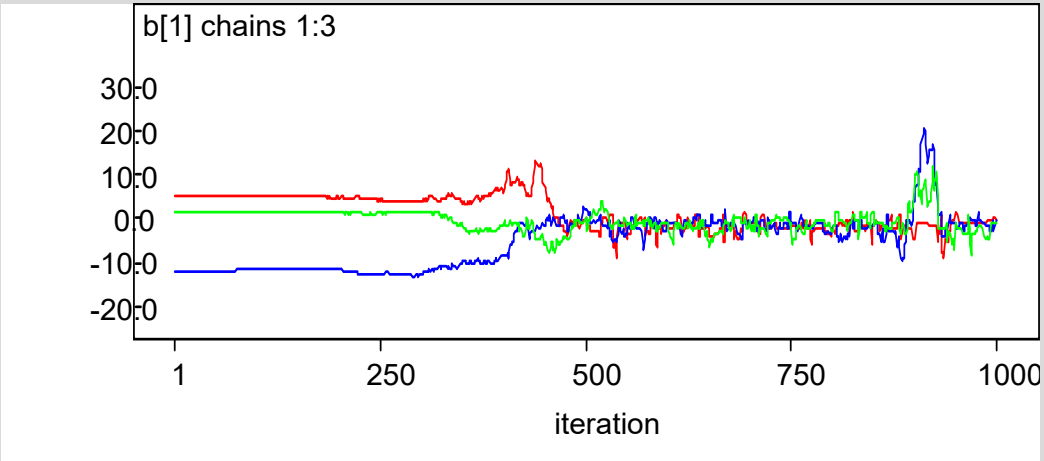
Assessing Convergence: No Guarantees

Multiple chains coming together does not guarantee they have converged



Assessing Convergence

- Recommend running multiple chains far apart and determine when they reach the same "place"
  - PSRF criterion an approximation to this
  - Akin to starting ML from different start values and seeing if they reach the same maximum
  - Here, convergence to a distribution, not a point
- A chain hasn't converged until ***all*** parameters converged
  - Brooks & Gelman multivariate PSRF



Assessing Convergence

Iterations prior to convergence should be discarded as ***burn-in***, and ***not*** used for inference

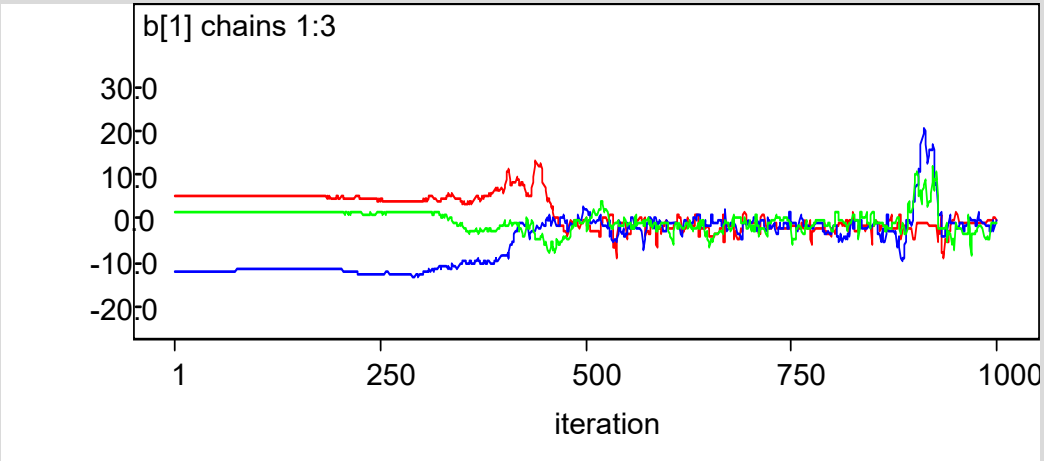
far apart and determine

is

rt values and seeing if they

not a point

- A chain hasn't converged until ***all*** parameters converged
  - Brooks & Gelman (1998) multivariate PSRF



Assessing Convergence

far apart and determine

is

rt values and seeing if they

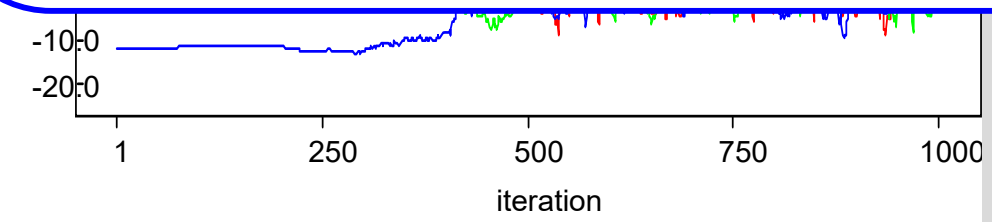
Iterations prior to convergence should be discarded as ***burn-in***, and ***not*** used for inference

Stan has a ***warmup*** phase, in which iterations are used to tune the algorithm

This is often sufficient for convergence (i.e., no additional burn-in needed)

Especially for simple models

- A chain hasn
  - Brooks &



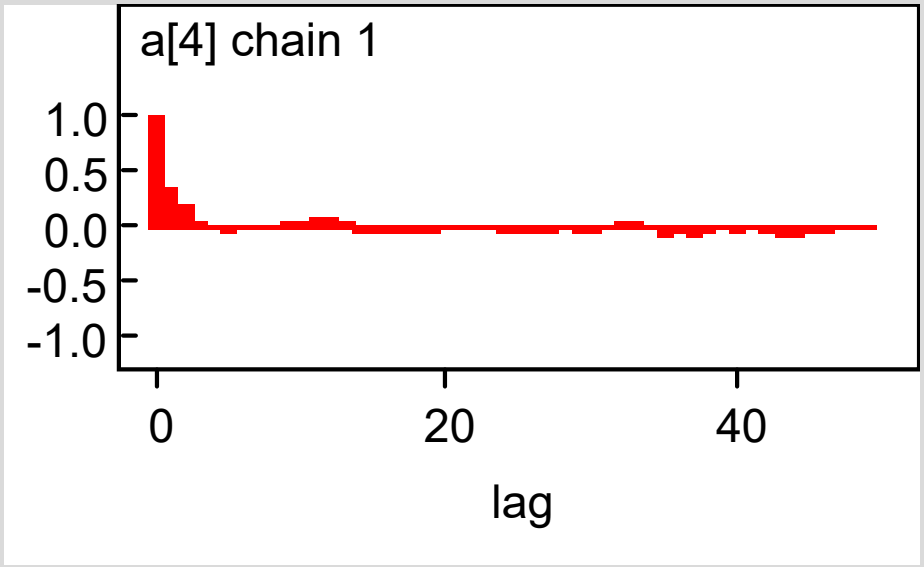
Serial Dependence

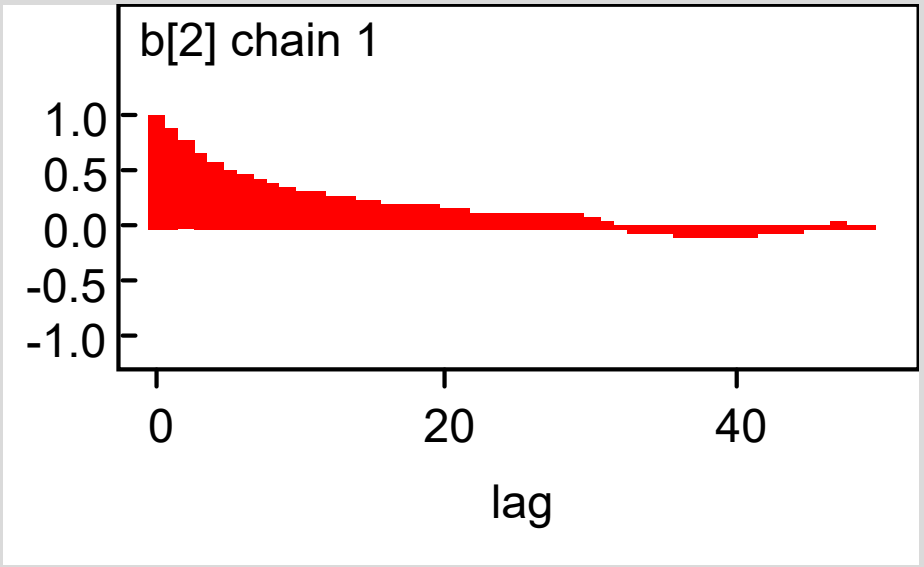
Serial Dependence

- Serial dependence between draws due to the dependent nature of the draws (i.e., the Markov structure)
- $p(\theta^{(t+1)} \mid \theta^{(t)}, \theta^{(t-1)}, \theta^{(t-2)}, \ldots) = p(\theta^{(t+1)} \mid \theta^{(t)})$

- However there is a ***marginal*** dependence across multiple lags
- Can examine the autocorrelation across different lags

Autocorrelation





Thinning

- Can "thin" the chain by dropping certain iterations

  Thin = 1 → keep every iteration

  Thin = 2 → keep every other iteration (1, 3, 5,…)

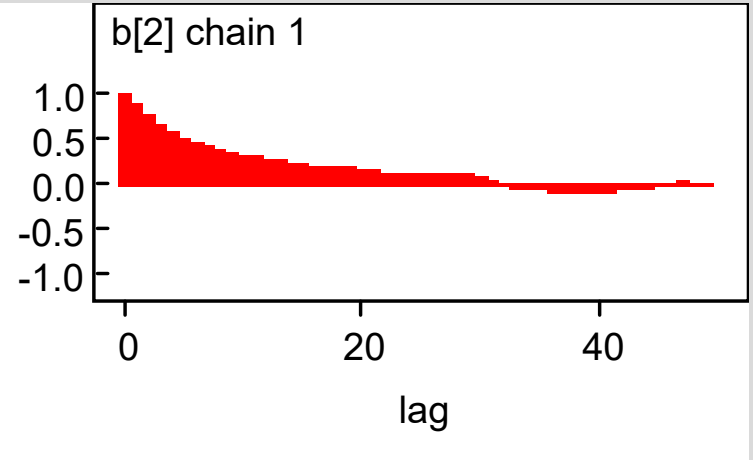
  Thin = 5 → keep every 5^{th} iteration (1, 6, 11,…)

  Thin = 10 → keep every 10^{th} iteration (1, 11, 21,…)

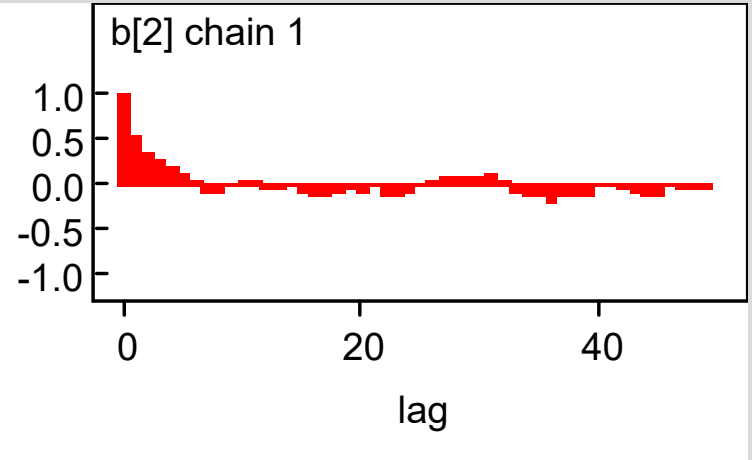
  Thin = 100 → keep every 100^{th} iteration (1, 101, 201,…)

Thinning

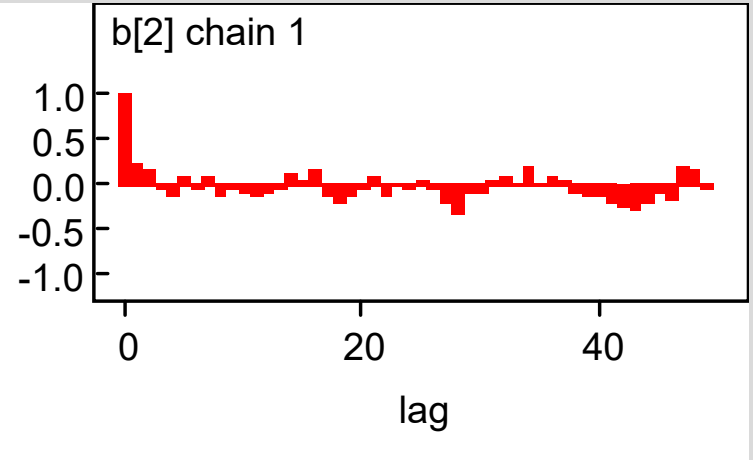
Thin = 1



Thin = 5



Thin = 10



Thinning

- Can "thin" the chain by dropping certain iterations

  Thin = 1 → keep every iteration

  Thin = 2 → keep every other iteration (1, 3, 5,…)

  Thin = 5 → keep every 5^{th} iteration (1, 6, 11,…)

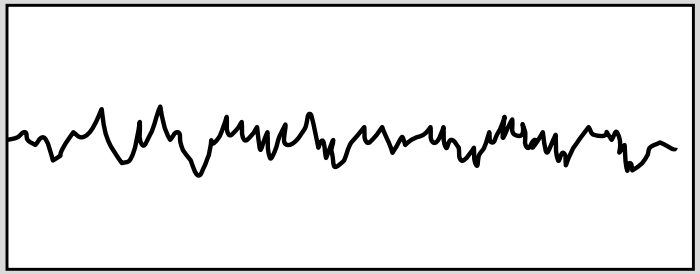
  Thin = 10 → keep every 10^{th} iteration (1, 11, 21,…)

  Thin = 100 → keep every 100^{th} iteration (1, 101, 201,…)
- Thinning ***does not*** provide a better portrait of the posterior
  - A loss of information
- May want to keep, and account for time-series dependence
- Useful when data storage, other computations an issue
  - *I want 1000 iterations, rather have 1000 approximately independent iterations*
- Dependence ***within*** chains, but none ***between*** chains

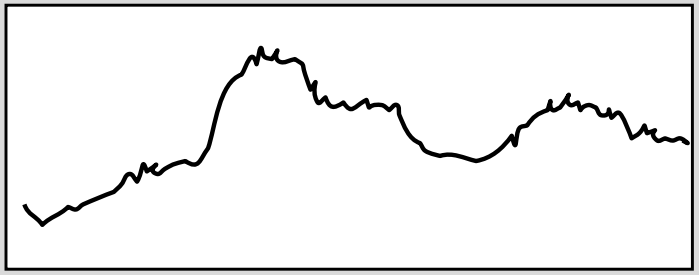
Mixing

Mixing

- We don't want the sampler to get "stuck" in some region of the posterior, or ignore a certain area of the posterior
- Mixing refers to the chain "moving" throughout the support of the distribution in a reasonable way



relatively good mixing



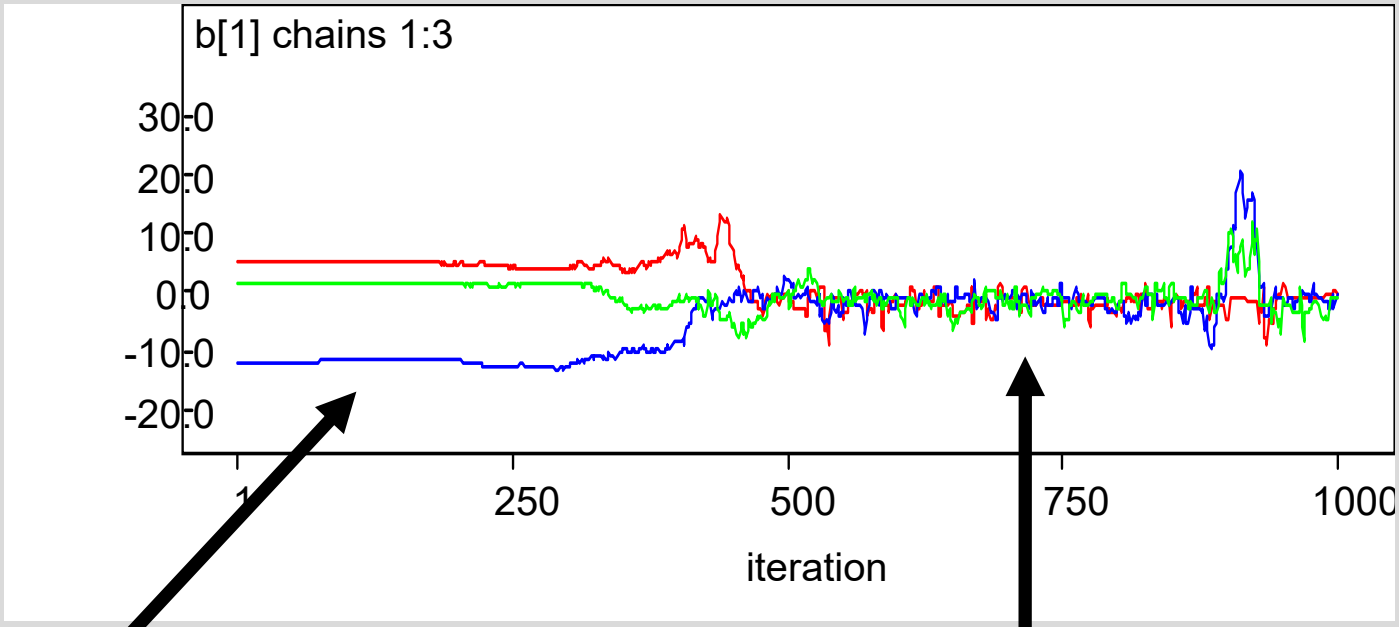
relatively poor mixing

Mixing

- Mixing ≠ convergence, but better mixing usually leads to faster convergence
- Mixing ≠ autocorrelation, but better mixing usually goes with lower autocorrelation (and cross-correlations between parameters)
- With better mixing, then for a given number of MCMC iterations, get more information about the posterior
  - Ideal scenario is independent draws from the posterior
- With worse mixing, need more iterations to (a) achieve convergence and (b) achieve a desired level of precision for the summary statistics of the posterior

Mixing

- Chains may mix differently at different times
- Often indicative of an adaptive MCMC algorithm



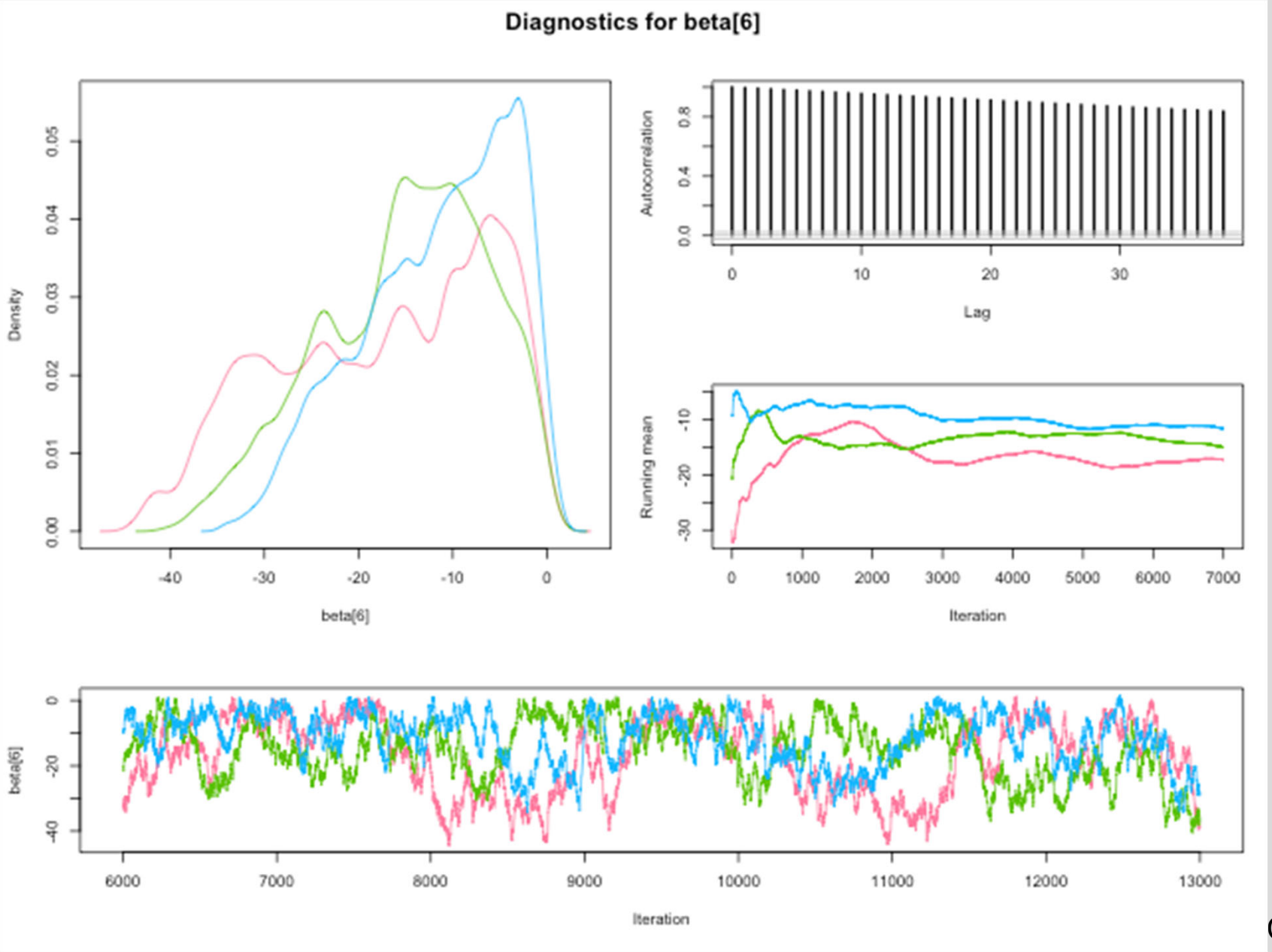
relatively poor mixing

relatively good mixing

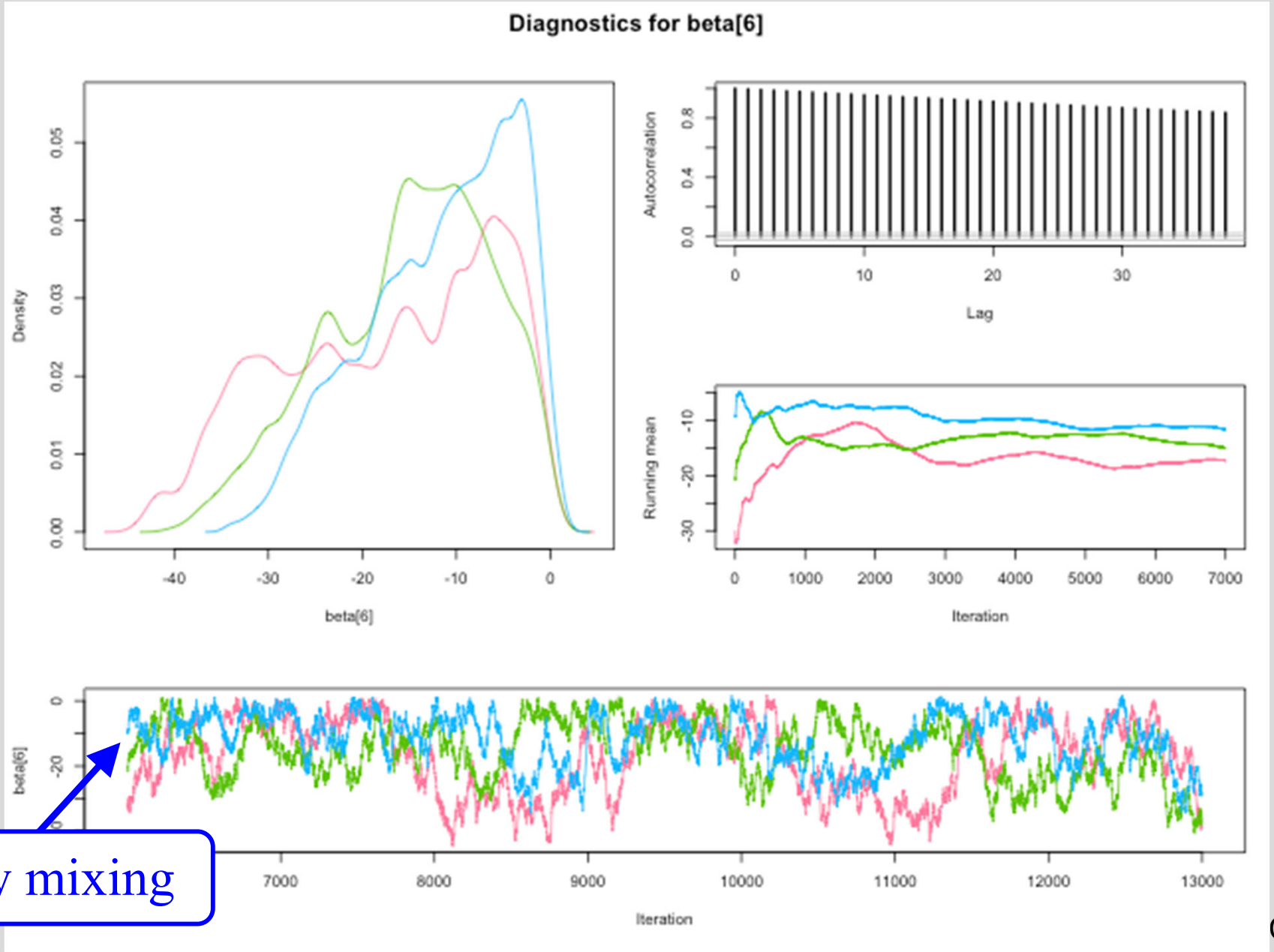
Mixing

- Slow mixing can also be caused by high dependence between parameters
  - Example: multicollinearity…
- Reparameterizing the model can improve mixing
  - Example: centering predictors in regression

Example Results

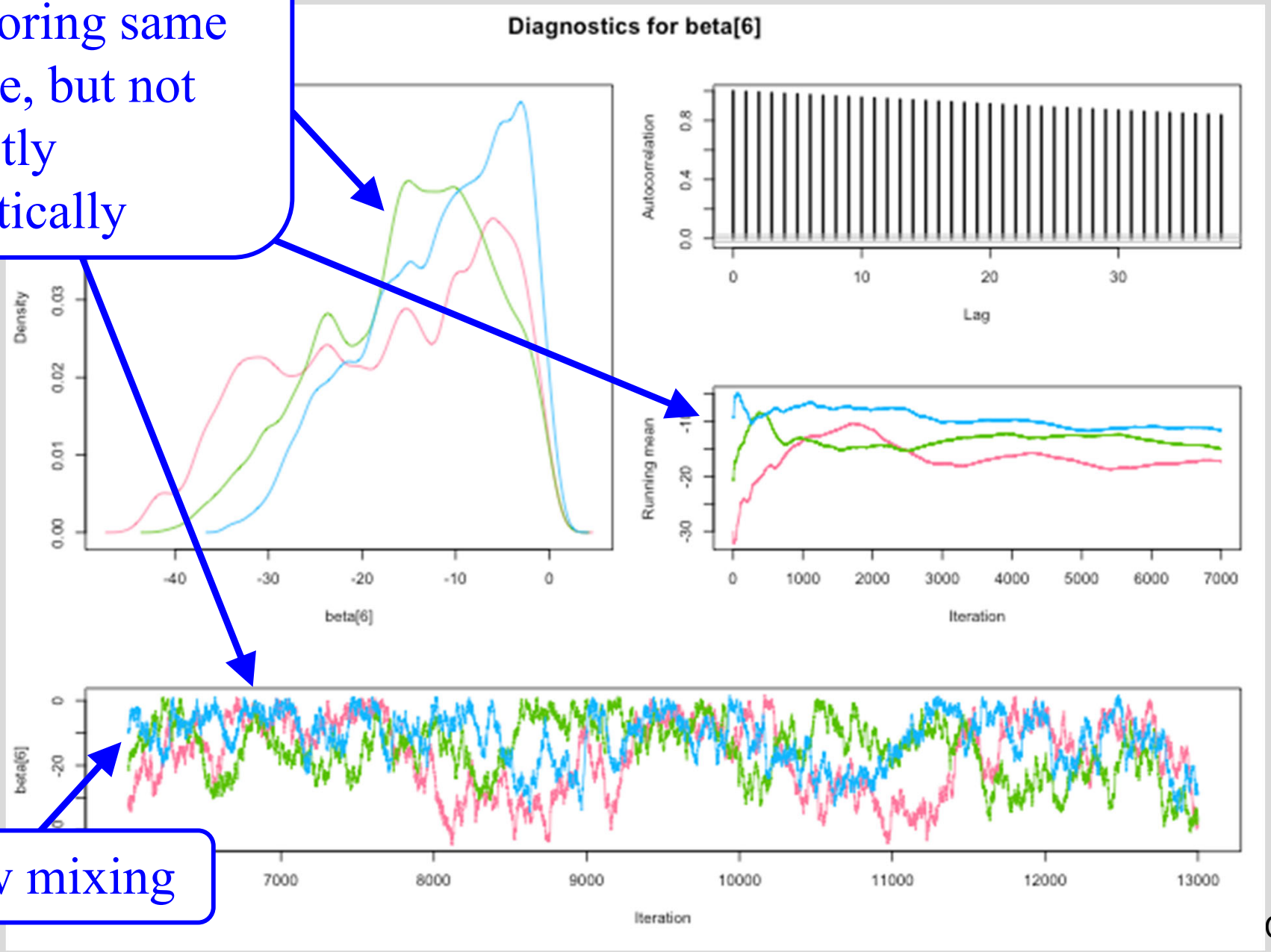


Example Results



ults

Chains broadly exploring same space, but not exactly identically

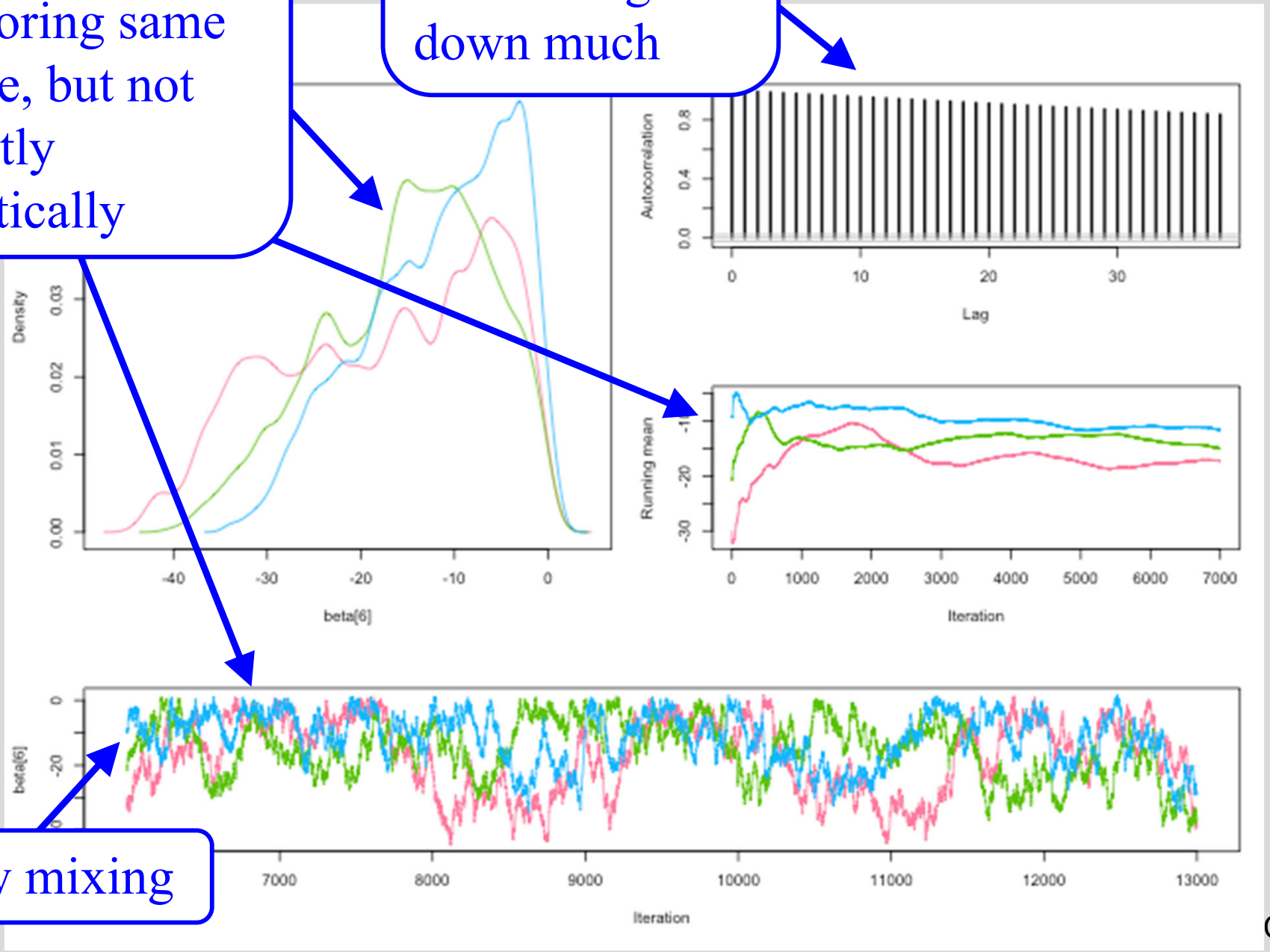


Slow mixing

ults

Chains broadly exploring same space, but not exactly identically

High autocorrelation, not coming down much

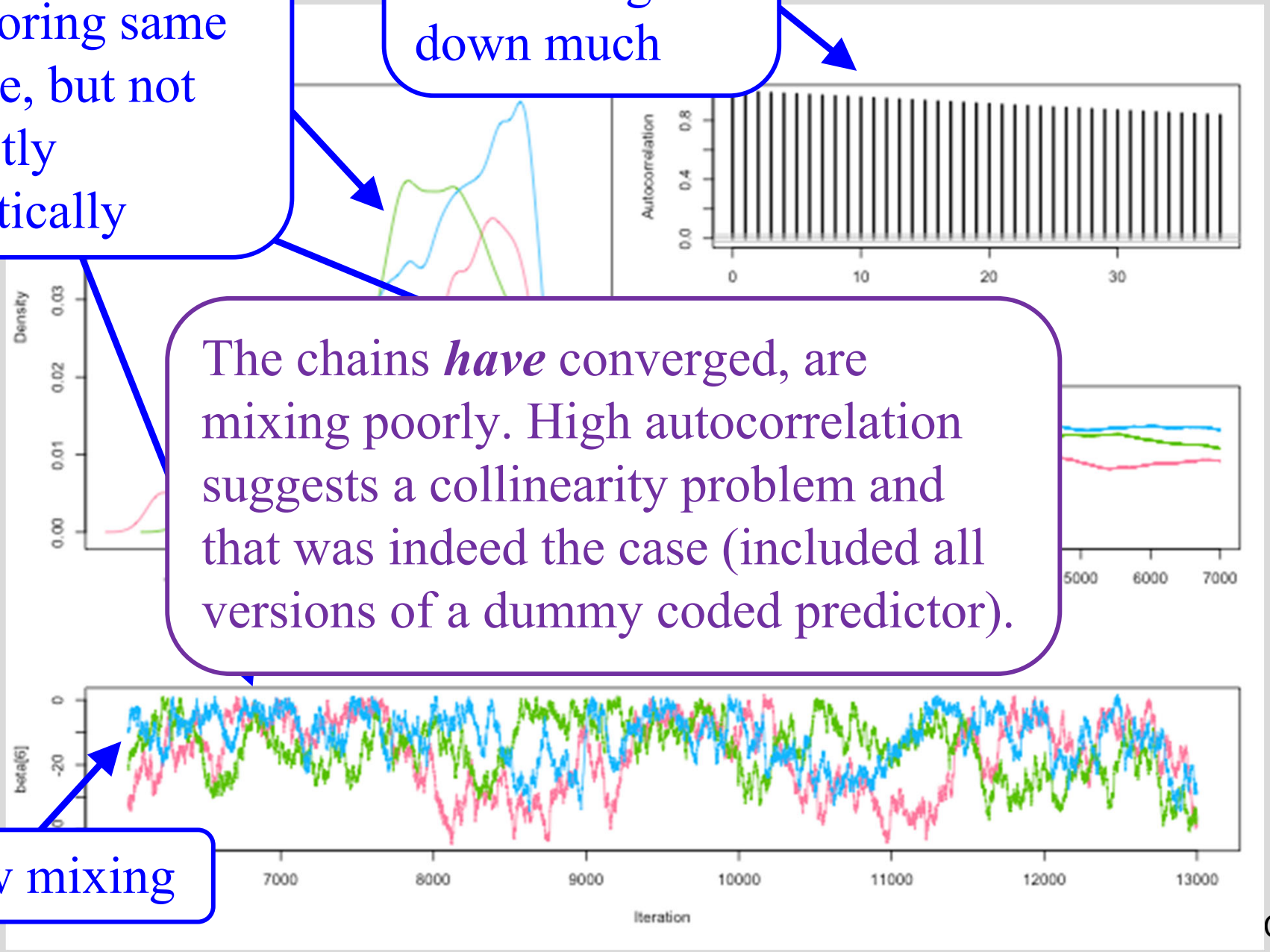


Slow mixing

ults

Chains broadly exploring same space, but not exactly identically

High autocorrelation, not coming down much



The chains ***have*** converged, are mixing poorly. High autocorrelation suggests a collinearity problem and that was indeed the case (included all versions of a dummy coded predictor).

Slow mixing

Stopping the Chain(s)

When to Stop The Chain(s)

- Discard the iterations prior to convergence as ***burn-in***
- How many more iterations to run?
  - As many as you want ☺
  - As many as time provides
- Autocorrelaion complicates things
- Rely on empirical results…

When to Stop The Chain(s)

Effective sample size

- Approximation of how many independent samples we have
- As proposed by Kass, Carlin, Gelman, & Neal (1998):

$$\mathrm{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \mathrm{ACF}(k)}$$

  - where T is the raw number of iterations and
  - ACF(k) is the autocorrelation of the chain at lag k.
- In practice, stop summation when ACF(k) < 0.05
- The idea is to divide the number of iterations by the amount of autocorrelation
- Different software use different algorithms to get at the same idea

When to Stop The Chain(s)

Effective sample size

- Approximation of how many independent samples we have
- The idea is to divide the number of iterations by the amount of autocorrelation

How big should ESS be? It depends on what you're looking at!

- Features of the posterior that are governed mostly by the dense regions (e.g., the median) require fewer
- Features of the posterior that are heavily influenced by sparse regions (e.g., limits of the 95% HDI) need more
- Kruschke (2014) suggests ESS of at least 10,000 for limits of 95% HDI

When to Stop The Chain(s)

Software may provide the "MC error" or "MC standard error"

- Estimate of the sampling variability of the sample mean
  - Sample here is the sample of ***iterations***
- Accounts for the dependence between iterations

$$\text{MCSE} = \frac{\text{SD from the chain}}{\sqrt{\text{ESS}}}$$

- Conveys the estimated SD of the sample mean of the chain on the scale of the parameter
  - How much is the sample mean from the chain going to bounce around?
- Guideline is to go at least until MC error is less than 5% of the posterior standard deviation

Steps in MCMC in Practice

Steps in MCMC (1)

- Setup MCMC using any of a number of algorithms
  - Program yourself (have fun ☺)
  - Use existing software (Stan, JAGS, BUGS, etc.)
- Run a number of iterations for chain(s)
  - Stan includes a ***warmup*** period for the chains to get going

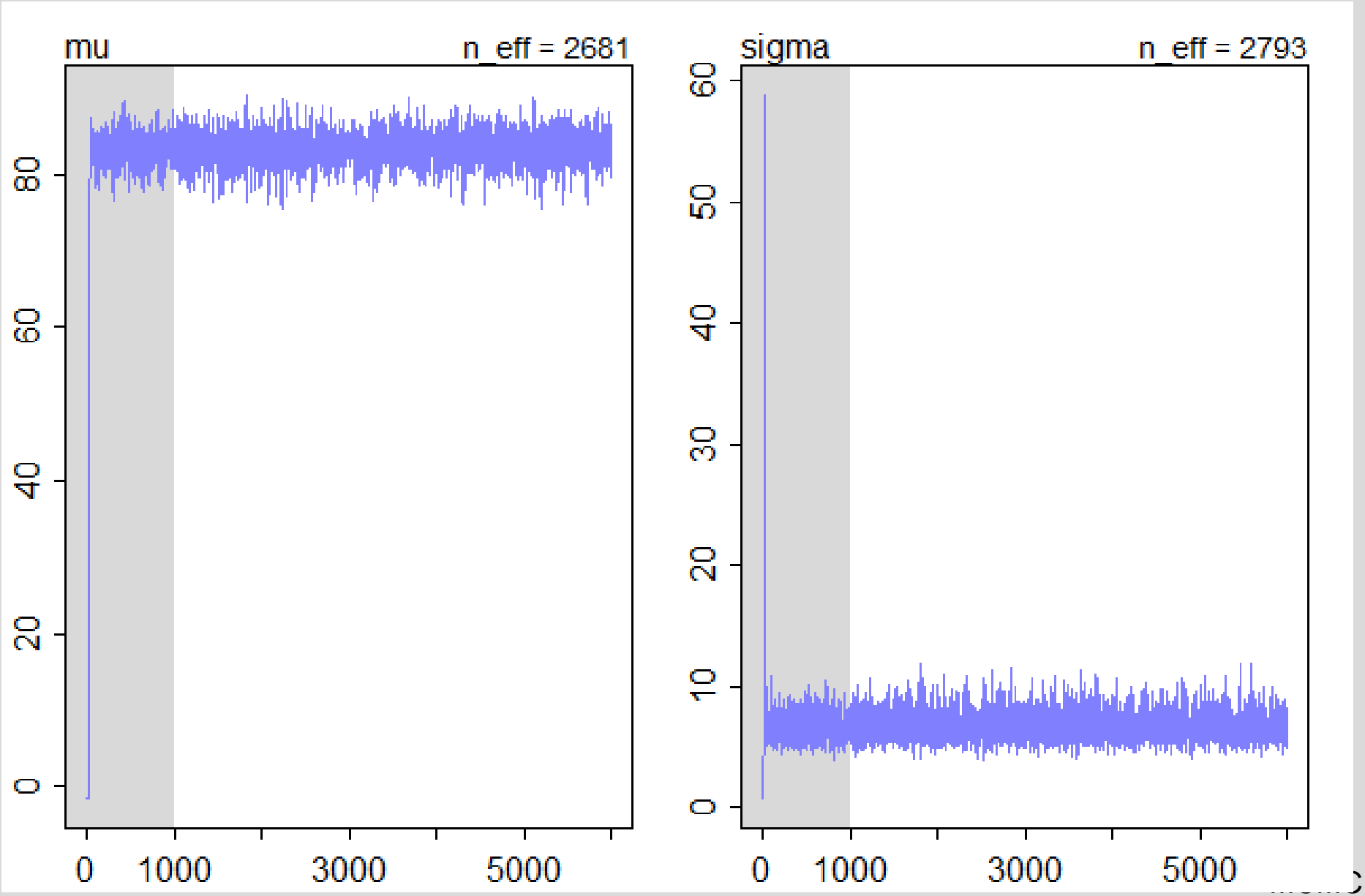
Running Stan via R

```
# Choose features of MCMC --------
#   the number of chains
#   the number of iterations to warmup
#   the total number of iterations after warmup
n.chains = 1
n.warmup = 1000
n.iters.per.chain.after.warmup = 5000
n.iters.total.per.chain =
```

Specifying 1000 iterations as warmup

Requesting 5000 iterations: number of simulations from the distribution, after warmup

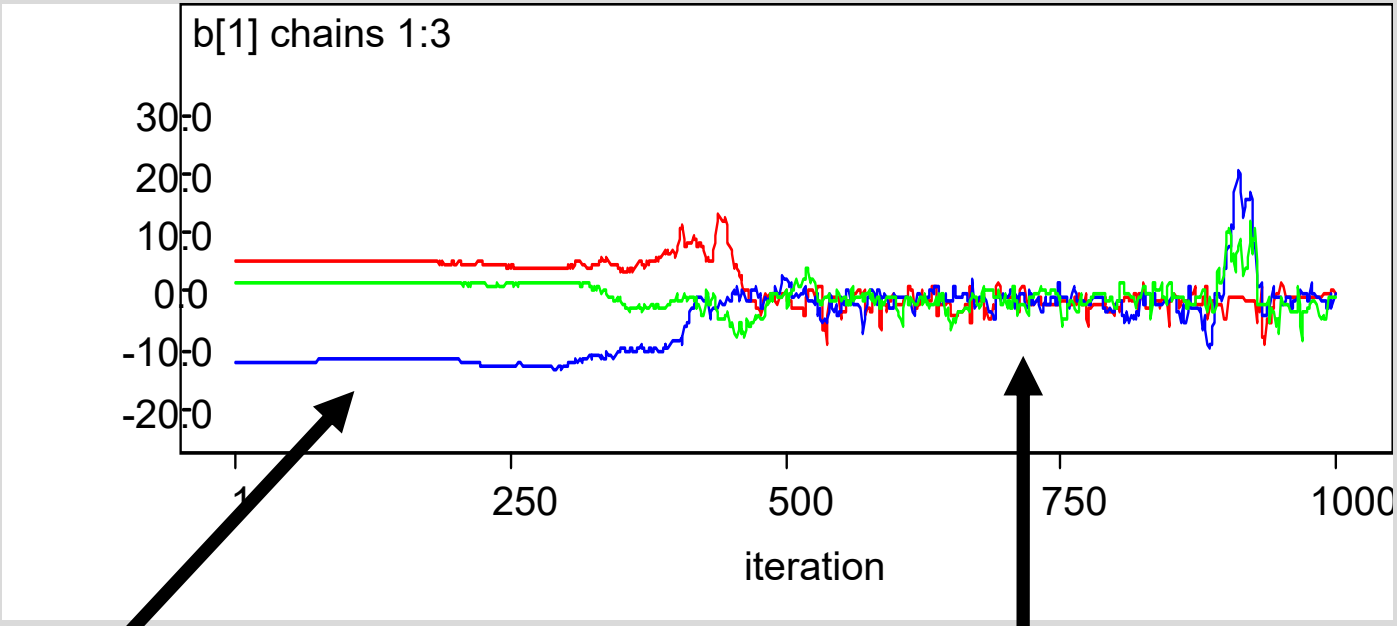
Trace Plot with Warmup



Steps in MCMC (1)

- Setup MCMC using any of a number of algorithms
  - Program yourself (have fun ☺)
  - Use existing software (Stan, JAGS, BUGS, etc.)
- Run a number of iterations for chain(s)
  - Stan includes a ***warmup*** period for the chains to get going
  - Other software may have analogous functions

Adapting MCMC → Automatic Discard



relatively poor mixing during adaptive phase

relatively good mixing after adaptive phase

Steps in MCMC (2)

- Diagnose convergence
  - Monitor trace plots, PSRF ($\hat{R}$) criteria
- Discard iterations prior to convergence as ***burn-in***
  - Software may indicate a minimum number of iterations needed
  - Examples include warmup or adaptive phases
  - A lower bound
  - Stan produces warnings…

Running Stan for 10 Iterations

```
Warning messages:
1: The largest R-hat is 1.14, indicating chains have not mixed.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#r-hat
2: Bulk Effective Samples Size (ESS) is too low,
indicating posterior means and medians may be unreliable.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#bulk-ess
3: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#tail-ess
4: Markov chains did not converge! Do not analyze results!
```

Whoa! Will need to run more iterations…

Steps in MCMC (3)

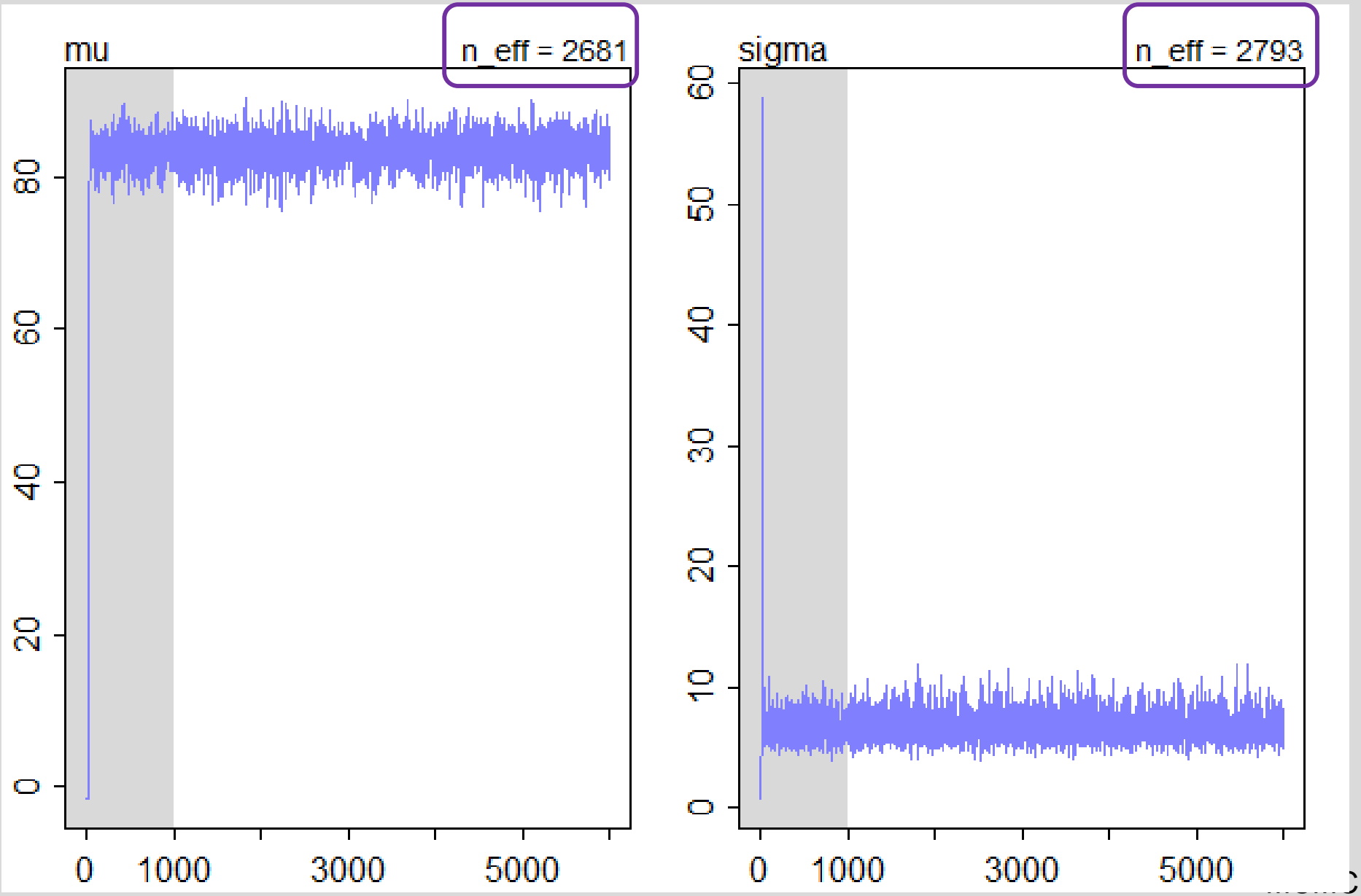
- Run the chain for a desired number of iterations
  - Understanding serial dependence/autocorrelation
  - Understanding mixing
  - Effective sample size (ESS) or MC error size as a strategy
  - Software may provide warnings…

Running Stan via R

```
Warning messages:
1: The largest R-hat is 1.14, indicating chains have not mixed.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#r-hat
2: Bulk Effective Samples Size (ESS) is too low,
indicating posterior means and medians may be unreliable.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#bulk-ess
3: Tail Effective Samples Size (ESS) is too low, indicating
posterior variances and tail quantiles may be unreliable.
Running the chains for more iterations may help.
See http://mc-stan.org/misc/warnings.html#tail-ess
4: Markov chains did not converge! Do not analyze results!
```

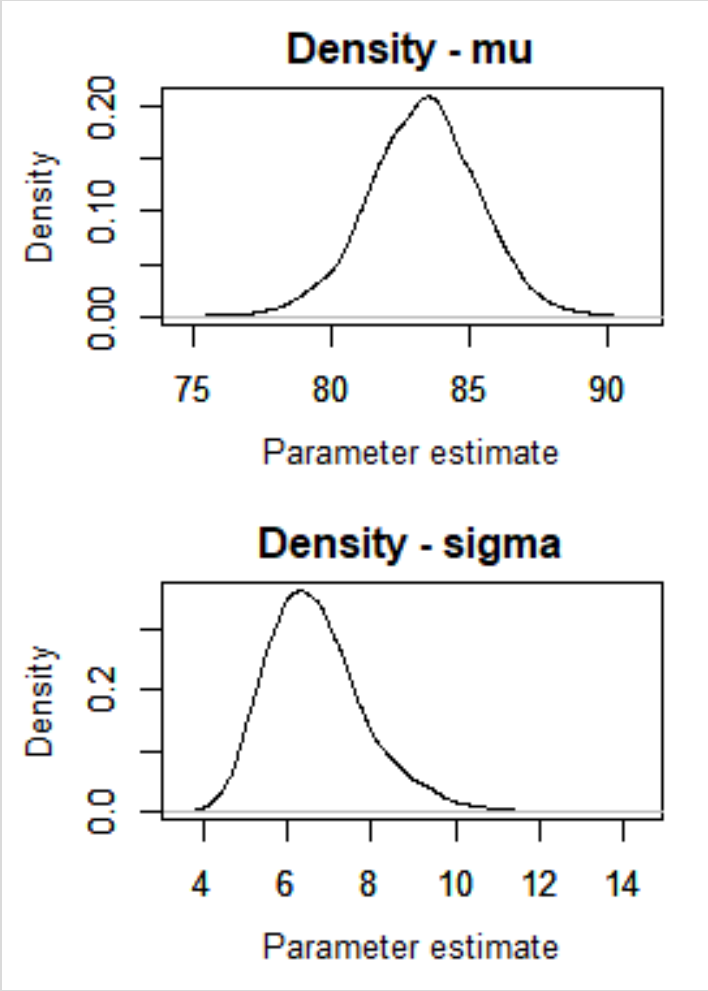
Whoa! Will need to run more iterations…

Trace Plot with Warmup



Steps in MCMC (4)

- Summarize results
  - Monte Carlo principle
  - Densities
  - Summary statistics



	mean	sd	95%_HPDL	95%_HPDU	median
mu	83.30	2.00	79.16	87.11	83.34
sigma	6.69	1.16	4.65	9.11	6.56

Example Write-Up

Example Write-Up

Four chains were run from dispersed starting points for xxxx iterations. Visual inspection of trace plots and $\hat{R}$ suggested xxxx mixing and convergence by xxxx iterations. Subsequent iterations were combined from the chains, yielded a total of xxxx iterations. The effective sample sizes ranged from xxxx to xxxx. The posterior distribution is summarized in Figure xxxx and Table xxxx.

Summary and Conclusion

Summary

- Dependence on initial values is “forgotten” after a sufficiently long run of the chain (memoryless)
- Convergence to a ***distribution***
  - Recommend monitoring multiple chains
  - PSRF ($\hat{R}$) as approximation
- Let the chain “burn-in” or “warmup”
  - Discard draws prior to convergence
  - Retain the remaining draws as draws from the posterior

Summary

- Dependence across draws induce autocorrelations
  - Can thin if desired
- Dependence across draws within and between parameters can slow mixing
  - Reparameterizing may help
- Run as many iterations as needed to adequately depict the posterior, or its features of interest
  - Effective sample size (ESS) a guideline
- Summarize results of iterations

Role of MCMC in History of Bayesian Analyses

- Prior to MCMC, applications somewhat limited
  - Bayes seen as conceptually elegant, computationally intractable
  - Conjugate priors or large sample approximations

- Applicability MCMC not immediately recognized

  "the technique is unlikely to be particularly helpful in many other than binary situations and the Markov chain itself has no practical interpretation"

  -- Besag (1975, p. 187)

- MCMC supports any model you can write down

  "The Bayesian 'machine' together with MCMC is arguably the most powerful mechanism ever created for processing data and knowledge."

  -- Berger (2000, p. 1273)

Wise Words of Caution

Beware: MCMC sampling can be dangerous!

-- Spiegelhalter, Thomas, Best, & Lunn (2007)
(WinBUGS User Manual)

Putting It All Together

See 'Normal model (mu and sigma.squared unknown) in Stan via rstanarm, My Workflow.R'