Today

- Recap & questions from homework
  - Stan etc installs?
- Maybe an example to work on

Reproducible workflows

- Reproducible analysis reports
  - not necessary but nice
  - supplementary material, data repositories
- Reports from .R files
- Any .R file can be knitted
- Can also include markup
  - e.g. 03_9_train_ssq_grid.R

McElreath Ch8

- Learning goals:
- Understand and use MCMC algorithms to sample from the posterior distribution
- Recognize and fix bad sampling scenarios
- Use HMC implemented in Stan via R packages

Main points McElreath Ch8

- MCMC: Monte Carlo Markov Chain
- Series of random numbers where each number depends on the previous one
- Sample less from low probability areas; more bang for your random buck
- Algorithms
  - Metropolis-Hastings
  - Gibbs sampling
  - HMC: Hamiltonian Monte Carlo
- We'll mostly use HMC

MCMC algorithms

Algorithm (general)

for many iterations

    propose new value for parameter

    calculate the probability of accepting the proposal:

        P_accept = min(Pr(proposal) / Pr(current), 1)

    accept proposal randomly with Bern(P_accept)

plot posterior distribution (histogram) of parameter values

where Pr() = prior x likelihood

Rosenbluth algorithm

aka Metropolis-Hastings

Algorithm (original)

for many iterations

    propose new value for parameter:

        draw Unif(-max_d, max_d)

        proposal = current parameter + draw

    calculate the probability of accepting the proposal:

        P_accept = min(Pr(proposal) / Pr(current), 1)

    accept proposal randomly with Bern(P_accept)

plot posterior distribution (histogram) of parameter values

where Pr() = prior x likelihood

MCMC algorithms

- Two other important algorithms
- Gibbs sampling
  - needs conjugate priors
  - prior such that the posterior is the same as the prior
  - e.g. norm prior x norm lik = norm posterior
  - beta prior x binom lik = beta posterior
- Hamiltonion Monte Carlo (HMC)

MCMC algorithms

- Get an intuition for their behavior:
- https://chi-feng.github.io/mcmc-demo/app.html#HamiltonianMC,standard

Bayesian tools

- Stan
  - Gelman group
  - Hamiltonian Monte Carlo
  - Betancourt (2017) A conceptual introduction to Hamiltonian Monte Carlo (https://arxiv.org/abs/1701.02434A)
  - open source
  - models with continuous parameters only
  - state of the art
- http://mc-stan.org

Bayesian tools

- BUGS (Bayesian inference Using Gibbs Sampling) (and Metropolis-Hastings)
- http://www.openbugs.info
- Older, original standard tool for MCMC
- Exceedingly difficult to run on Mac
- Many newer tools are based on BUGS code style
- Lots of books and publications use BUGS
- Recommend: need to know about historically but don't use anymore

Bayesian tools

- JAGS (Just Another Gibbs Sampler)
- http://mcmc-jags.sourceforge.net/
  - cross platform, open source
  - basically the same as BUGS
  - often faster
  - highly recommended for models that can't be fit in Stan (e.g. discrete parameters)
  - easy install
- Best to run from R
  - Install R2jags package (install from R)

Bayesian tools

- Others:
- Nimble
  - somewhat common in ecology
- Julia: Turing package

Main points McElreath Ch8

- Using HMC via Stan to fit models
- Now getting posterior samples from HMC
- Use ulam in rethinking to do HMC to follow examples
- Same syntax as sampost

Main points McElreath Ch8

ulam or sampost

```
m1 <- ulam(
  alist(
    y ~ dnorm(mu, sigma),
    mu <- a + b * x,
    a ~ dnorm(0, 100),
    b ~ dnorm(0, 10),
    sigma ~ dcauchy(0, 2)
  ),
  data=d1)
```

Main points McElreath Ch8

ulam or sampost

```
m1 <- ulam(
  alist(
    y ~ dnorm(mu, sigma),
    mu <- a + b * x,
    a ~ dnorm(0, 100),
    b ~ dnorm(0, 10),
    sigma ~ dcauchy(0, 2)
  ),
  data=d1)
```

rstanarm

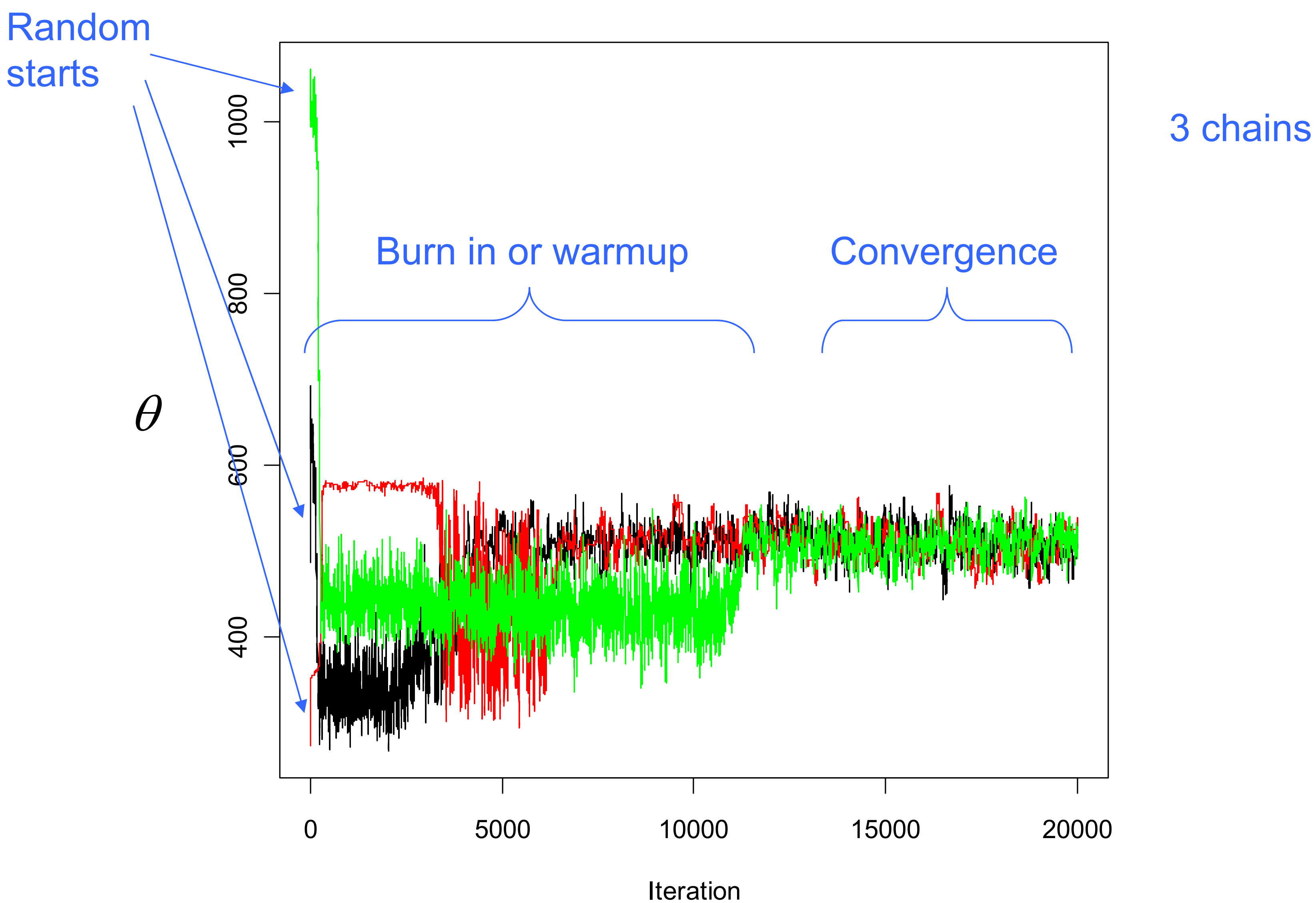
```
m1 <- stan_glm(y ~ x, data=d1)
```

- rstanarm also uses Stan, HMC
  - Stan group's linear models package
  - syntax like lm
  - sensible default priors

Main points McElreath Ch8

- Good choice of priors (weakly informative) can be helpful to tame model fit
  - e.g. Half-Cauchy instead of uniform
- MCMC diagnostics to judge convergence of fit
  - rhat, n_eff
  - plot chain traces ("time series")
- Visualize posteriors
  - histograms, pairs plot

Chains



Random starts
3 chains
Burn in or warmup
Convergence
θ
1000
800
600
400
0
5000
10000
15000
20000
Iteration

Chains

Bad

Good

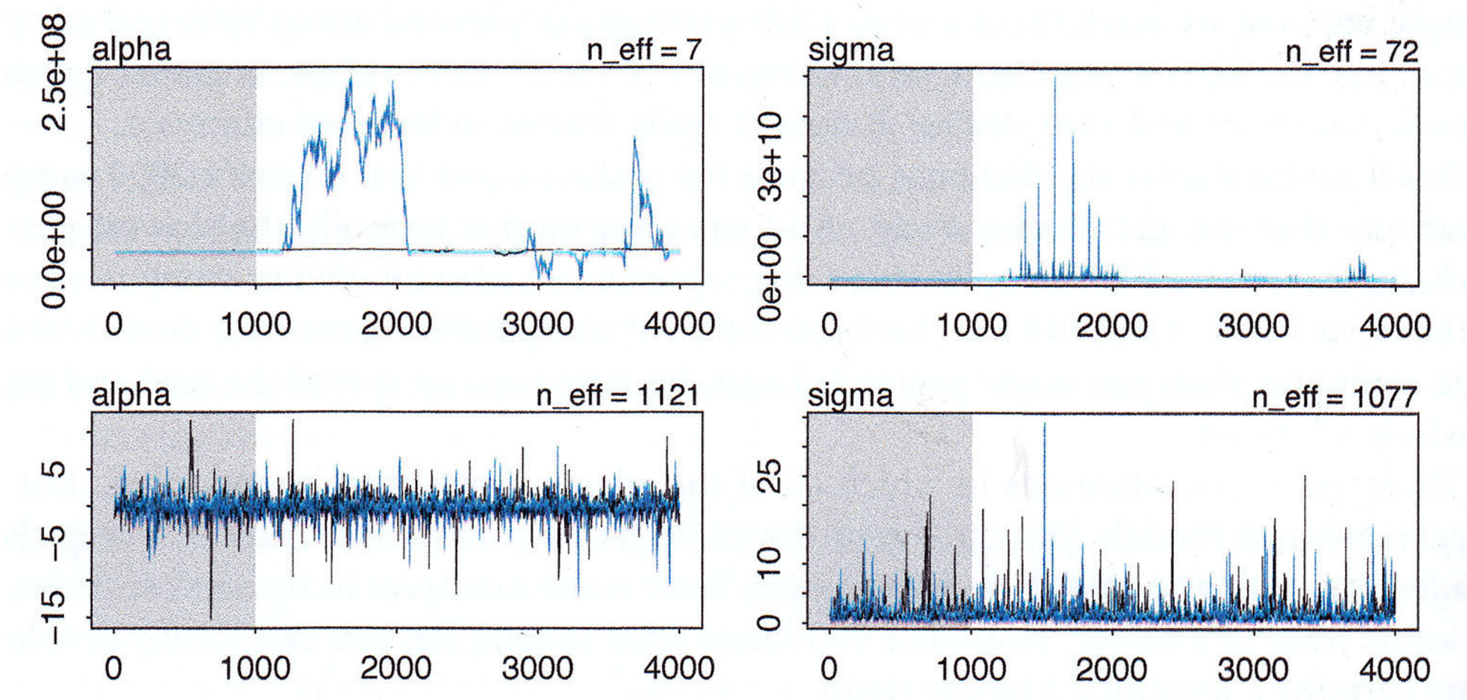


Fig 8.5

Chains

Not converged Converged

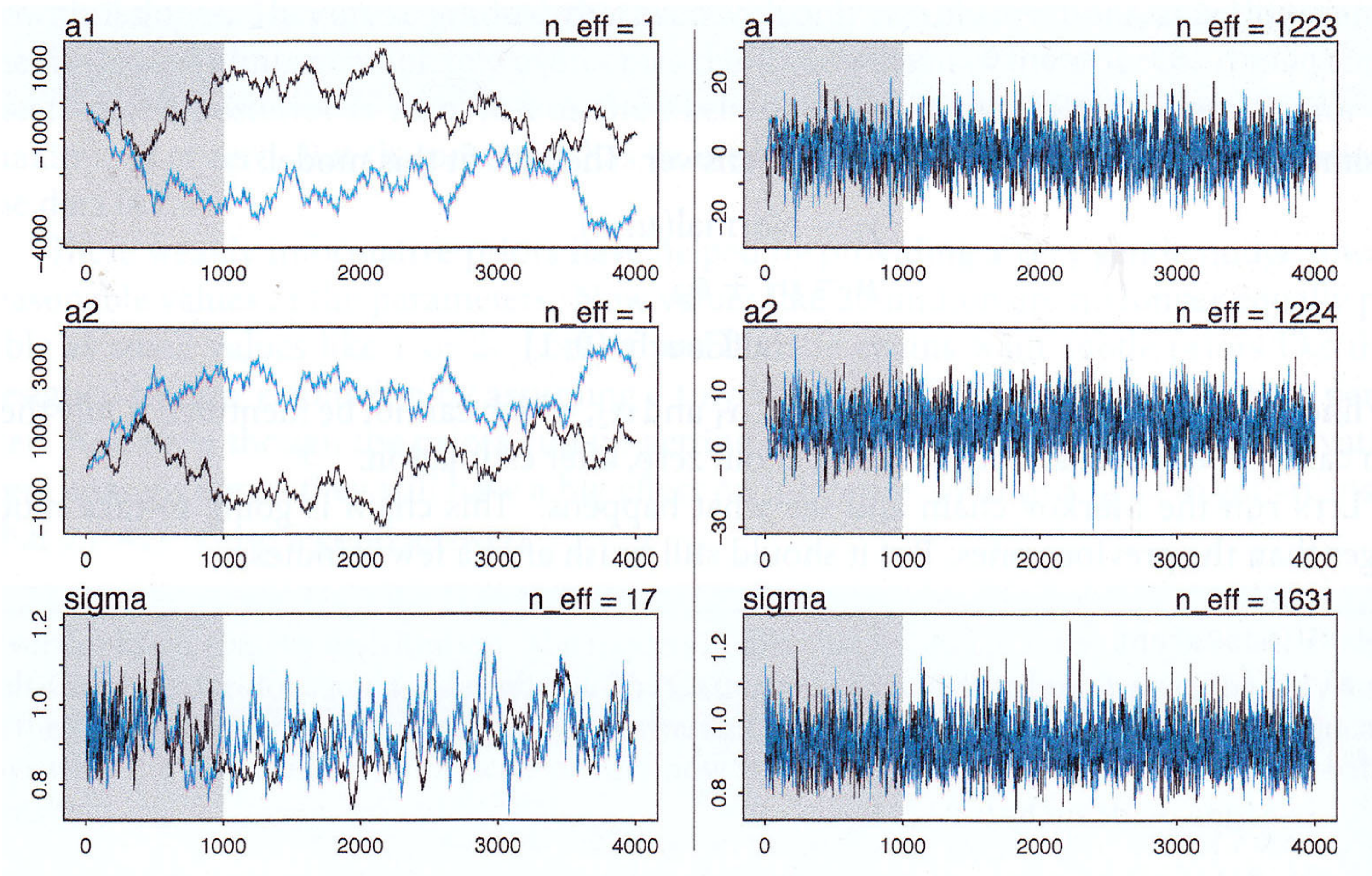


Fig 8.7

How to fix

- Better starting values
- Weakly informative priors
- Uncorrelated parameters (e.g. standardized)
- Less common: adjust MCMC algorithm parameters