36-617: Applied Hierarchical Models

Bayes, Multilevel Models and Stan

Brian Junker
132E Baker Hall
brian@stat.cmu.edu

Announcements

- **<u>Peer Reviews</u>** *(Due Fri 1159pm)*
  - Reviews should be collegial and helpful. Point out things the paper is doing right, and suggestions for improvement.
  - Write in the rubric categories provided, but do not assign points
- **<u>Reading</u>** *(in HW10 & weeks 13 & 14 folders on Canvas)*
  - Lynch, Ch 3 (read), Ch 4 (skim)
  - Lynch, Ch 9 (read)
- **<u>HW10</u>** *(Due Wed Dec 7, 1159pm)*
  - Just some "finger exercises" so you can play with estimating multilevel models with Stan, examining Stan output, etc.
- **<u>Last Quiz</u>** *(Mon-Tue Dec 5-6)*
  - Like midsemester survey – your thoughts about the class.

Outline

- Bayes
  - When we can recognize the posterior
  - When we can't recognize the posterior
  - Monte Carlo, MCMC, and STAN
- *Example 1*: Minnesota Radon – Intercept Only
  - What's new?
  - What is $\hat{R}$?
  - What is n_{eff}?
- *Example 2*: Mn Radon: Level 1 predictor "floor", Level 2 predictor "log(uranium)"
- *Example 3*: CD4 levels in HIV-positive youth

Bayes

- The Slogan
  - (posterior)$\propto$(likelihood)$\times$(prior)
  - (posterior)$\propto$(level 1)$\times$(level 2)

$$f(\theta|data) = \frac{f(data|\theta)f(\theta)}{\int f(data|t)f(t)dt}$$

- Inferences based on features of $f(\theta|data)$, e.g.
  - $\mu_{\theta,post} = \int \theta f(\theta|data)d\theta$
  - $\hat{\theta}_{post} = argmax_{\theta} f(\theta|data)$
  - $\sigma^2_{\theta,post} = \int \left(\theta - \mu_{\theta,post}\right)^2 f(\theta|data)d\theta$
  - E.g., ≈95% CI is $(\mu_{\theta,post} - 2\sigma_{\theta,post}, \mu_{\theta,post} + 2\sigma_{\theta,post})$
- We saw some specific examples last time…

Hierarchical Beta-Binomial model

- Likelihood is binomial:

$$f(x|n,p) = \binom{n}{x} p^x (1-p)^{n-x}$$

- Level 1: x ~ Binom(x|n,p)

- Prior is beta distribution:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

- Level 2: p ~ Beta(p|α,β)

- (posterior) $\propto$ (likelihood)$\times$(prior)

- (posterior) $\propto$ (level 1)$\times$(level 2) = Beta(p|α+x,β+n-x)

Hierarchical Normal-Normal model

- Likelihood (for mean) is normal:

$$f(x_1, \ldots, x_n|\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2}$$

- Prior is normal (for mean):

$$f(\mu) = \frac{1}{\sqrt{2\pi}\tau_0} e^{-\frac{1}{2\tau_0^2}(\mu-\mu_0)^2}$$

- (posterior) $\propto$ (likelihood)$\times$(prior)

- Level 1:

$$x_1, x_2, \ldots, x_n \overset{iid}{\sim} N(\mu, \sigma^2)$$

- Level 2:

$$\mu \sim N(\mu_0, \tau_0^2)$$

- (posterior)$\propto$ (level 1)$\times$(level 2)

$$\mu \sim N(\mu_n, \tau_n^2)$$

When we can recognize the posterior…

- Hierarchical Beta-Binomial
  - $f(p|data) = Beta(p|\alpha', \beta'), \alpha' = \alpha + x, \beta' = \beta + n - x$
  - $\mu_{\theta,post} = \frac{\alpha\prime}{\alpha\prime+\beta\prime} = \frac{\alpha+x}{\alpha+\beta+n}$
  - $\sigma^2_{\theta,post} = \frac{\alpha\prime\beta\prime}{(\alpha'+\beta')^2(\alpha\prime+\beta\prime+1)} = \frac{(\alpha+x)(\beta+n-x)}{(\alpha+\beta+n)^2(\alpha+\beta+n+1)}$
- Hierarchical Normal-Normal
  - $f(\mu|data) = N\left(\mu\middle|\mu_n, \tau_n^2\right)$
  - $\mu_n = \frac{\tau_0^2}{\tau_0^2+\sigma^2/n}\,\bar{y} + \frac{\sigma^2/n}{\tau_0^2+\sigma^2/n}\mu_0$
  - $\tau_n^2 = \frac{1}{n/\sigma^2+1/\tau_0^2}$

When we can't recognize the posterior…

- We still need a way to calculate (or approximate) things like
  - Posterior mean $\mu_{\theta,post} = \int \theta f(\theta|data)d\theta$
  - Posterior mode $\hat{\theta}_{post} = argmax_{\theta} f(\theta|data)$
  - Posterior Variance $\sigma^2_{\theta,post} = \int \left(\theta - \mu_{\theta,post}\right)^2 f(\theta|data)d\theta$
  - Posterior quantile $\theta_{q,post}\ s.t. P\left[\theta \leq \theta_{q,post}\middle|data\right] = q$
    - (e.g. 2.5th %tile, 25th %tile, median, 75th %tile, 97.5th %tile)

- There are a lot of numerical methods to do this
  - Midpoint/trapezoid/Simpson rules, Gaussian quadrature, Laplace's method, *Monte Carlo Integration*, etc., etc. etc.

Monte Carlo Integration

- Suppose $f(\theta)$ is a density, and we want
$$\int g(\theta) f(\theta) d\theta$$
- We know
  - $\int g(\theta) f(\theta) d\theta = E[\mathrm{g}(\Theta)]$, where $\Theta \sim f(\theta)$
  - If we have an iid sample $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ from $f(\theta)$, then by the Law of Large Numbers
$$\overline{g(\theta)} = \frac{1}{M} \sum_{m=1}^{M} g\left(\theta^{(m)}\right) \approx E[g(\Theta)]$$
- By the CLT, a CI for $E[g(\Theta)]$ is approximately
$$\left(\overline{g(\theta)} - 2 \cdot SD_{g(\theta)}/\sqrt{M},\ \ \overline{g(\theta)} + 2 \cdot SD_{g(\theta)}/\sqrt{M}\right)$$

Problem: What if there are many $\theta's$?

- If $\theta \in \Re^1$ there are many good ways to sample from $f(\theta)$
- For our multilevel models,
$$f(\theta) = f(\alpha's, \beta's, \tau^{2\prime}s, \sigma^2 | data)$$
- Even for a "simple" problem like the random intercept model for the Mn Radon data,
$$f(\theta) = f(\alpha_1, \dots, \alpha_{85}, \beta_0, \tau^2, \sigma^2 | data)$$
this is 88 parameters: $\theta \in \Re^{88}$!
- How can we sample from such a high-dimensional density??

Solution: Markov-Chain Monte Carlo (MCMC)

- MCMC is very useful for multivariate distributions, e.g. f(θ_1,θ_2, …,θ_K)
- Naive MCMC: Instead of dreaming up a way to make a draw (simulation) of all K variables at once MCMC takes draws one variable at a time
- We "pay" for this by not getting independent draws. The draws are the states of a *Markov Chain.*
- The draws will not be "exactly right" right away; the Markov chain has to "burn in" or "warm up" to a stationary distribution; the draws after the "burn-in" or "warm up" segment are what we want!

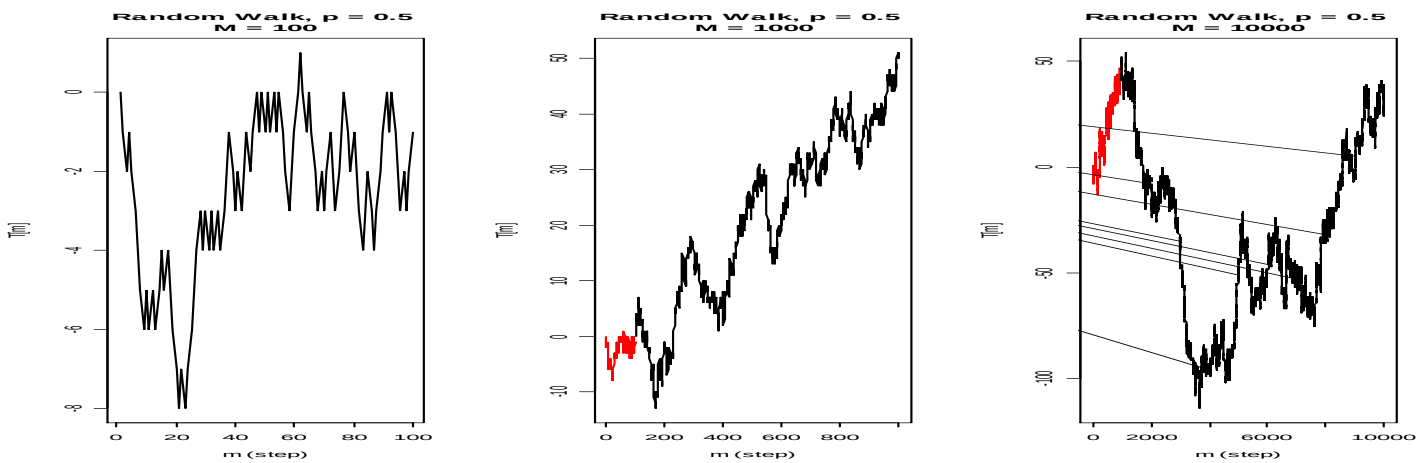
(Digression: What is a Markov Chain?)

- A *Markov Chain* is a stochastic process, i.e. it is a sequence of random variables T_1, T_2, T_3, T_4, T_5, …
- The thing that makes it a Markov Chain is the *Markov Property*:
  - T_{m+1} is independent of T_1, … T_{m-1}, given T_m
  - "the future is independent of the past, given the present"
- A *stationary* Markov Chain has a transition probability function $f(t_m | t_{m-1})$…
  - If the T's are discrete rv's, can write $f(t_m | t_{m-1})$ in terms of a matrix of probabilities
  - If the T's are continuous rv's, $f(t_m | t_{m-1})$ is just a conditional density

(Digression: What is a Markov Chain? …An Example)

- Random Walk
  - T_0 = initial state or "starting point", e.g. 0
  - The transition probability is

$$p(T_m = t_m | T_{m-1} = t_{m-1}) = \begin{cases} p, & \text{if } t_m = t_{m-1} + 1 \\ 1 - p, & \text{if } t_m = t_{m-1} - 1 \\ 0 & \text{else} \end{cases}$$



Back to MCMC: The Gibbs Sampler

- We want to simulate draws from f(θ_1, …,θ_K).
  - Let $T_m = (\theta_1^{(m)}, \theta_2^{(m)}, \ldots, \theta_K^{(m)})$ be a reasonable initial state
  - Now successively sample[1] each θ_k from its *"complete conditional"* distribution:

$$
\begin{aligned}
\theta_1^{(m+1)} &\sim f(\theta_1 | \theta_2^{(m)}, \theta_3^{(m)}, \ldots, \theta_K^{(m)}) \\
\theta_2^{(m+1)} &\sim f(\theta_2 | \theta_1^{(m+1)}, \theta_3^{(m)}, \ldots, \theta_K^{(m)}) \\
\theta_3^{(m+1)} &\sim f(\theta_3 | \theta_1^{(m+1)}, \theta_2^{(m+1)}, \theta_4^{(m)}, \ldots, \theta_K^{(m)}) \\
\vdots \quad & \vdots \qquad\qquad \vdots \\
\theta_K^{(m+1)} &\sim f(\theta_K | \theta_2^{(m+1)}, \theta_3^{(m+1)}, \ldots, \theta_{K-1}^{(m+1)})
\end{aligned}
$$

    and let $T_{m+1} = (\theta_1^{(m+1)}, \theta_2^{(m+1)}, \ldots, \theta_K^{(m+1)})$
  - After "burn-in" B, T_{B+1}, T_{B+2}, …, T_M are MCMC draws "from f"

[1] A kind of stochastic Gauss-Seidel algorithm

MCMC generalities…

- The theory of MCMC (e.g. Chib & Greenberg, *American Statistician*, 1995, pp. 327-335) tells us that
  - $T_m = (\theta_1^{(m)}, \theta_2^{(m)}, \ldots, \theta_K^{(m)})$ is a stationary Markov Chain
  - T_m has stationary distribution f(θ_1, …., θ_K)
- So, if we sample M steps, and throw away the first few, the remaining T_m's can be treated like a sample from f(θ_1, …., θ_K)
  - Not an iid sample though! $\sqrt{M}$-law may not apply!
- *Pretty easy to build adequate MCMC sampler when K is small and posterior well-behaved.*

STAN: A software add-on to R…

- Working out the complete conditionals (CC's) & sampling from them is easy but mortally inefficient for large parameter spaces
- STAN[1] sidesteps the problem:
  - Works out posterior distribution from your spec
  - Uses a modern version of MCMC called *Hamiltonian Monte Carlo* (No U-Turn Sampler - NUTS) to provide highly efficient, nearly-iid samples from posterior
  - Writes & compiles code in C++ to increase speed
- library(bayesplot) for diagnostic tests & plots
  - Also links to additional documentation/tutorials

[1]http://mc-stan.org/users/documentation/

Predecessors to STAN…

- BUGS[1] and JAGS[2] automate MCMC
  - Describe "slogan" in R-like language
  - BUGS figures out complete conditionals & runs parameter-at-a-time Metropolis-Hastings MCMC for you
- STAN[3] implements a faster MCMC method for models with continuous parameters
  - Uses a BUGS-like language
  - Requires more preliminary declarations
  - Usually faster than BUGS/JAGS, often by 10x or more…

1. Bayesian inference Using Gibbs Sampling
2. Just Another Gibbs Sampler
3. In honor of Stanislaw Ulam (1909–1984) – Monte Carlo & Hydrogen Bomb…

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

$$\textbf{Level 1:} \quad y_i \sim N(\alpha_{0j[i]}, \sigma^2)$$

$$\textbf{Level 2:} \quad \alpha_{0j} \sim N(\beta_0, \tau^2)$$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS ON "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

Level 1: $y_i \sim N(\alpha_{0j[i]}, \sigma^2)$

Level 2: $\alpha_{0j} \sim N(\beta_0, \tau^2)$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS ON "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

Level 1: $y_i \sim N(\alpha_{0j[i]}, \sigma^2)$

Level 2: $\alpha_{0j} \sim N(\beta_0, \tau^2)$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS OG "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

Level 1: $y_i \sim N(\alpha_{0j[i]}, \sigma^2)$

Level 2: $\alpha_{0j} \sim N(\beta_0, \tau^2)$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS ON "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

Level 1: $y_i \sim N(\alpha_{0j[i]}, \sigma^2)$

Level 2: $\alpha_{0j} \sim N(\beta_0, \tau^2)$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS ON "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Multilevel Models in STAN

- MLM form

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \qquad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \qquad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical form

$$\textbf{Level 1:} \quad y_i \sim N(\alpha_{0j[i]}, \sigma^2)$$

$$\textbf{Level 2:} \quad \alpha_{0j} \sim N(\beta_0, \tau^2)$$

- STAN form

```
model {

  // LEVEL 1
  for (i in 1:N) {
    log_radon[i] ~
          normal(a0[county[i]],sigma);
  }

  // LEVEL 2
  for (j in 1:J) {
    a0[j] ~ normal(b0,tau0);
  }

  // PRIORS ON "FREE PARAMETERS"
  b0 ~ normal(0,1e+6);
  sigma ~ uniform(0,50);
  tau0 ~ uniform(0,50);

}
```

Have to add priors to all free parameters

Example 1: Minnesota Radon – Intercept Only

- MLM:

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \epsilon_i, \\
& \quad \epsilon_i \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_0 + \eta_j, \\
& \quad \eta_j \sim N(0, \tau^2)
\end{aligned}
$$

- Hierarchical:

$$\textbf{Level 1:}\ y_i \sim N(\alpha_{0j[i]}, \sigma^2)$$

$$\textbf{Level 2:}\ \alpha_{0j} \sim N(\beta_0, \tau^2)$$

- Demonstration in R and STAN…
- (comparison with lmer also)

Review…

- `print(stanfit.object)` and `summary(stanfit.object)`: point estimates and CI's for parameters.
- MCMC samples themselves available via `extract(stanfit.object)`
- Other estimation, plotting and diagnostic functions (see `library(help=rstan)`)
- `Library(bayesplot)` and `library(shinystan)`: graphical estimation and diagnostic tools

What's new?

- STAN automatically
  - Runs 4 separate MCMC chains of 2000 steps each
    - Number of steps for each specified with "iter=" in stan() function
  - Throws away the first half of each chain as "burn-in/warm-up"
  - *You can change these when you run stan(); see help(stan)*
  - *You can also set initial values for the chains; again help(stan)*
- STAN reports
  - an "Rhat" statistic for each parameter estimated
  - an "neff" statistic for each parameter (effective sample size)
- We'll look at their definitions on the next page
  - For STAN, Rhat usually quite close to the "ideal" value of 1.00

What is $\hat{R}$?

- Suppose we have M chains:

Chains				Means	Variances
$\theta^{(1;1)}$,	$\theta^{(1;2)}$,	$\ldots$,	$\theta^{(1;N)}$	$\overline{\theta}_1$	W_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$\theta^{(M;1)}$,	$\theta^{(M;2)}$,	$\ldots$,	$\theta^{(M;N)}$	$\overline{\theta}_M$	W_M
			Grand mean	$\overline{\theta}$	

- Define

$$
\begin{aligned}
W &= \tfrac{1}{M(N-1)} \textstyle\sum_{m=1}^{M} \sum_{n=1}^{N} (\theta^{(m;n)} - \bar{\theta}_m)^2 \;=\; \tfrac{1}{M} \textstyle\sum_{m=1}^{M} W_m \\
&= \text{Average within-chain variance}
\end{aligned}
$$

$$
\begin{aligned}
B &= \tfrac{M}{M-1} \textstyle\sum_{m=1}^{M} (\bar{\theta}_m - \bar{\theta})^2 \\
&= \text{Between-chain variance, inflated for sample size}
\end{aligned}
$$

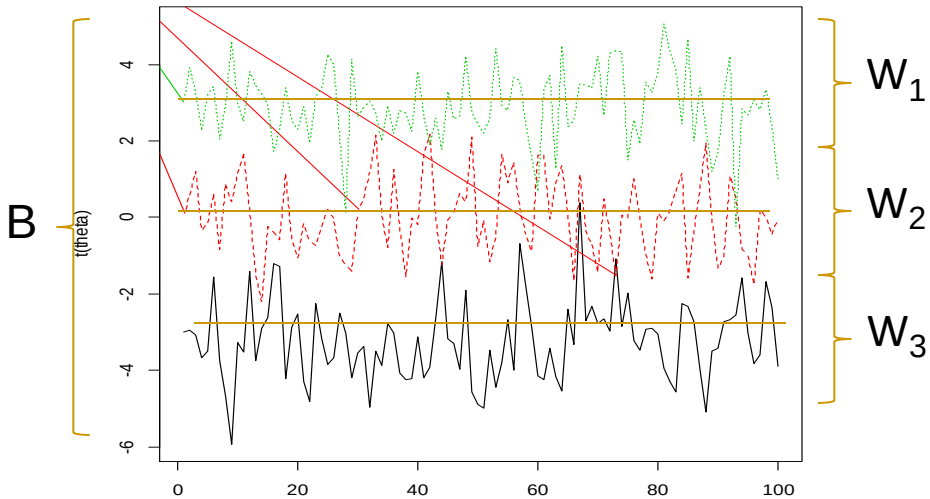
$$
\begin{aligned}
V &= \tfrac{M-1}{M} W + \tfrac{1}{M} B \\
&= \text{Pooled variance estimate,}
\end{aligned}
$$

What is $\hat{R}$?

- Separated chains:

$$
\begin{aligned}
W &= \frac{1}{N}\sum_{n=1}^{N} W_n &&= 1.02 \\
B &= \frac{M}{N-1}\sum_{n=1}^{N}(\bar{\theta}_n - \bar{\theta})^2 &&= 938.53 \\
V &= \frac{M-1}{M}W + \frac{1}{M}B &&= 10.40
\end{aligned}
$$

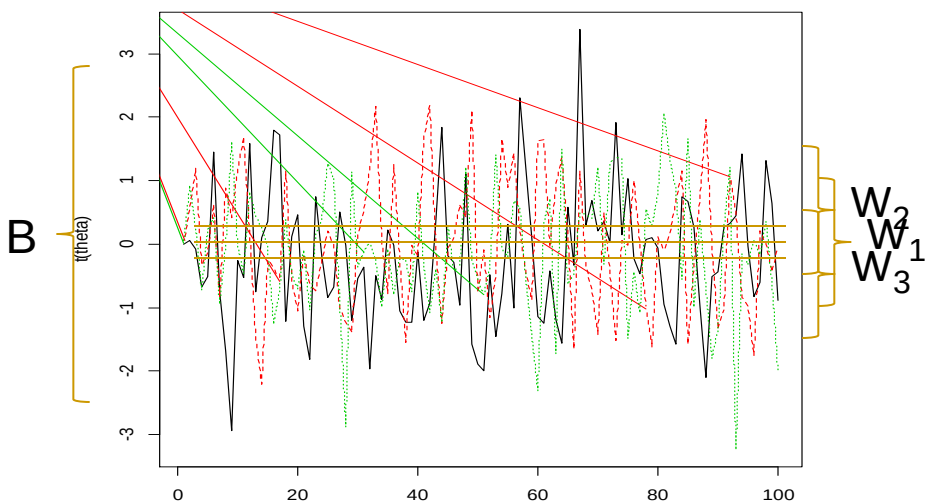
$$\hat{R} = \sqrt{V/W} = 3.19$$



- Converged chains:

$$
\begin{aligned}
W &= \frac{1}{N}\sum_{n=1}^{N} W_n &&= 1.02 \\
B &= \frac{M}{N-1}\sum_{n=1}^{N}(\bar{\theta}_n - \bar{\theta})^2 &&= 1.32 \\
V &= \frac{M-1}{M}W + \frac{1}{M}B &&= 1.03
\end{aligned}
$$

$$\hat{R} = \sqrt{V/W} = 1.00$$



What is n_{eff}?

- Because the Markov Chain draws may have dependence, the usual rule of thumb for a 95% estimation interval from the MC draws

$$(\bar{\theta}_{draws} - \frac{2 \cdot SD(\hat{\theta}_{draws})}{\sqrt{n_{draws}}}, \bar{\theta}_{draws} + \frac{2 \cdot SD(\hat{\theta}_{draws})}{\sqrt{n_{draws}}})$$

  doesn't work.

- One way to deal with this is to calculate what the equivalent (or "effective") sample size would be, if the draws were independent. With this value, n_{eff}, we could get our usual interval,

$$(\bar{\theta}_{draws} - \frac{2 \cdot SD(\hat{\theta}_{draws})}{\sqrt{n_{eff}}}, \bar{\theta}_{draws} + \frac{2 \cdot SD(\hat{\theta}_{draws})}{\sqrt{n_{eff}}})$$

https://en.wikipedia.org/wiki/Effective_sample_size
https://mc-stan.org/docs/2_21/reference-manual/effective-sample-size-section.html

What is n_{eff}?

- If $Var(\theta_k) \equiv \sigma^2 = \sigma^2_{draws}$, and $Cov(\theta_j, \theta_k) \equiv \rho$, then it is easy to figure out the effective sample size: for $n = n_{draws}$ samples of θ, we have

$$
\begin{aligned}
Var(\bar{\theta}) &= Var\left(\frac{1}{n}\sum_{k=1}^{n}\theta_k\right) \\
&= \sum_{k=1}^{n}\frac{1}{n^2}Var(\theta_k) + \sum_{k=1}^{n}\sum_{j=1, j\neq k}^{n}\frac{1}{n^2}Cov(\theta_j, \theta_k) \\
&= n\frac{\sigma^2}{n^2} + n(n-1)\frac{\rho\sigma^2}{n^2} = \sigma^2\frac{1+(n-1)\rho}{n}
\end{aligned}
$$

so $n_{eff} = \frac{n}{1+(n-1)\rho}$, where $n = n_{draws}$.

- If the correlations depend on the lag t between θ_i and θ_{i+t}, then one can calculate that

$$
n_{eff} = \frac{n_{draws}}{1 + 2\sum_{t=1}^{\infty}\rho_t} \approx \frac{n_{draws}}{1 + 2\sum_{t=1}^{t_{\max}}\rho_t}
$$

where $\rho_t \approx 0,\ \forall t \geq t_{\max}$ (usually around 20 or 30 at most, as you can see from the acf plots…).

https://en.wikipedia.org/wiki/Effective_sample_size
https://mc-stan.org/docs/2_21/reference-manual/effective-sample-size-section.html

Rules of thumb for $\hat{R}$ and n_{eff}

- $\hat{R} = 1$ at perfect "convergence" to the Markov Chain's stationary distribution
  - $\hat{R} \leq 1.05$ is ideal
  - $\hat{R} \leq 1.10$ is often acceptable
- n_{eff} is a measure of accuracy but also of how "bad" the correlations ρ_t in the Markov Chain are
  - $n_{eff} \geq 100$ is often "good enough" for estimation
  - $n_{eff} \geq (0.5)n_{draws}$ suggests low ρ_t's
  - $n_{eff} \geq (0.1)n_{draws}$ suggests acceptable ρ_t's
  - $n_{eff} < (0.1)n_{draws}$ suggests worrisome ρ_t's

Example 2: Mn Radon: Level 1 predictor "floor", Level 2 predictor "log(uranium)"

- MLM:

$$
\begin{aligned}
y_i &= \alpha_{0j[i]} + \alpha_{1j[i]}(\text{floor})_i + \epsilon_{ij[i]},\ \epsilon_{ij} \sim N(0, \sigma^2) \\
\alpha_{0j} &= \beta_{00} + \beta_{01}\log(\text{uranium}_j) + \eta_{0j},\ \eta_{0j} \sim N(0, \tau_0^2) \\
\alpha_{1j} &= \beta_{10} + \eta_{1j},\ \eta_{1j} \sim N(0, \tau_1^2)
\end{aligned}
$$

- Hierarchical:

$$\textbf{Level 1:}\ y_i \sim N(\alpha_{0j[i]} + \alpha_{1j[i]}(\text{floor})_i, \sigma^2)$$

$$\textbf{Level 2:}\ \alpha_{0j} \sim N(\beta_{00} + \beta_{01}\log(\text{uranium}_j), \tau_0^2)$$

$$\alpha_{1j} \sim N(\beta_{10}, \tau_1^2)$$

- Demonstration in R and STAN…

Example 3: CD4 in HIV-positive youth

- See R handout, and demonstration in class

Wrap-Up…

- STAN automates MCMC
  - Specify (posterior)$\propto$(level 1)$\times$(level 2)$\times$…
    in an R-like language
  - STAN designs and runs the MCMC for you
  - Gelman & Hill use BUGS, we will use STAN
- Summaries of parameter estimates, and good graphs: rstan helper functions, basyesplot & shinystan…
  - Rhat $\leq$ 1.05 is a handy "convergence diagnostic"
  - $n_{eff} \geq (0.5)n_{draws}$ suggests nice low values for ρ_t's
  - Use n_{eff} rather than n_{draws} for "back of envelope" CI's

Summary

- Bayes
  - When we can recognize the posterior
  - When we can't recognize the posterior
  - Monte Carlo, MCMC, and STAN
- *Example 1*: Minnesota Radon – Intercept Only
  - What's new?
  - What is $\hat{R}$?
  - What is n_{eff}?
- *Example 2*: Mn Radon: Level 1 predictor "floor", Level 2 predictor "log(uranium)"
- *Example 3*: CD4 levels in HIV-positive youth