Chapter 8 Questions

Melissa Van Bussel

July 5, 2018

Chapter 8 points: 2 + 6 + 7 = 15.

Easy (2 points total)

8E1

The proposal distribution must be symmetric.

8E2

Gibbs sampling uses conditional distributions whereas Metropolis does not. We can't use Gibbs sampling in situations where we can't use conditional distributions, plus sometimes the estimate can get stuck in a small part of the posterior (won't jump around enough and converge to the correct value) if there are correlated parameters.

8E3

HMC can only deal with continuous parameters (not discrete), because it needs a continuous surface while sampling.

8E4

The n_eff is the number of effective samples. It will always be smaller than the number of iterations, but the higher it is, the better.

8E5

Rhat should approach 1 if the chain is converging.

8E6

A good traceplot should look like a "fuzzy caterpillar". A bad traceplot will look like a bunch of mountains going up and down and not all in the same place (it'll look very erratic). It could also look flat if it were a bad traceplot.

Medium (6 points total)

8M1 (2 points)

```
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Warning: package 'rstan' was built under R version 3.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.3.3

## Loading required package: StanHeaders

## Warning: package 'StanHeaders' was built under R version 3.3.3

## rstan (Version 2.17.3, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

## Loading required package: parallel

## rethinking (Version 1.59)
```

```
data(rugged)
d <- rugged
d$log_gdp <- log(d$rgdppc_2000)
dd <- d[ complete.cases(d$rgdppc_2000) , ]
dd.trim <- dd[ , c("log_gdp","rugged","cont_africa") ]

lmod8m1_unif <- map2stan(
  alist(
    log_gdp ~ dnorm(mu,sigma),
    mu <- a + bR*rugged + bA*cont_africa + bAR*rugged*cont_africa,
    a ~ dnorm(0,100),
    c(bR, bA, bAR) ~ dnorm(0,10),
    sigma ~ dunif(0,10)
),
data = dd.trim)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions
```

```
lmod8m1_exp <- map2stan(
    alist(
      log_gdp ~ dnorm(mu,sigma),
      mu <- a + bR*rugged + bA*cont_africa + bAR*rugged*cont_africa,
      a ~ dnorm(0,100),
      bR ~ dnorm(0,10),
      bA ~ dnorm(0,10),
      bAR ~ dnorm(0,10),
      sigma ~ dexp(1)
),
data = dd.trim)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

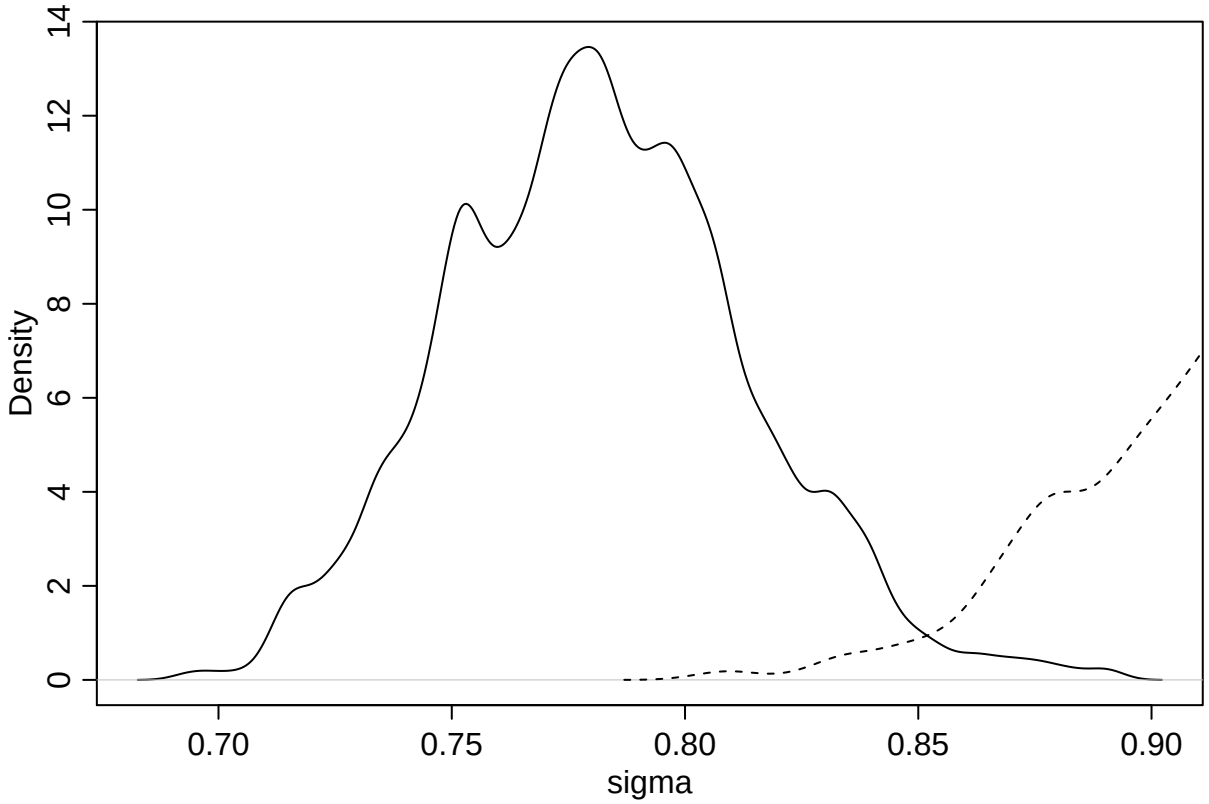
```
## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions
```

8M2 (2 points)

```
# a model which will have an effect on the posterior distribution of sigma

lmod8m2_exp_new2 <- map2stan(
  alist(
    log_gdp ~ dnorm(mu,sigma),
    mu <- a + bR*rugged + bA*cont_africa + bAR*rugged*cont_africa,
    a ~ dnorm(0,100),
    bR ~ dnorm(0,10),
    bA ~ dnorm(0,10),
    bAR ~ dnorm(0,10),
    sigma ~ dexp(100)
),
data = dd.trim)
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions
```

```
sigma_old <- extract.samples(lmod8m1_exp, pars = "sigma")
sigma_new2 <- extract.samples(lmod8m2_exp_new2, pars="sigma")
dens(sigma_new2[[1]], xlab = "sigma")
dens(sigma_old[[1]] , add=TRUE , lty=2 )
```



You have to make extreme changes in order to see change in the posterior distribution since the Cauchy distribution has such a large tail.

8M3 (2 points)

We want to re-estimate one of the models from earlier in the chapter, but ramp up the number of warmup iterations. Usually, 1000 warmup iterations is pretty standard for a good chain.

```
warmup_values <- c(1,5,10,100,500,1000)

# first make matrix to hold n_eff results
n_eff <- matrix(NA , nrow = length(warmup_values), ncol = 5)

# lets use the model we used in the previous question
for (i in 1:length(warmup_values)) {
  new_map2stan <- resample(1mod8m2_exp_new2, warmup = 1000 + warmup_values[i])
  n_eff[i,] <- precis(new_map2stan)@output$n_eff
}
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning in map2stan(object, ...): 'iter' less than or equal to 'warmup'.
## Setting 'iter' to sum of 'iter' and 'warmup' instead (4000).

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions
```

```
n_eff
```

We can see that the more we increase warmup, the better the n_eff becomes, obviously. It usually comes down to a matter of time.

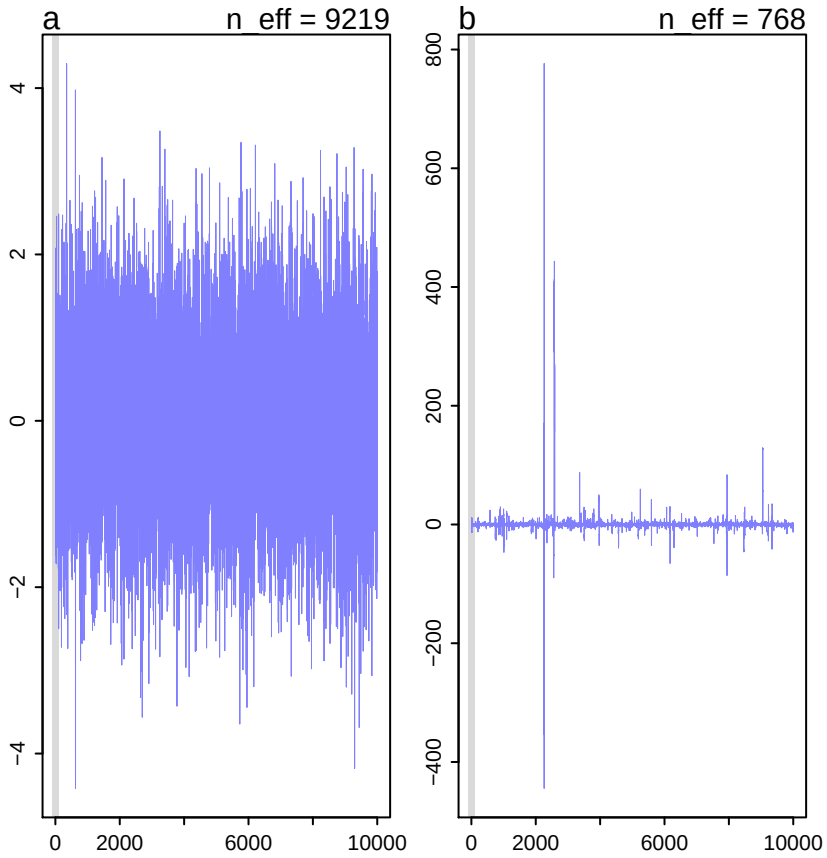
Hard (7 points total)

8H1 (1 point)

```
mp <- map2stan(
  alist(
    a ~ dnorm(0,1),
    b ~ dcauchy(0,1)),
  data = list(y = 1),
  start = list(a = 0, b = 0), iter = 1e4, warmup = 100, WAIC = FALSE)
```

```
## Warning: There were 2 transitions after warmup that exceeded the maximum treedepth. Increase max_tree
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
precis(mp)
plot(mp)
```



We can see that the n_eff for the first coefficent is much larger than the second. Also, on the traceplots, the second one looks a bit wonky, but that's because it's a Cauchy and this is how they behave.

8H2 (2 points)

We want to use the **compare()** function on the 3 models to see which one is the best in terms of the lowest WAIC. We just copy the code right out of the textbook for the model specifications:

```
library(rethinking)
data(WaffleDivorce)
d <- WaffleDivorce
d$MedianAgeMarriage_s <- (d$MedianAgeMarriage-mean(d$MedianAgeMarriage))/
sd(d$MedianAgeMarriage)
d$Marriage_s <- (d$Marriage - mean(d$Marriage))/sd(d$Marriage)

m5.1_stan <- map2stan(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <- a + bA * MedianAgeMarriage_s,
```

```
    a ~ dnorm(10, 10),
    bA ~ dnorm(0, 1),
    sigma ~ dunif(0, 1)
),
  data = d, chains = 4)
```

```
## Warning: Variable 'Marriage.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Warning: Variable 'Divorce.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
m5.2_stan <- map2stan(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <- a + bR * Marriage_s,
    a ~ dnorm(10, 10),
    bR ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
),
data = d, chains = 4)
```

```
## Warning: Variable 'Marriage.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Warning: Variable 'Divorce.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
m5.3_stan <- map2stan(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <- a + bR*Marriage_s + bA*MedianAgeMarriage_s,
    a ~ dnorm(10, 10 ),
    bR ~ dnorm(0, 1 ),
    bA ~ dnorm(0, 1 ),
    sigma ~ dunif(0, 10)
),
data = d, chains = 4)
```

```
## Warning: Variable 'Marriage.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Warning: Variable 'Divorce.SE' contains dots '.'.
## Will attempt to remove dots internally.
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions
```

```
compare(m5.1_stan, m5.2_stan, m5.3_stan)
```

We can see that the WAICs of the top 2 models are extremely similar (almost a tie).

8H3 (1 point)

```
N <- 100
height <- rnorm(N, 10, 2)
leg_prop <- runif(N, 0.4, 0.5)
leg_left <- leg_prop*height +
  rnorm(N, 0, 0.02)
leg_right <- leg_prop*height +
  rnorm(N, 0, 0.02)
d <- data.frame(height,leg_left,leg_right)

m5.8s <- map2stan(
  alist(
    height ~ dnorm(mu, sigma),
    mu <- a + bl*leg_left + br*leg_right,
    a ~ dnorm(10, 100),
    bl ~ dnorm(2, 10),
    br ~ dnorm(2, 10),
    sigma ~ dcauchy(0, 1)),
  data = d, chains = 4, start = list(a = 10, bl = 0, br = 0, sigma = 1))
```

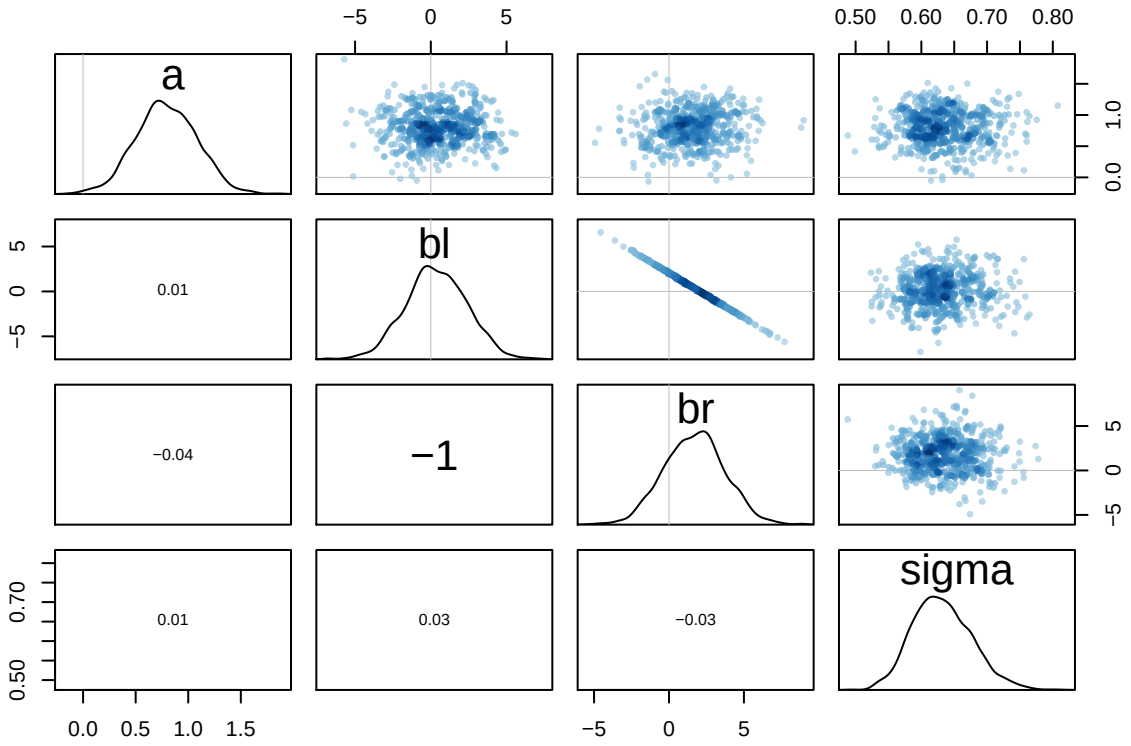
```
## Warning: There were 805 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC

## Constructing posterior predictions
```

```
m5.8s2 <- map2stan(
  alist(height ~ dnorm(mu, sigma),
        mu <- a + bl*leg_left + br*leg_right,
        a ~ dnorm(10, 100),
        bl ~ dnorm(2, 10),
        br ~ dnorm(2, 10) & T[0,],
        sigma ~ dcauchy(0, 1)),
  data = d, chains = 4, start = list(a = 10, bl = 0, br = 0, sigma = 1))
```

```
## Warning: There were 1216 divergent transitions after warmup. Increasing adapt_delta above 0.8 may he
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: There were 94 transitions after warmup that exceeded the maximum treedepth. Increase max_tr
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

## Computing WAIC
## Constructing posterior predictions

## Warning in map2stan(alist(height ~ dnorm(mu, sigma), mu <- a + bl * leg_left + : There were 1216 div
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```

```
pairs(m5.8s)
```



8H4 (2 points)

```
compare(m5.8s, m5.8s2)
```

We can see that the two models are pretty much tied!

8H5 (1 point)

```
pop_size <- sample(1:10)
num_weeks <- 1e5
positions <- rep(0, num_weeks)
pop_size <- sample(1:10)
current <- 10
for (i in 1:num_weeks) {
  positions[i] <- current
  proposal <- current + sample( c(-1,1) , size=1 )
  if (proposal < 1)
    proposal <- 10
  if (proposal > 10)
    proposal <- 1
prob_move <- pop_size[proposal] / pop_size[current]
current <- ifelse(runif(1) < prob_move, proposal, current)
}

f <- table(positions)

plot(as.vector(f), pop_size, type = "n",
  xlab = "frequency", ylab = "population size")
text(x = f, y = pop_size)
```