

# Homework 3

Olivia Hackworth

4/1/2019

```
library(tidyverse)
library(MASS)
```

In the evaluation of drugs for possible clinical application, studies are routinely performed on rodents. For a particular study drawn from the statistical literature, suppose the immediate aim is to estimate  $\theta$ , the probability of tumor in a population of female laboratory rats of type F344 that receive a zero dose of the drug (a control group). The data show that 4 out of 14 rats developed endometrial stromal polyps (a kind of tumor). It is natural to assume a binomial model for the number of tumors given  $\theta_{68}$ . For convenience, we select a prior distribution for  $\theta_{68}$  from the conjugate family,  $\theta \sim \text{Beta}(\alpha, \beta)$ .

Typically, the mean and standard deviation of underlying tumor risks are not available. Rather, historical data are available on previous experiments on similar groups of rats. In the rat tumor example, the historical data were in fact a set of observations of tumor incidence in 67 groups of rats. In the  $j$ -th historical experiment, let the number of rats with tumors be  $y_j$  and the total number of rats be  $n_j$ . We model the  $y_j$ 's as independent binomial data, given sample sizes  $n_j$  and study-specific means  $\theta_j$ . Assuming that the beta prior distribution with parameters  $\alpha, \beta$  is a good description of the population distribution of the  $\theta_j$ 's in the historical experiments, we can display the hierarchical model schematically as in Figure 1, with  $\theta_{68}$  and  $y_{68}$  corresponding to the current experiment.

1. Write down the joint posterior distribution of  $(\theta_1, \dots, \theta_{68})$  given  $(\alpha, \beta)$ . Visualize the posterior distributions of  $(\theta_1, \dots, \theta_{68})$  when  $(\alpha, \beta) = (1.4, 8.6)$ .
2. What is the expectation of each  $y_j$  given  $\alpha, \beta$ ? What is the expectation of each  $y_j^2$  given  $\alpha, \beta$ ? Can you give a possible value of  $(\alpha, \beta)$  based on summaries of these moments? Plug in your guess for  $(\alpha, \beta)$ , calculate the posterior distribution for  $(\theta_1, \dots, \theta_{68})$  and compare with the results in 1.
3. Show that a uniform prior on  $(\frac{\alpha}{\alpha+\beta}, (\alpha+\beta)^{-1/2})$  yields a prior on the original scale  $p(\alpha, \beta) \propto (\alpha+\beta)^{-5/2}$  and on natural transformed scale  $p(\log \frac{\alpha}{\beta}, \log(\alpha+\beta)) \propto \alpha\beta(\alpha+\beta)^{-5/2}$ .
4. Write down the joint posterior distribution of  $(\theta_1, \dots, \theta_{68}, \alpha, \beta)$ . Write down the marginal posterior distribution of  $(\alpha, \beta)$  on the original scale  $p(\alpha, \beta|y)$  and on the log transformed scale  $p(\log \frac{\alpha}{\beta}, \log(\alpha+\beta)|y)$ .
5. Work on either  $p(\alpha, \beta|y)$  or  $p(\log \frac{\alpha}{\beta}, \log(\alpha+\beta)|y)$ , design a Metropolis-Hastings algorithm to obtain posterior samples for  $\alpha, \beta$ . Visualize your results.
6. Given the posterior samples of  $\alpha, \beta$  can you obtain posterior samples for  $(\theta_1, \dots, \theta_{68})$ ? Visualize your results.
7. Work with either  $p(\alpha, \beta; \theta_1, \dots, \theta_{68}|y)$  or  $p(\log \frac{\alpha}{\beta}, \log(\alpha+\beta); \theta_1, \dots, \theta_{68}|y)$ , design a Metropolis-Hastings (maybe combined with Gibbs sampling) algorithm to obtain posterior samples for  $(\alpha, \beta; \theta_1, \dots, \theta_{68})$ . Visualize your results.
8. Implement 7 in STAN.
9. Calculate  $E(\alpha|y)$  and  $Pr(\frac{\alpha}{\alpha+\beta} < 0.2|y)$  based on posterior samples in 7.
10. Calculate  $E(\alpha|y)$  and  $Pr(\frac{\alpha}{\alpha+\beta} < 0.2|y)$  based on posterior samples in 8.

data

```
y <- c(0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2,
      2,1,5,2,5,3,2,7,7,3,3,2,9,10,4,4,4,4,4,4,10,4,4,4,5,
      11,12, 5,5,6,5,6,6,6,6,16,15,15,9,4)
```

```
n <- c(20,20,20,20,20,19,19,19,19,18,17,20,20,20,20,19,19,18,18,25,24,
      23,20,20,20,20,20,20,10,49,19,46,27,17,49,47,20,20,13,48,50,20,
      20,20,20,20,20,20,48,19,19,19,22,46,49,20,20,23,19,22,20,20,20,
      52,46,47,24,14)
```

1

From page 110

$$p(\theta|\alpha, \beta, y) = \prod_{j=1}^J \frac{\Gamma(\alpha+\beta+n_j)}{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)} \theta_j^{\alpha+y_j-1} (1-\theta_j)^{\beta+n_j-y_j-1}$$

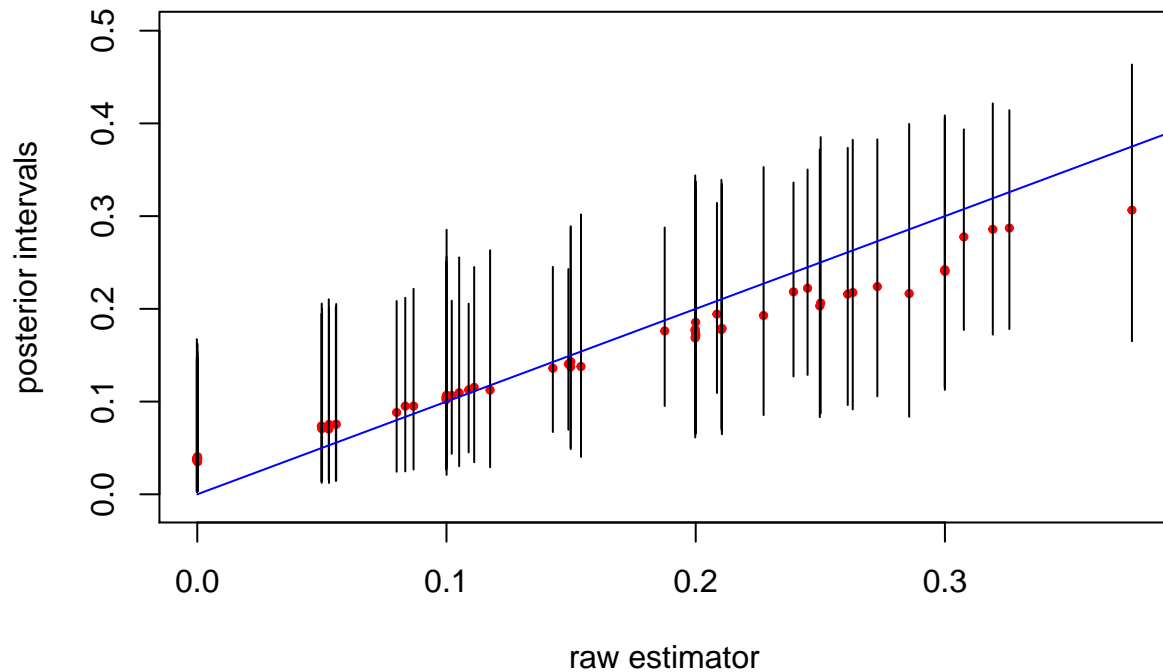
```
alpha = 1.4
beta = 8.6
obs_mat = cbind(y,n)

K = 1000 #number of sims
post_theta_mat = matrix(NA,nrow = K, ncol = length(y)) #rows are sims, cols are thetas
for(i in 1:K){
  #one posterior draw of thetas
  post_theta_draw = apply(obs_mat,1,function(x) rbeta(1,alpha+x[1], beta+x[2]-x[1]))
  post_theta_mat[i,] = post_theta_draw
}
```

recreating figure 5.4 on page 113 Code for plot from heirarchical models code on canvas.

```
postintervals_sample <- apply(post_theta_mat, 2, function(x) c(median(x), c(quantile(x, c(0.025, 0.975)))))

rawest <- jitter(y / n)
plot(rawest, postintervals_sample[1, ], pch = 19, col = 'red', cex = 0.5, ylim = c(-0.01, 0.5), ylab =
for(k in 1:length(y)){
  lines(cbind(rep(rawest[k], 2), postintervals_sample[2:3, k]))
}
lines(seq(0, 0.4, by = 0.01), seq(0, 0.4, by = 0.01), col = 'blue')
```



```
#phatpool = sum(y)/sum(n)
#points(phatpool, phatpool, cex = 2, col = 'green')
#lines(cbind(rep(qbeta(0.5, sum(y)+1, sum(n)-sum(y)+1), 2), c(qbeta(0.025, sum(y)+1, sum(n)-sum(y)+1),
```

## 2

To solve this, I made the assumption that the  $y_j/n_j$  are good estimates of the  $\theta_j$ . This means I ignored some of the variance of the problem and solved equations based on the fact that I assumed  $\frac{y_j}{n_j} \sim \text{Beta}(\alpha, \beta)$ .

The system of equations I worked with is:

$$\frac{1}{68} \sum \frac{y_j}{n_j} = \frac{\alpha}{\alpha + \beta}$$

$$\frac{1}{68} \sum \frac{y_j^2}{n_j^2} = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} + \left(\frac{\alpha}{\alpha + \beta}\right)^2$$

The second equation comes from the fact that  $E[X^2] = V(X) + (E[X])^2$

For notational convenience,  $A = \frac{1}{68} \sum \frac{y_j}{n_j}$  and  $B = \frac{1}{68} \sum \frac{y_j^2}{n_j^2}$ .

```
A = (1/68)*sum(y/n)
A
```

```
## [1] 0.1434731
```

```
B = (1/68)*sum(y^2/n^2)
B
```

```
## [1] 0.03107371
```

I plugged the system of equations and the A and B estimates into the wolfram alpha system of equations solver and the results are that  $\hat{\alpha} = 1.53742$  and  $\hat{\beta} = 9.17833$

```
alpha_est = 1.53742
beta_est = 9.17833
```

Now, I can repeat the sampling procedure and plot from part 1.

```

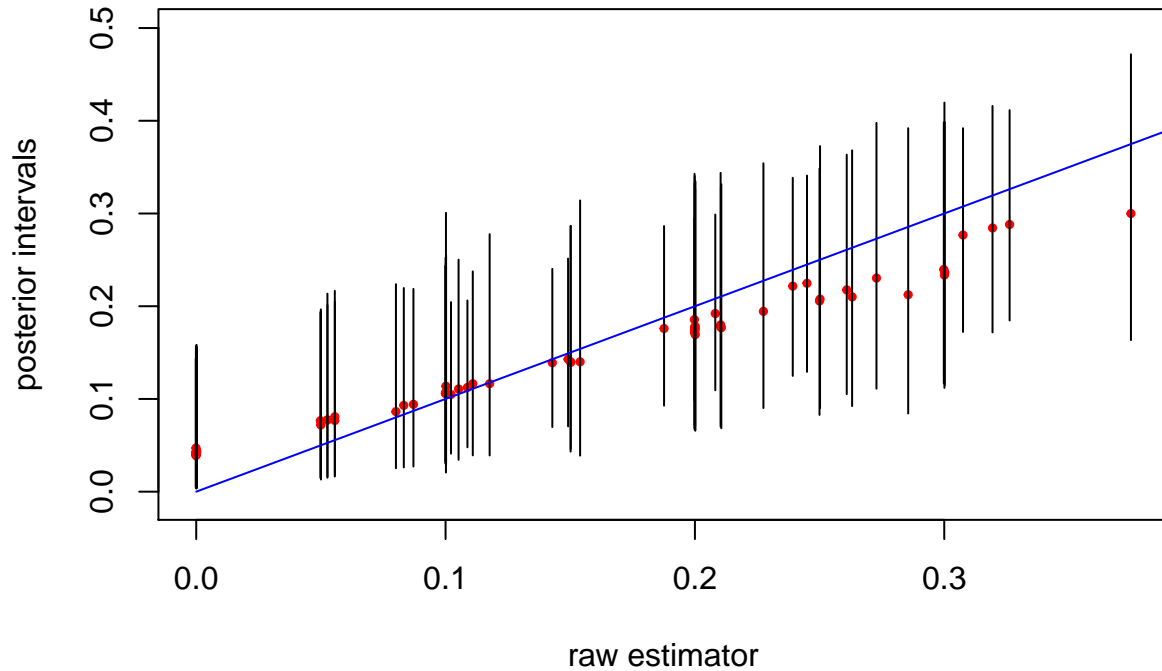
obs_mat2 = cbind(y,n)

K = 1000 #number of sims
post_theta_mat2 = matrix(NA,nrow = K, ncol = length(y)) #rows are sims, cols are thetas
for(i in 1:K){
  #one posterior draw of thetas
  post_theta_draw = apply(obs_mat2,1,function(x) rbeta(1,alpha_est+x[1], beta_est+x[2]-x[1]))
  post_theta_mat2[i,] = post_theta_draw
}

#figure code
postintervals_sample2 <- apply(post_theta_mat2, 2, function(x) c(median(x), c(quantile(x, c(0.025, 0.975)))))

rawest <- jitter(y / n)
plot(rawest, postintervals_sample2[1, ], pch = 19, col = 'red', cex = 0.5, ylim = c(-0.01, 0.5), ylab = "posterior intervals")
for(k in 1:length(y)){
  lines(cbind(rep(rawest[k], 2), postintervals_sample2[2:3, k]))
}
lines(seq(0, 0.4, by = 0.01), seq(0, 0.4, by = 0.01), col = 'blue')

```



The alpha and beta estimates from this part are similar to the estimates given in part one. The two plots are also very similar, which makes sense because of the similarity in alpha and beta.

### 3

First transformation:

$$f = \frac{a}{a+b}, g = (a+b)^{-1/2}$$

$$J = \det \begin{vmatrix} \frac{\delta f}{\delta a} & \frac{\delta f}{\delta b} \\ \frac{\delta g}{\delta a} & \frac{\delta g}{\delta b} \end{vmatrix}$$

$$\begin{aligned}
&= \det \begin{vmatrix} \frac{b}{(a+b)^2} & \frac{-a}{(a+b)^2} \\ \frac{-1}{2(a+b)^{3/2}} & \frac{-1}{2(a+b)^{3/2}} \end{vmatrix} \\
&= \frac{-b}{2(a+b)^{7/2}} - \frac{a}{2(a+b)^{7/2}} \\
&= \frac{-(a+b)}{2(a+b)^{7/2}} \\
&= \frac{-1}{2}(a+b)^{-5/2} \\
&\propto (a+b)^{-5/2}
\end{aligned}$$

So  $p(a, b) \propto 1 * (a+b)^{-5/2}$

Second transformation:

$$c = \log(a/b), d = \log(a+b)$$

To go from c,d to a,b

$$\begin{aligned}
J &= \det \begin{vmatrix} \frac{\delta c}{\delta a} & \frac{\delta c}{\delta b} \\ \frac{\delta d}{\delta a} & \frac{\delta d}{\delta b} \end{vmatrix} \\
&= \det \begin{vmatrix} \frac{1}{a+b} & \frac{-1}{a+b} \\ \frac{1}{a+b} & \frac{1}{a+b} \end{vmatrix} \\
&= \frac{1}{a(a+b)} + \frac{1}{b(a+b)} \\
&= \frac{(a+b)}{(a+b)(ab)} \\
&= \frac{1}{ab}
\end{aligned}$$

So the jacobian to go from a,b to c,d is  $J = ab$

To get from f,g to c,d we can multiply the jacobians together which leaves us with

$$p(c, d) \propto ab(a+b)^{-5/2}$$

## 4

Joint posterior:

$$p(\alpha, \beta, \theta|y) \propto p(\alpha, \beta) * \prod_{j=1}^J \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1} (1-\theta_j)^{\beta-1} * \prod_{j=1}^J \theta_j^{y_j} (1-\theta_j)^{n_j-y_j}$$

Marginal posterior of a and b:

$$p(\alpha, \beta|y) \propto (\alpha + \beta)^{-5/2} * \prod_{j=1}^J \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} * \frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)}$$

Marginal posterior in the transformed scale (in terms of c and d):

$$\alpha = \frac{e^d e^c}{1+e^c}$$

$$\beta = \frac{e^d}{1+e^c}$$

$$\alpha + \beta = e^d$$

$$p(c, d|y) \propto p(c, d) * \prod_{j=1}^J \frac{\Gamma(e^d)}{\Gamma(\frac{e^d e^c}{1+e^c})\Gamma(\frac{e^d}{1+e^c})} * \frac{\Gamma(\frac{e^d e^c}{1+e^c} + y_j)\Gamma(\frac{e^d}{1+e^c} + n_j - y_j)}{\Gamma(e^d + n_j)}$$

## 5

Metropolis-Hastings algorithm for posterior samples of  $\alpha$  and  $\beta$ .

```

#p(alpha,beta/y)
postab = function(a, b){
  #data
  y <- c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,
        2,1,5,2,5,3,2,7,7,3,3,2,9,10,4,4,4,4,4,4,4,4,10,4,4,4,5,

```

```

11,12, 5,5,6,5,6,6,6,6,16,15,15,9,4)
n <- c(20,20,20,20,20,19,19,19,19,18,17,20,20,20,20,19,19,18,18,25,24,
23,20,20,20,20,20,20,10,49,19,46,27,17,49,47,20,20,13,48,50,20,
20,20,20,20,20,48,19,19,19,22,46,49,20,20,23,19,22,20,20,20,
52,46,47,24,14)

#marginal posterior
p = log(a+b)*(-5/2) +
sum(lgamma(a+b)-lgamma(a)-lgamma(b)+lgamma(a+y)+lgamma(b+n-y)-lgamma(a+b+n))
#p = exp(p)
return(p)
}

map_to_ab = function(c,d){
a = (exp(c)*exp(d))/(1+exp(c))
b = exp(d)/(1+exp(c))
return(c(a,b))
}

one_iteration <- function(start){
#jumping rule = multivariate normal
#proposal
sig = matrix(c(1,0,0,1),2,2)
proposal = mvrnorm(n = 1, mu = c(start[1],start[2]), Sigma = sig)

#map proposal to a,b
prop_ab = map_to_ab(proposal[1],proposal[2])
start_ab = map_to_ab(start[1],start[2])

#calculate r
#postab spits out log probs
#Jacobian = a*b
num = postab(prop_ab[1],prop_ab[2])+log(prop_ab[1]*prop_ab[2])
denom = postab(start_ab[1],start_ab[2])+log(start_ab[1]*start_ab[2])
r = exp(num-denom)

#accept or reject
accept_rate <- min(1, r)
new <- start
if(runif(1) < accept_rate){
new <- proposal
}
#random = runif(1)
#new = ifelse(random < accept_rate,proposal,start)
return(new)
}

MH_postcd <- function(Niter = 1000, burnin = 500){
#blank matrix to store samples
samples_mat <- matrix(NA, nrow = Niter, ncol = 2) #cols = c and d
#random starting point
#samples_mat[1,] = c(-1.671564,1.729955) #sam's #2 answer in terms of c and d
samples_mat[1,] = c(-1,1)

```

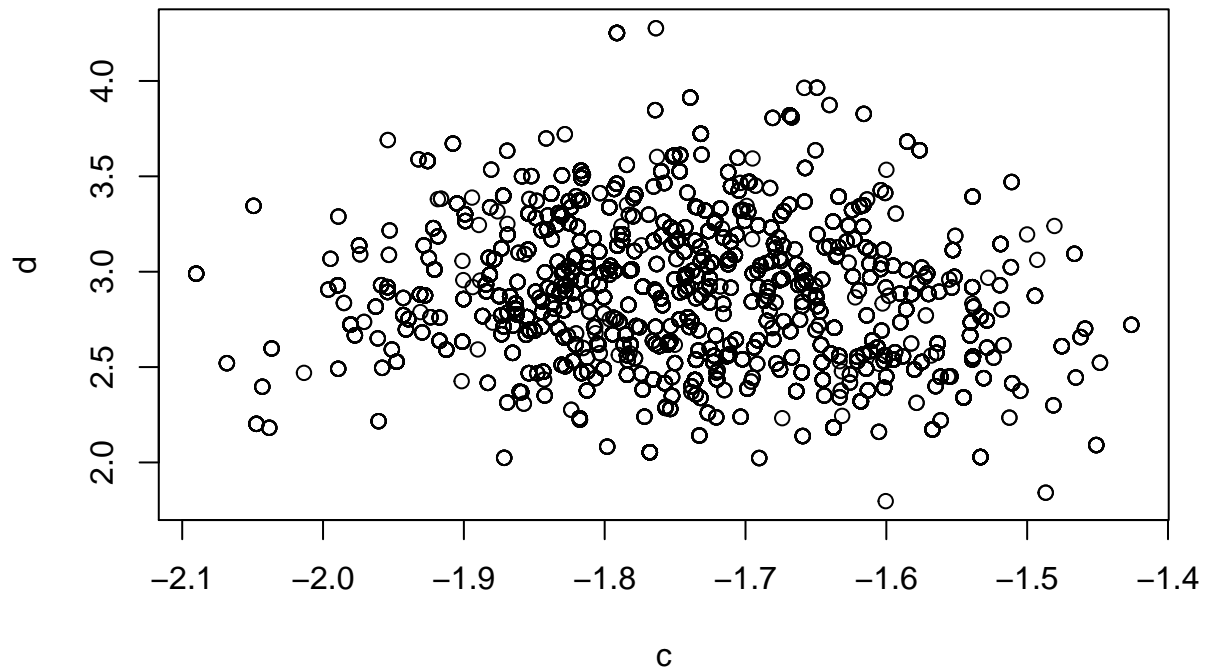
```

#run through algorithm
for(i in 2:Niter){
  samples_mat[i,] <- one_iteration(samples_mat[i - 1,])
  #print(i)
}
return(samples_mat[burnin:Niter,])
}

cd_post_data = MH_postcd(Niter = 10000, burnin = 500)
#cd_post_data

plot(cd_post_data[,1],cd_post_data[,2], xlab = "c", ylab = "d")

```

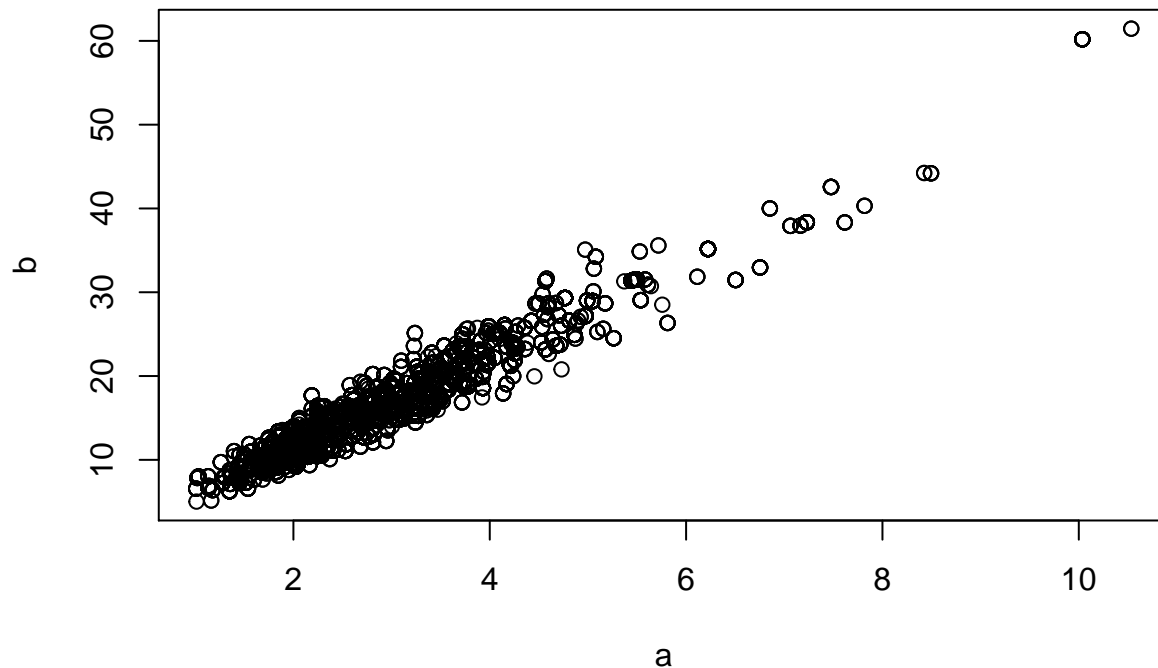


Above is the plot of the posterior samples in terms of  $c$  and  $d$ . The shape is fairly circular, which makes me think that using the bivariate normal without correlation was a pretty good jumping rule. It was able to traverse that space in a fairly efficient way.

```

ab_post_data = apply(cd_post_data,1,function(x) map_to_ab(x[1],x[2]))
plot(ab_post_data[,1],ab_post_data[,2], xlab = "a", ylab = "b")

```



Above is the plot of the posterior samples in terms of  $a$  and  $b$  ( $\alpha$  and  $\beta$ ). The shape shows that there is correlation between  $a$  and  $b$ . The shape matches the results from HW2, which means the metropolis algorithm did a good job of pulling samples from the marginal posterior.

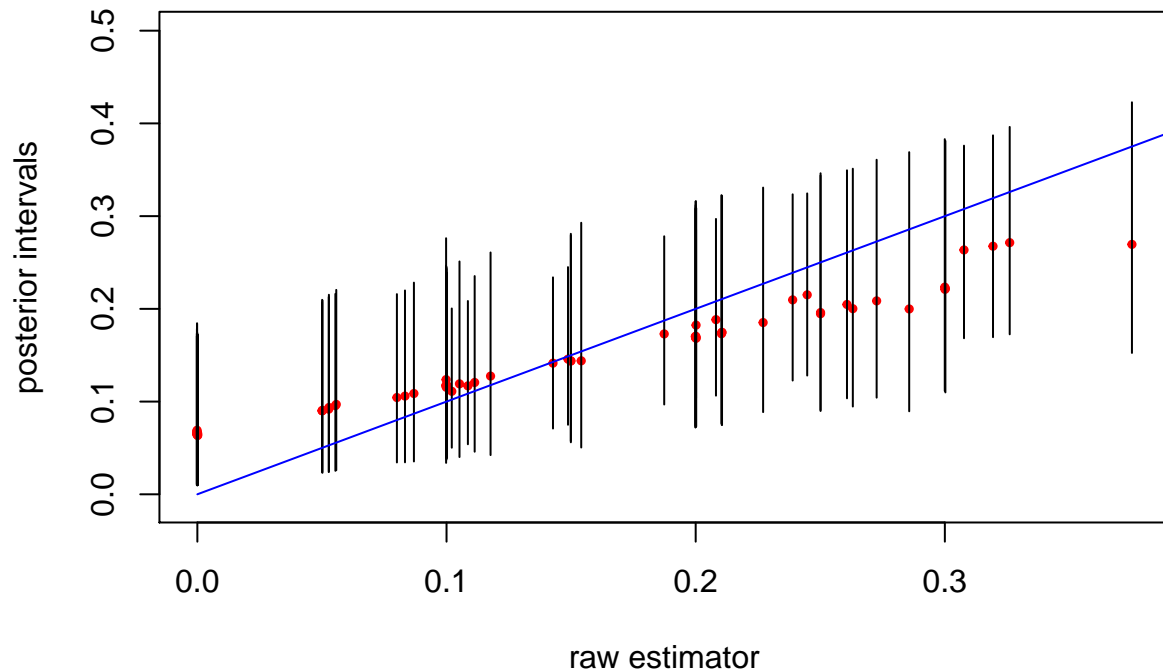
## 6

```
ab_post_data = t(ab_post_data)
theta_mat = matrix(NA, nrow = nrow(ab_post_data), ncol = length(y))
for(i in 1:length(y)){
  theta = apply(ab_post_data, 1, function(x) rbeta(1,x[1]+y[i], x[2]+n[i]-y[i]))
  theta_mat[,i] = theta
}

#figure code
#figure code
postintervals_sample3 <- apply(theta_mat, 2, function(x) c(median(x), c(quantile(x, c(0.025, 0.975))))))

rawest <- jitter(y / n)
plot(rawest, postintervals_sample3[1, ], pch = 19, col = 'red', cex = 0.5, ylim = c(-0.01, 0.5), ylab =
for(k in 1:length(y)){
  lines(cbind(rep(rawest[k], 2), postintervals_sample3[2:3, k]))
}
lines(seq(0, 0.4, by = 0.01), seq(0, 0.4, by = 0.01), col = 'blue')
```





Similar to parts 1 and 2, here is a plot of posterior intervals for the 68 thetas.

7

```
jointpost = function(a, b, theta){
  #theta is a vector of thetas
  #data
  y <- c(0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,
        2,1,5,2,5,3,2,7,7,3,3,2,9,10,4,4,4,4,4,4,4,10,4,4,4,5,
        11,12, 5,5,6,5,6,6,6,6,16,15,15,9,4)
  n <- c(20,20,20,20,20,20,19,19,19,19,18,17,20,20,20,20,19,19,18,18,25,24,
        23,20,20,20,20,20,20,20,10,49,19,46,27,17,49,47,20,20,13,48,50,20,
        20,20,20,20,20,20,48,19,19,19,22,46,49,20,20,23,19,22,20,20,20,
        52,46,47,24,14)

  #log joint posterior
  pab = log(a+b)*(-5/2)
  #print(pab)
  ptheta = sum(lgamma(a+b)-lgamma(a)-lgamma(b)+(a-1)*log(theta)+(b-1)*log(1-theta))
  #print(ptheta)
  py = sum(y*log(theta)+(n-y)*log(1-theta))
  #print(py)
  p = pab+ptheta+py
  #p = exp(p)
  return(p)
}

map_to_cd = function(a,b){
  c = log(a/b)
  d = log(a+b)
  return(c(c,d))
}
```

```

}

#start = c,d,68thetas
one_iteration2 <- function(start){
  thetas = start[3:70]

  #jumping rule = multivariate normal
  #proposal
  sig = matrix(c(1,0,0,1),2,2)
  proposal = mvrnorm(n = 1, mu = c(start[1],start[2]), Sigma = sig)

  #map proposal to a,b
  prop_ab = map_to_ab(proposal[1],proposal[2])
  start_ab = map_to_ab(start[1],start[2])

  #calculate r
  #jointpost spits out log probs
  num = jointpost(prop_ab[1],prop_ab[2],thetas)+log(prop_ab[1]*prop_ab[2])
  denom = jointpost(start_ab[1],start_ab[2],thetas)+log(start_ab[1]*start_ab[2])
  r = exp(num-denom)
  #print(r)
  #print(num)
  #print(denom)

  #accept or reject
  accept_rate <- min(1, r)
  new <- c(start[1],start[2])
  if(runif(1) < accept_rate){
    new <- proposal
  }

  new_ab = map_to_ab(new[1],new[2])

  #draw new thetas
  post_theta_draw = apply(obs_mat,1,function(x) rbeta(1,new_ab[1]+x[1], new_ab[2]+x[2]-x[1]))

  new_vec = c(new,post_theta_draw)
  #print(new)

  return(new_vec)
}

MH_postcd2 <- function(Niter = 1000, burnin = 500){
  #blank matrix to store samples
  samples_mat2 <- matrix(NA, nrow = Niter, ncol = 70) #cols = c and d, 68 thetas
  samples_mat2[1,] = c(-1,1,y/n)+0.0001 #avoid 0 for thetas
  #run through algorithm
  for(i in 2:Niter){
    samples_mat2[i,] <- one_iteration2(samples_mat2[i - 1,])
    #print(i)
  }
  return(samples_mat2[burnin:Niter,])
}

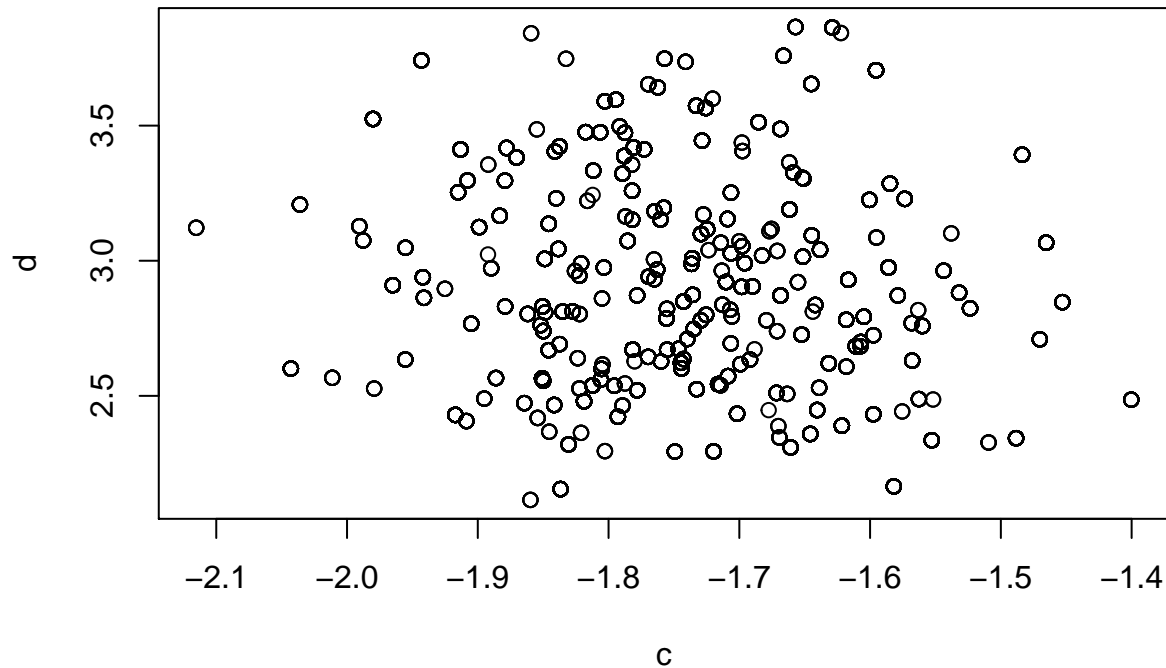
```

```
}
```

```
cd_joint_post_data = MH_postcd2(Niter = 10000, burnin = 500)
```

Here is the posterior plot of c and d. It looks pretty similar to the plot from #5.

```
plot(cd_joint_post_data[,1],cd_joint_post_data[,2], xlab = "c", ylab = "d")
```

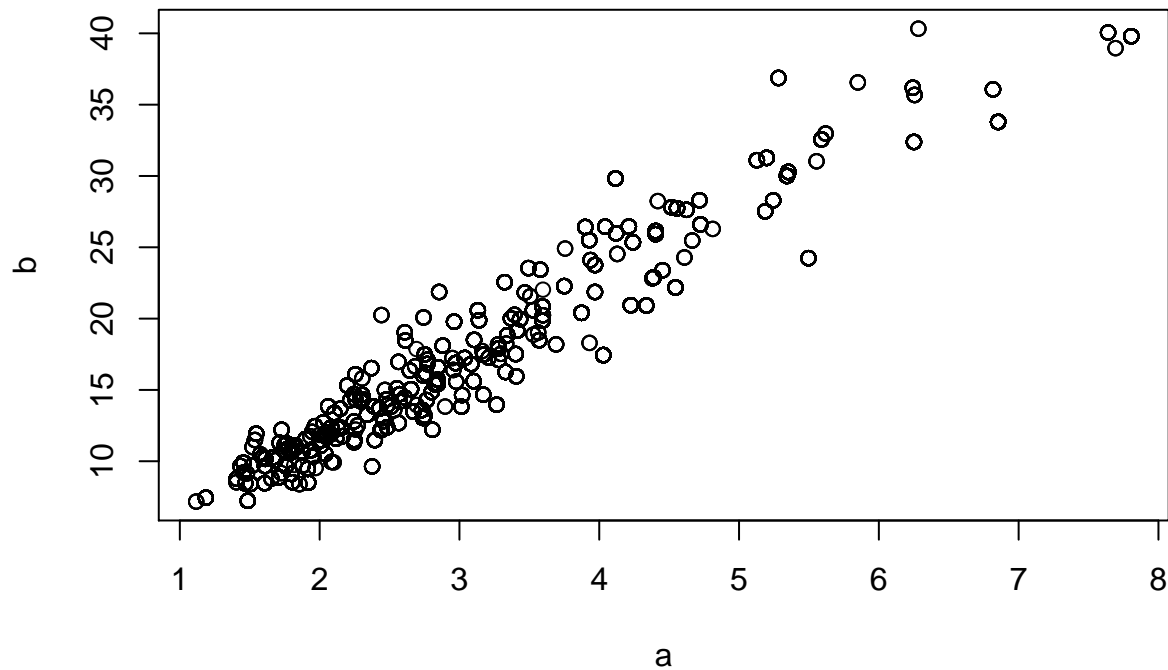


Here is the posterior plot in terms of a and b (alpha and beta). It also looks similar to the plot from #5.

```
ab_joint_post_data = apply(cd_joint_post_data,1,function(x) map_to_ab(x[1],x[2]))
```

```
ab_joint_post_data = t(ab_joint_post_data)
```

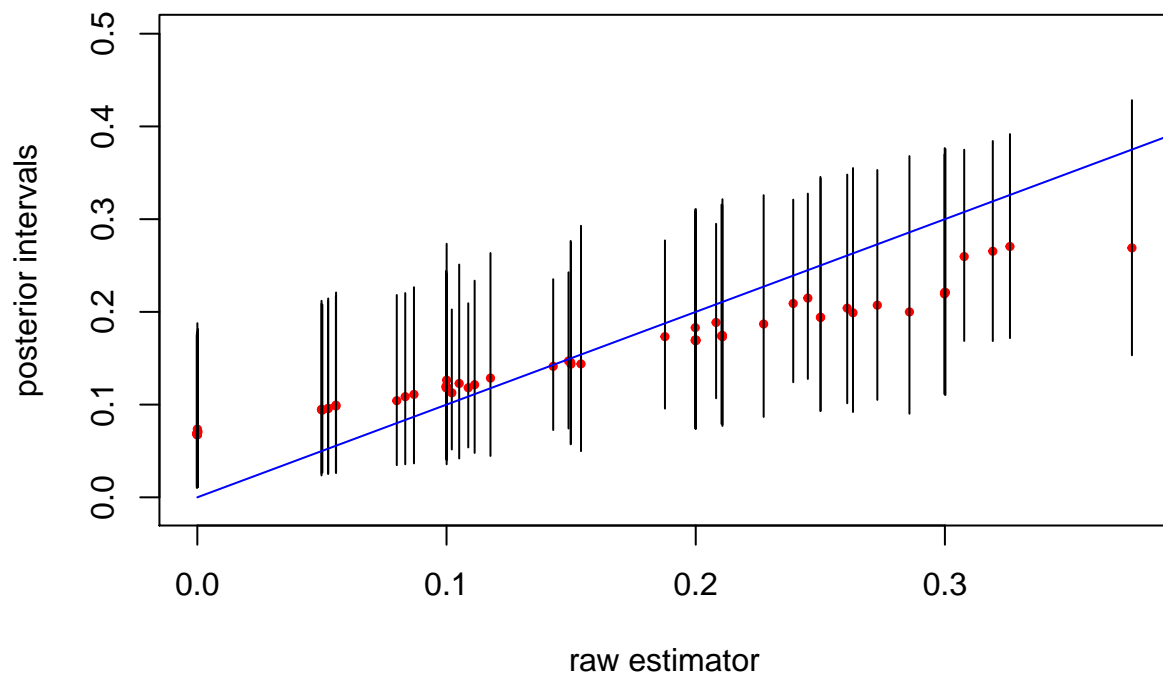
```
plot(ab_joint_post_data[,1],ab_joint_post_data[,2], xlab = "a", ylab = "b")
```



Here is a plot of the posterior sample intervals. This looks similar to the plots from #1, #2, and #6.

```
theta_mat2 = cd_joint_post_data[,3:70]
postintervals_sample4 <- apply(theta_mat2, 2, function(x) c(median(x), c(quantile(x, c(0.025, 0.975))))))

rawest <- jitter(y / n)
plot(rawest, postintervals_sample4[1, ], pch = 19, col = 'red', cex = 0.5, ylim = c(-0.01, 0.5), ylab = 'posterior intervals')
for(k in 1:length(y)){
  lines(cbind(rep(rawest[k], 2), postintervals_sample4[2:3, k]))
}
lines(seq(0, 0.4, by = 0.01), seq(0, 0.4, by = 0.01), col = 'blue')
```



```
#setwd("/Users/oliviahackworth/Desktop/Stats 551")
library(rstan)

## Warning: package 'StanHeaders' was built under R version 3.5.2

data <- list(
  y = c(0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,2,2,
        2,1,5,2,5,3,2,7,7,3,3,2,9,10,4,4,4,4,4,4,10,4,4,4,5,
        11,12, 5,5,6,5,6,6,6,6,16,15,15,9,4),
  J = 68,
  N = c(20,20,20,20,20,19,19,19,19,18,17,20,20,20,20,19,19,18,18,25,24,
        23,20,20,20,20,20,20,10,49,19,46,27,17,49,47,20,20,13,48,50,20,
        20,20,20,20,20,20,48,19,19,19,22,46,49,20,20,23,19,22,20,20,20,
        52,46,47,24,14)
)

#model <- stan_model('rats_hier_unconstrained.stan')
#model <- stan_model('rats2.stan')

#fit <- sampling(model, data = data, iter = 1000, chains = 4)
#samples <- rstan::extract(fit)
```

Unfortunately, I wasn't able to sort out how to get Stan to run on my machine. I got the same error whether I was running the rats stan files that I had written or the 8 schools example files.

I tried running stan files for the model with the tranformed hyperparamers, and the constrained, original hyperparameters. rats2.stan is the file that Katherine, Sam and I worked on together. It looks like:

```
data {
  int<lower=0> J;
  int<lower=0> n[J];
  int<lower=0> y[J];
}

parameters {
  real<lower=0> alpha;
  real<lower=0> beta;
  real<lower=0, upper=1> theta[J];
}

model{
  target += log(alpha + beta)*(-2.5);
  target += beta_lpdf(theta | alpha, beta);
  target += binomial_lpmf(y | n, theta);
}
```

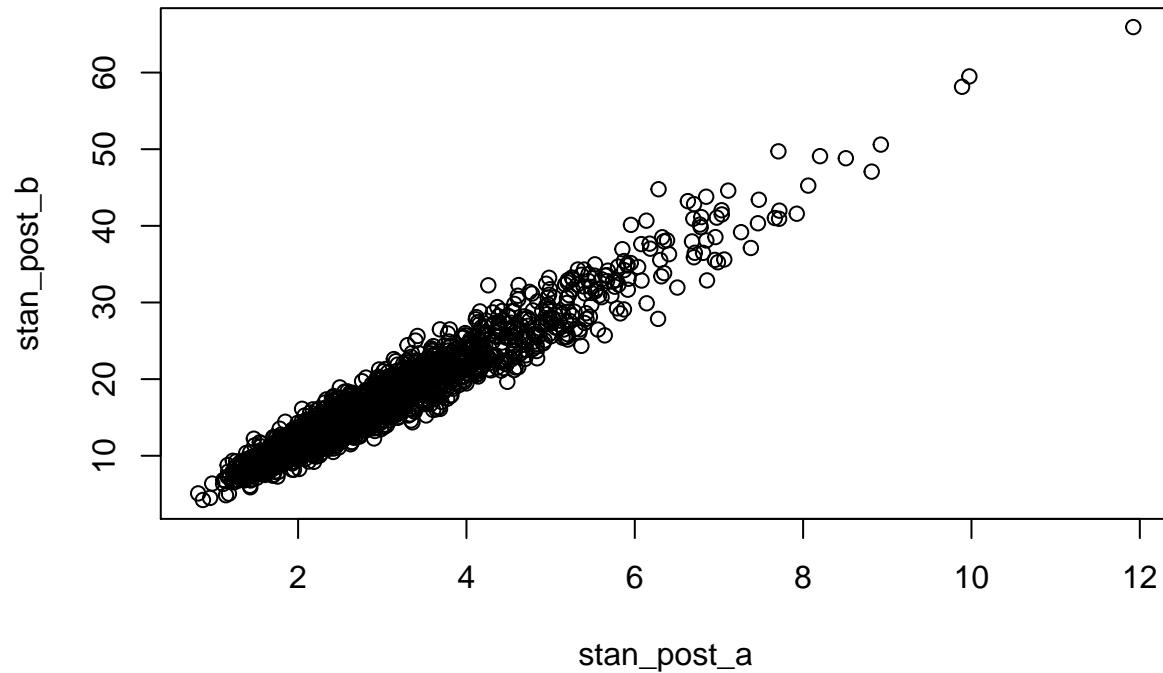
Katherine was able to run the file on her machine, so she shared the model with me. The model and samples for the rest of this part, and to complete part 10 come from her results.

```
Katherine_fit = readRDS('stan_fit')
samples <- rstan::extract(Katherine_fit)
```

Below is the posterior plot of the alphas and betas from Stan.

```
stan_post_a = samples$alpha
stan_post_b = samples$beta
```

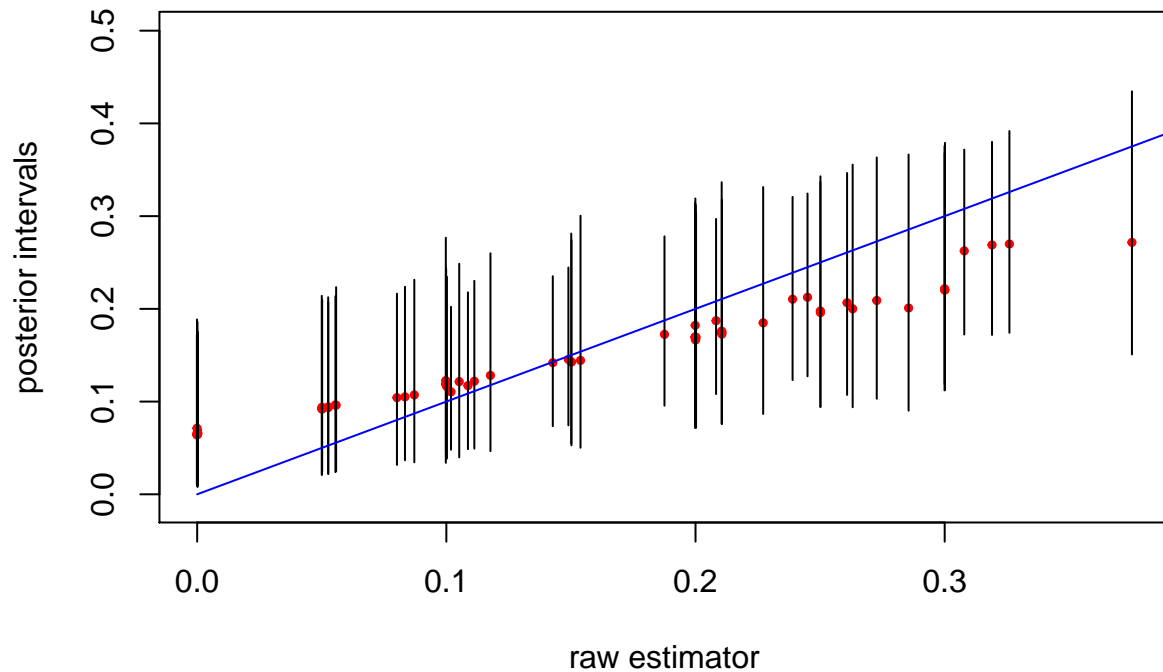
```
stan_post_ab = cbind(stan_post_a, stan_post_b)
plot(stan_post_ab)
```



Here is the posterior intervals of the thetas drawn using Stan.

```
theta_mat3 = samples$theta
postintervals_sample5 <- apply(theta_mat3, 2, function(x) c(median(x), c(quantile(x, c(0.025, 0.975))))))

rawest <- jitter(y / n)
plot(rawest, postintervals_sample5[1, ], pch = 19, col = 'red', cex = 0.5, ylim = c(-0.01, 0.5), ylab =
for(k in 1:length(y)){
  lines(cbind(rep(rawest[k], 2), postintervals_sample5[2:3, k]))
}
lines(seq(0, 0.4, by = 0.01), seq(0, 0.4, by = 0.01), col = 'blue')
```



9

From #7

$E[\alpha|y]$

```
E_a = mean(ab_joint_post_data[,1])
E_a
```

```
## [1] 3.158517
```

$Pr(\frac{\alpha}{\alpha+\beta} < 0.2|y)$

```
mean_vec = ab_joint_post_data[,1]/(ab_joint_post_data[,1]+ab_joint_post_data[,2])
Prob_post = length(mean_vec[mean_vec<0.2])/length(mean_vec)
Prob_post
```

```
## [1] 1
```

10

From #8

$E[\alpha|y]$

```
E_a2 = mean(stan_post_a)
E_a2
```

```
## [1] 2.971643
```

$Pr(\frac{\alpha}{\alpha+\beta} < 0.2|y)$

```
mean_vec2 = stan_post_a/(stan_post_a+stan_post_b)
Prob_post2 = length(mean_vec2[mean_vec2<0.2])/length(mean_vec2)
Prob_post2
```

```
## [1] 1
```

## Conclusion

This homework shows how all three methods (metropolis for hyperparameters then drawing parameters, metropolis/gibbs for hyperparameters and parameters, Stan) lead to similar results. It illustrates that when you have a well defined model and scenario, you have options for how to draw samples from posterior distributions. It also hints at the power and efficiency of Stan. The Stan portion has very few lines of code compared to part 7, but leads to the same place. In a scenario with a less clearly laid out problem, Stan would be a good option to get to the posterior samples.