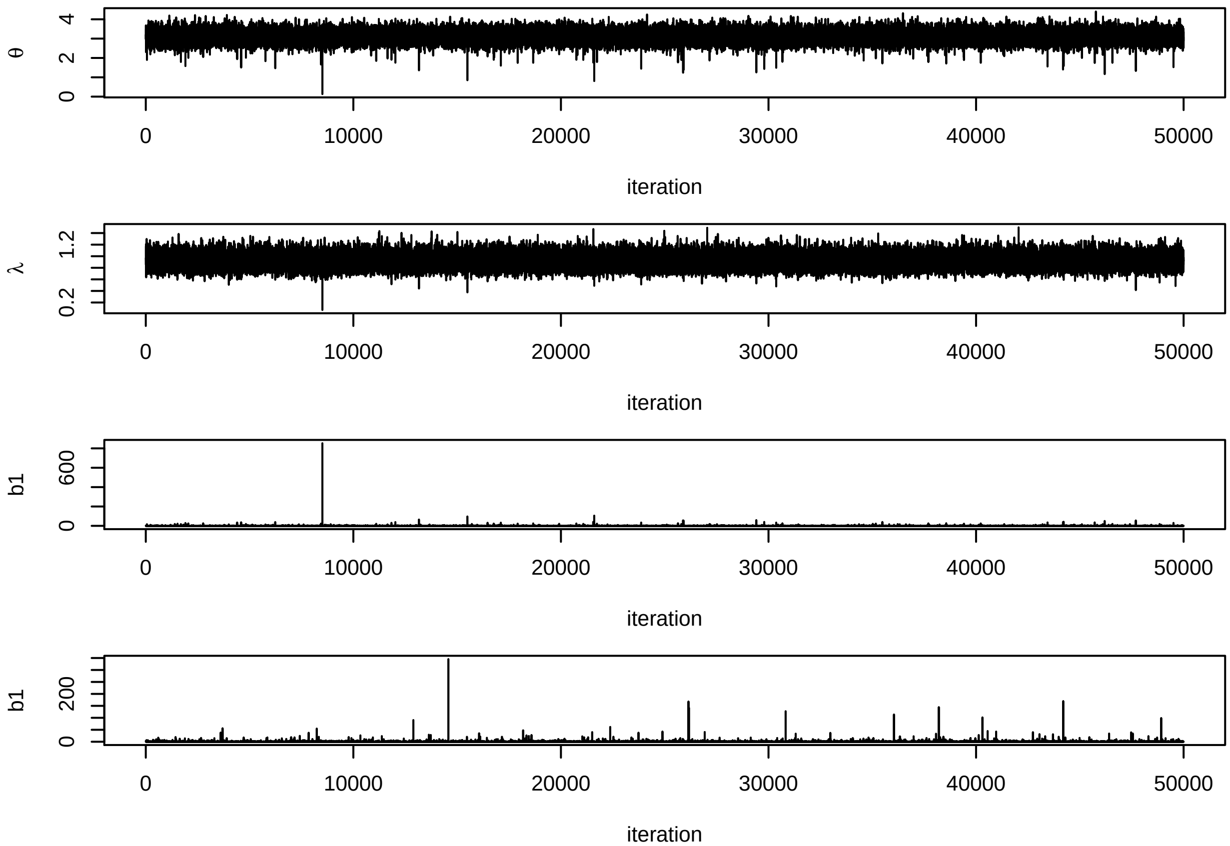
Homework 3 PubH 7440

Jake Wittman

3/1/2020



1a)

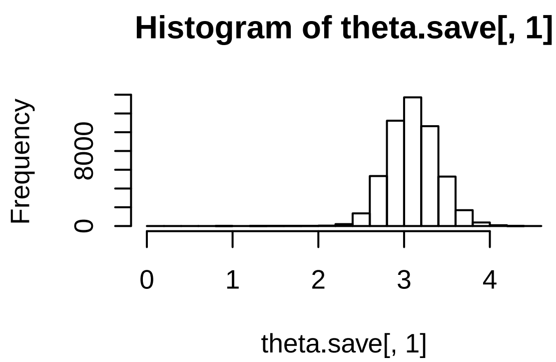
```r
par(mfrow = c(2, 2))
summary(theta.save)
```

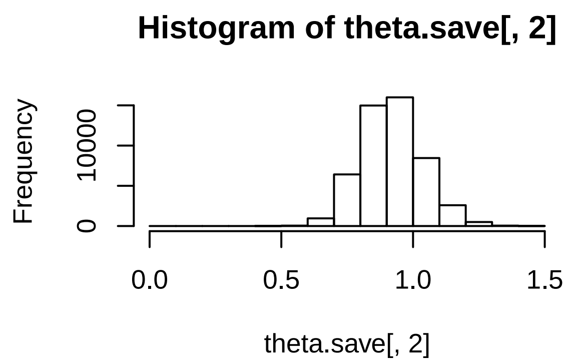
```
##        V1               V2         
##  Min.   :0.1306   Min.   :0.07091  
##  1st Qu.:2.9095   1st Qu.:0.83905  
##  Median :3.0980   Median :0.91479  
##  Mean   :3.1034   Mean   :0.91872  
##  3rd Qu.:3.2912   3rd Qu.:0.99289  
##  Max.   :4.4029   Max.   :1.49858  
```

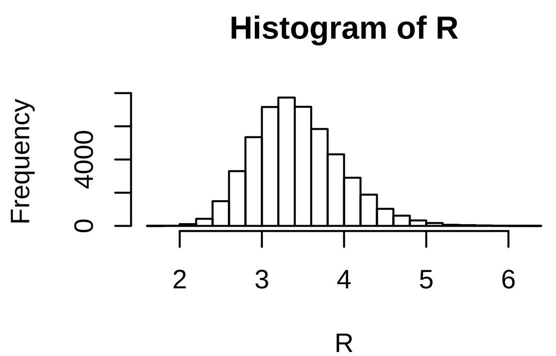
```r
hist(theta.save[, 1])
hist(theta.save[, 2])


R <- theta.save[, 1] / theta.save[, 2]
summary(R)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.650   3.053   3.384   3.430   3.757   6.229
```

```r
hist(R)
```





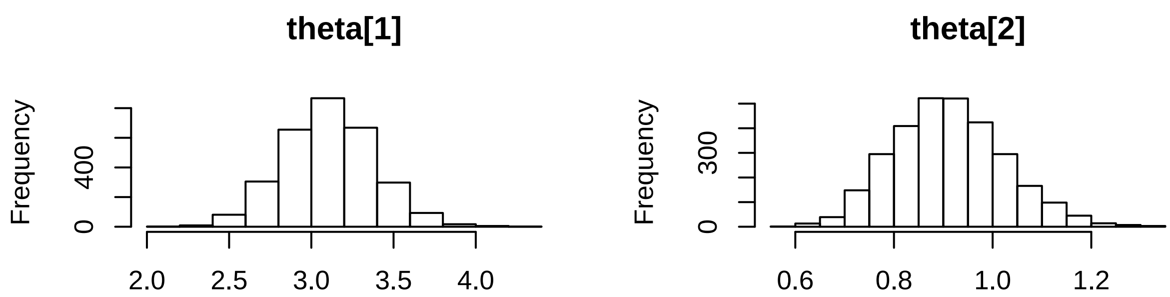


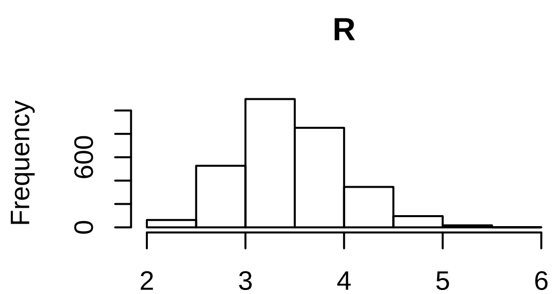
Theta[1] representes θ and Theta[2] is λ. The average number of accidents is about 3 times higher before 1890 than after (R = 3.43).

1b)

```
## Inference for Bugs model at "poisson_model", fit using jags,
##  3 chains, each with 10000 iterations (first 1000 discarded), n.thin = 9
##  n.sims = 3000 iterations saved
##          mu.vect sd.vect    2.5%     25%     50%     75%   97.5% Rhat n.eff
## R          3.460   0.533   2.525   3.088   3.415   3.797   4.616 1.001  3000
## theta[1]   3.106   0.275   2.583   2.921   3.102   3.286   3.674 1.001  3000
## theta[2]   0.912   0.112   0.713   0.831   0.907   0.986   1.143 1.001  3000
## deviance 340.234   1.979 338.286 338.818 339.619 341.039 345.649 1.001  3000
```

```
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 2.0 and DIC = 342.2
## DIC is an estimate of expected predictive error (lower deviance is better).
```





The JAGS output is very similar to the Gibbs sampler I wrote above.

1c) To include k as a parameter to be estimated, we could use the unnormalized density $p^*(k|\theta, \lambda, y)$. We can draw samples from this unnormalized density and use rejection sampling with a Normal proposal density to draw estimates of our parameter k. These draws can then be used in Gibbs sampler to sample for the other parameters.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
# 1a

# Gibbs sampler
rm(list=ls())
set.seed(12345)
```

```r
library(invgamma) # this contains the function to generate inv-gamma random variable

### First, read data from txt file
data <- read.csv(here::here("homework/coalminingdisaster.csv"))
n <- nrow(data)
year <- data$Year
y <- data$Count


### Specify priors
a1 <- a2 <- 0.5
c1 <- c2 <- 1
d1 <- d2 <- 1
k <- 40
group <- c(rep(1, k), rep(2, (n - k)))

### Specify MCMC runs, burnin numbers
runs <- 50000
burn <- 1000


### Specify initial values
theta.init <- rep(1, 2)
b.init <- c(1, 1)


### Create arrays to store MCMC samples
theta.save <- array(NA, c(runs, 2))
b.save <- array(NA, c(runs,2))



### Now conduct Gibbs sampling
theta <- theta.init
b <- b.init


for(iter in 1:(runs+burn)){

    # at each iteration t

    # Step 1: generate theta from Gamma full conditional
    for(i in 1:2) {
        theta[i] <- rgamma(1,
                           ((length(y[group == i]) * mean(y[group == i])) + 0.5),
                           rate = b[group[i]] + length(y[group == i]))
    }

   for(i in 1:2) {
     b[i] <- rinvgamma(1, 0.5 + 1, scale = (sum(theta[group[i]]) + 1)/1)
   }

    # save the current value of theta after burnin interations
```

```r
    if(iter>burn) {
        theta.save[iter-burn,] <- theta
        b.save[iter-burn,] <- b
    }
}

### Check convergence
par(mfrow=c(4,1),mar=c(4,4.5,1,0.5))
plot(theta.save[, 1], type='l', xlab='iteration', ylab=expression(theta))
plot(theta.save[, 2], type='l', xlab='iteration', ylab=expression(lambda))
plot(b.save[, 1], type='l', xlab='iteration', ylab = "b1")
plot(b.save[, 2], type='l', xlab='iteration', ylab = "b1")
par(mfrow = c(2, 2))
summary(theta.save)
hist(theta.save[, 1])
hist(theta.save[, 2])


R <- theta.save[, 1] / theta.save[, 2]
summary(R)
hist(R)


# Fit model in JAGS
set.seed(12345)
pacman::p_load(rjags,
               invgamma,
               R2jags)
#devtools::install.packages("jagsplot")
library(jagsplot)
y1 <- y[1:k]
y2 <- y[(k + 1):n]

poisson_model <- function() {

  # Priors
  for (i in (1:2)) {
    invb[i] ~ dgamma(1, 1)
    b[i] <- 1 / invb[i]
    theta[i] ~ dgamma(0.5, b[i])
  }

  # Likelihood
  for (i in 1:n) {
      y[i] ~ dpois(theta[group[i]])
  }

  R <- theta[1] / theta[2]
}

jags.data <- list(y = y,
                  n = n,
                  k = k,
```

```r
                  group = group)

params <- c("theta", "R")
jags.sample <- jags.parallel(model = poisson_model,
                             parameters.to.save = params,
                             data = jags.data,
                             n.chains = 3,
                             n.burnin = 1000,
                             n.iter = 10000)

jags.sample
par(mfrow = c(2, 2))
jags.hist(jags.sample, which = c(3, 4, 1))
```