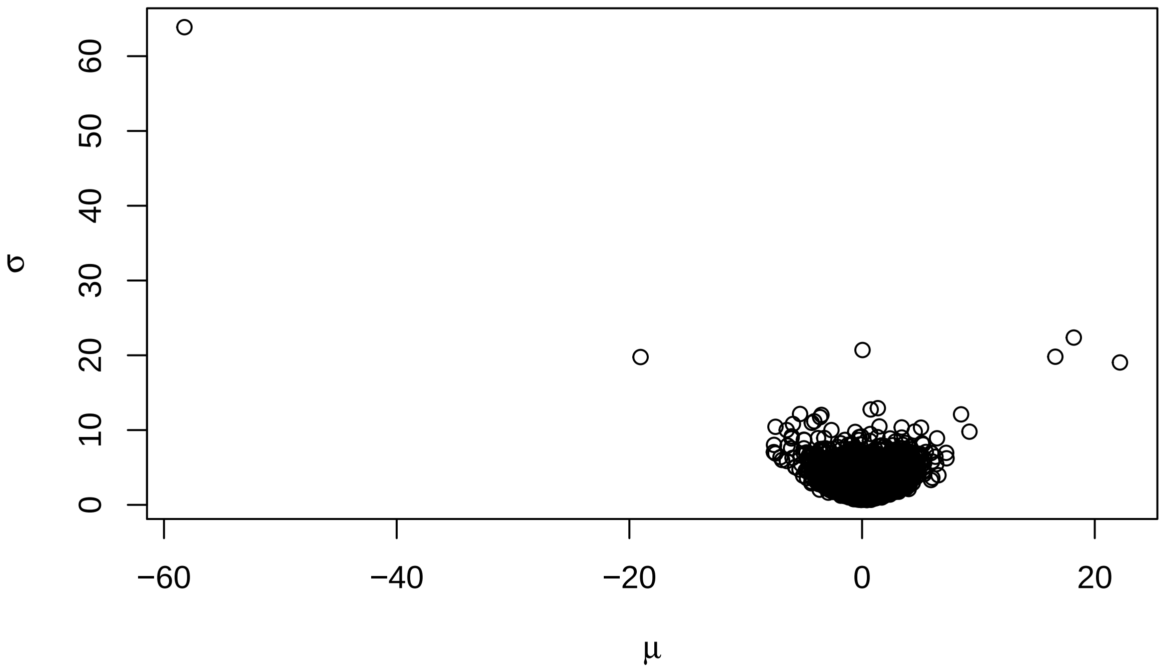
Homework 5

Drew Dahlquist

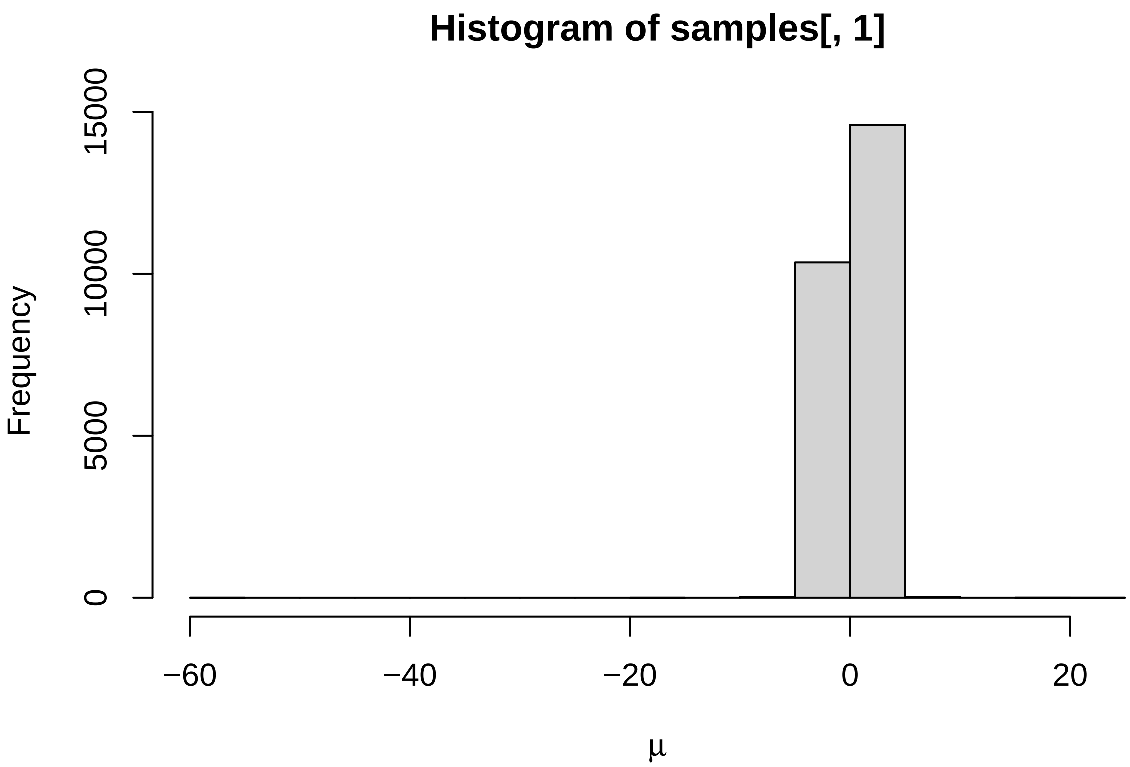
3/12/2022

1)

```
# Load the data

Y = c(2.68,1.18,-0.97,-0.98,-1.03)
n = length(Y)

# Create an empty matrix for the MCMC samples

S = 25000
samples = matrix(NA,S,2)
colnames(samples) = c('mu', 'sigma')

# Initial values

mu = 10^(-100)
sig2 = 10^(100)

# prior: mu ~ N(gamma, tau), sig2 ~ InvG(a,b)

gamma = 0
tau = 100^2
a = 0.1
b = 0.1

# Gibbs sampling

for(s in 1:S) {
  P = n/sig2 + 1/tau
  M = sum(Y)/sig2 + gamma/tau
  mu = rnorm(1,M/P,1/sqrt(P))

  A = n/2 + a
  B = sum((Y-mu)^2)/2 + b
  sig2 = 1/rgamma(1,A,B)

  samples[s,] = c(mu,sqrt(sig2))
}

# Plot the join posterior and marginal of mu
plot(samples,xlab=expression(mu),ylab=expression(sigma))
```




```
hist(samples[,1],xlab=expression(mu))
```



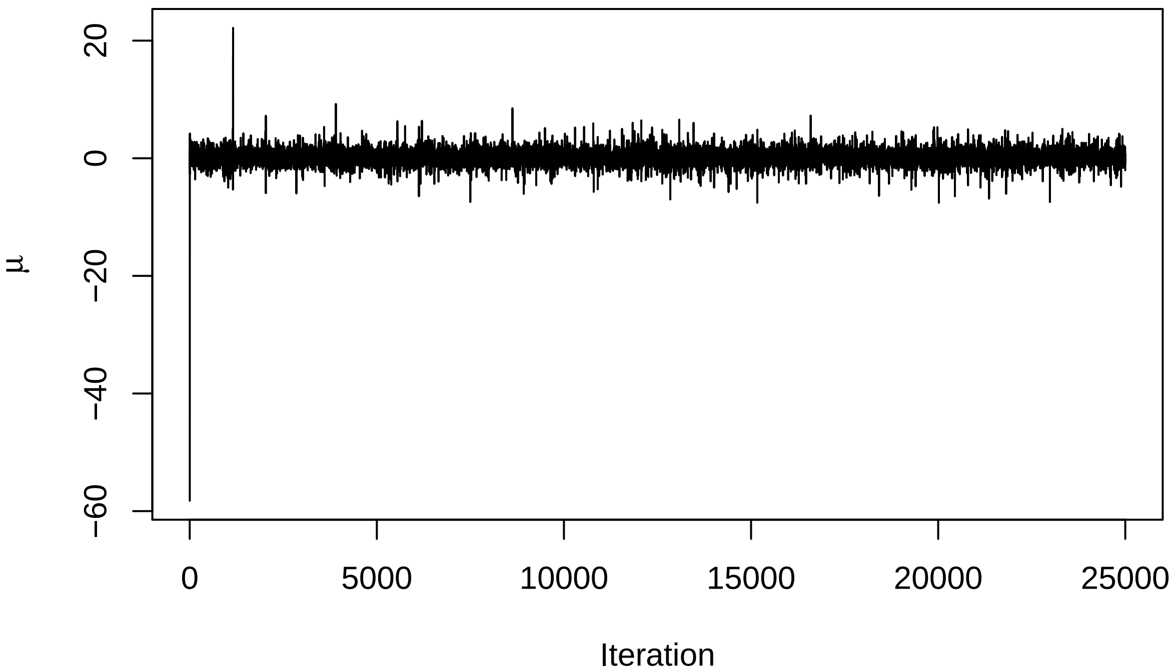

```
# Posterior mean, median, and credible intervals
apply(samples,2,mean)
```

```
##        mu     sigma
## 0.1737721 2.0550475
```

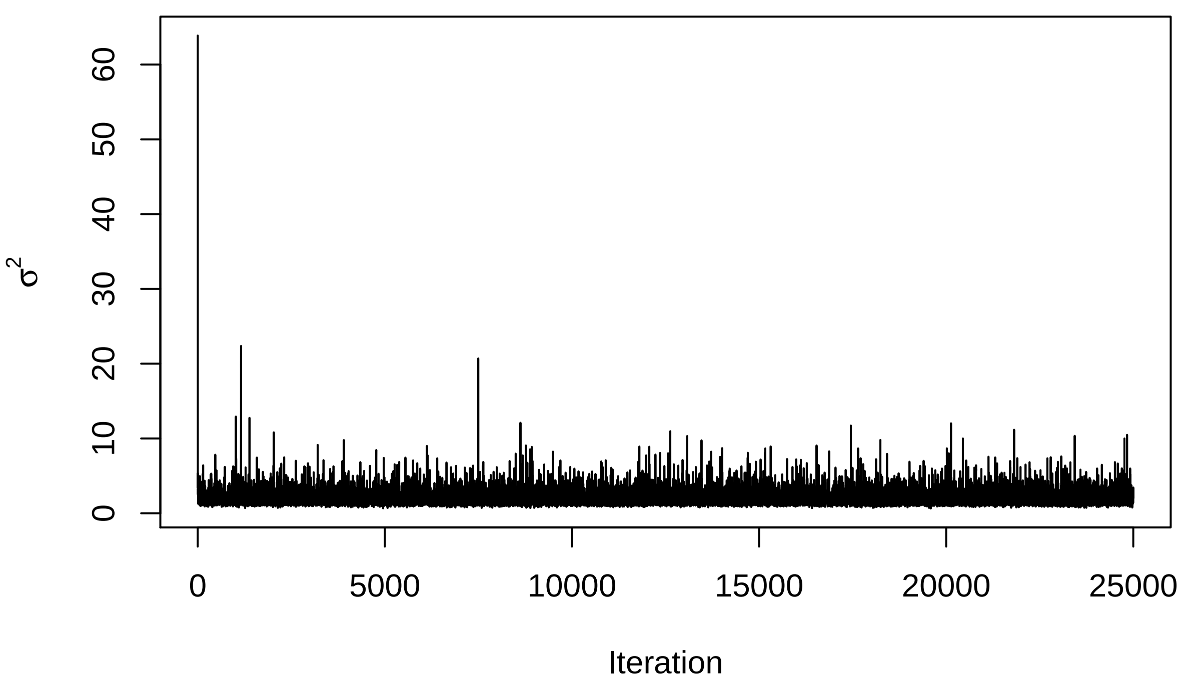
```r
apply(samples,2,quantile,c(0.025,0.500,0.975))
```

```
##              mu    sigma
## 2.5%  -1.857300 1.010176
## 50%    0.171849 1.810441
## 97.5%  2.247542 4.534349
```

```r
plot(samples[,1],type='l',xlab='Iteration',ylab=expression(mu))
```




```r
plot(samples[,2],type='l',xlab='Iteration',ylab=expression(sigma^2))
```



Setting mu = mean(Y) and sig2 = var(Y) results in almost instant convergence, and most values near those estimates have the same results. Even with very unreasonable estimates, such as mu = 10^100 and sig2 = 10^(-100), the chain still converges almost instantly. The only noticeable affects I was able to get where when both mu and sig2 were very large (10^100), but this would still require only very minor burn-in.

2)

(a)

$Y_i \sim Poisson(N_i \lambda_i)$

$\lambda_i | \gamma$ sim $Gamma(1, \gamma)$

$\gamma \sim Gamma(0.001, 0.001)$

(b)

Step 1: Select initial values for λ_1, λ_2, λ_3, λ_4, λ_5, λ_6, λ_7, and γ

Then for s = 1 . . . S, iterate through the following:

Step 2a: $p(\lambda_1 | Y, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \gamma) \sim Gamma(Y_1 + 1, N_1 + \gamma)$

Step 2b: $p(\lambda_2 | Y, \lambda_1, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \gamma) \sim Gamma(Y_2 + 1, N_2 + \gamma)$

Step 2c: $p(\lambda_3 | Y, \lambda_1, \lambda_2, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \gamma) \sim Gamma(Y_3 + 1, N_3 + \gamma)$

Step 2d: $p(\lambda_4 | Y, \lambda_1, \lambda_2, \lambda_3, \lambda_5, \lambda_6, \lambda_7, \gamma) \sim Gamma(Y_4 + 1, N_4 + \gamma)$

Step 2e: $p(\lambda_5 | Y, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_6, \lambda_7, \gamma) \sim Gamma(Y_5 + 1, N_5 + \gamma)$

Step 2f: $p(\lambda_6 | Y, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_7, \gamma) \sim Gamma(Y_6 + 1, N_6 + \gamma)$

Step 2g: $p(\lambda_7 | Y, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \gamma) \sim Gamma(Y_7 + 1, N_7 + \gamma)$

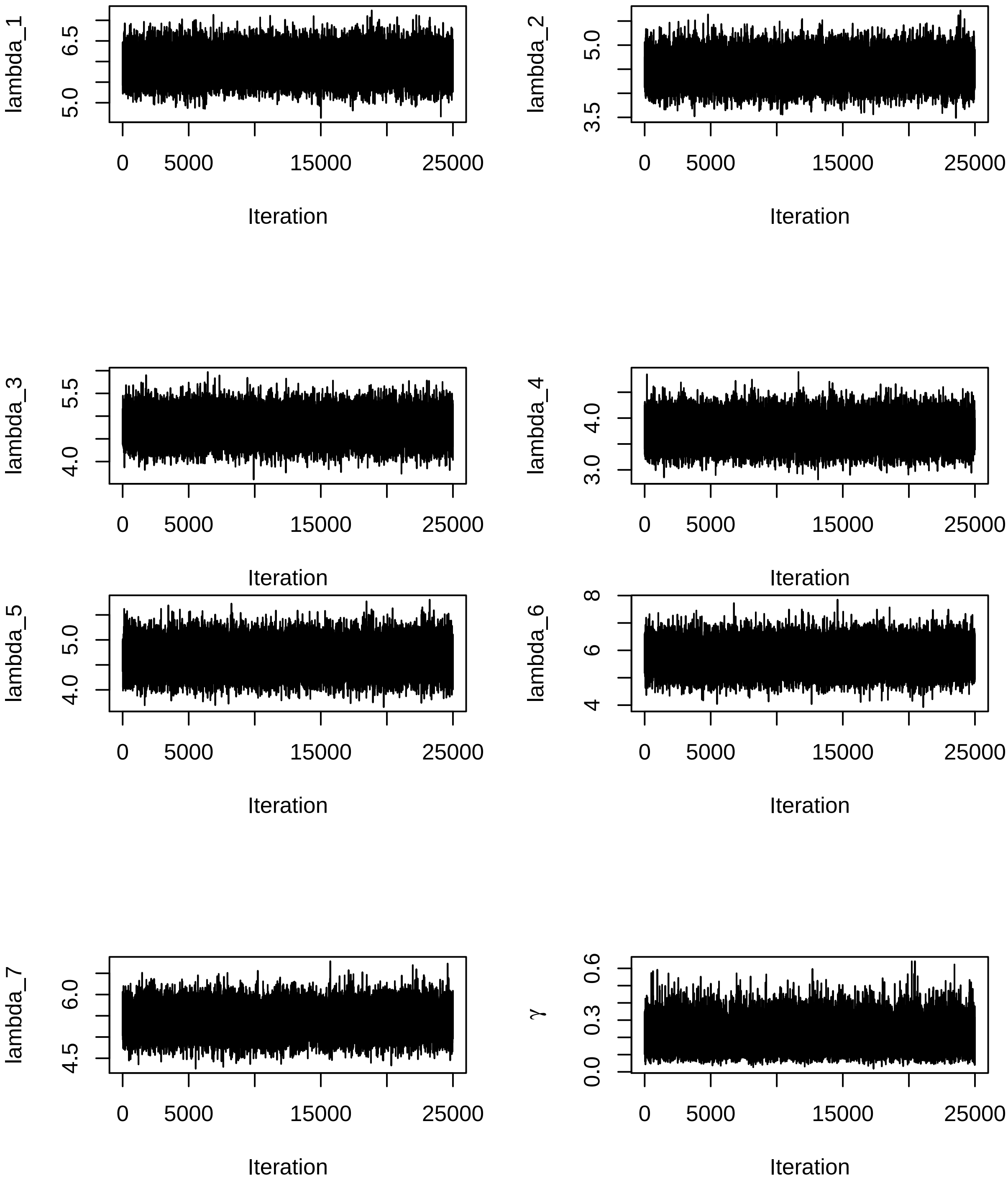
Step 2h: $p(\gamma | Y, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7) \sim Gamma(7 + a, \sum_{i=1}^{7} \lambda_i + b)$

```r
# Load data
Y = c(343,261,271,220,278,149,317)
n = length(Y) # = 7
N = c(58,58,57,59,60,26,59)

# Create empty matrix for MCMC samples
S = 25000
samples = matrix(NA,S,n+1)
colnames(samples) = c('lam1','lam2','lam3','lam4','lam5','lam6','lam7','gamma')

# Initial values
lambda = log(Y/N)
gamma = 1/mean(lambda)

# priors: lambda[i]|gamma ~ Gamma(1, gamma), gamma ~ InvG(a,b)
a = 0.001
b = 0.001

# Gibbs sampling
for(s in 1:S) {
  # sample for lambda_i
  for(i in 1:n) {
    lambda[i] = rgamma(1,Y[i]+1,N[i]+gamma)
  }
  # sample for gamma
  gamma = rgamma(1,n+a,sum(lambda)+b)
```

```r
  # record
  samples[s,]=c(lambda,gamma)
}
```

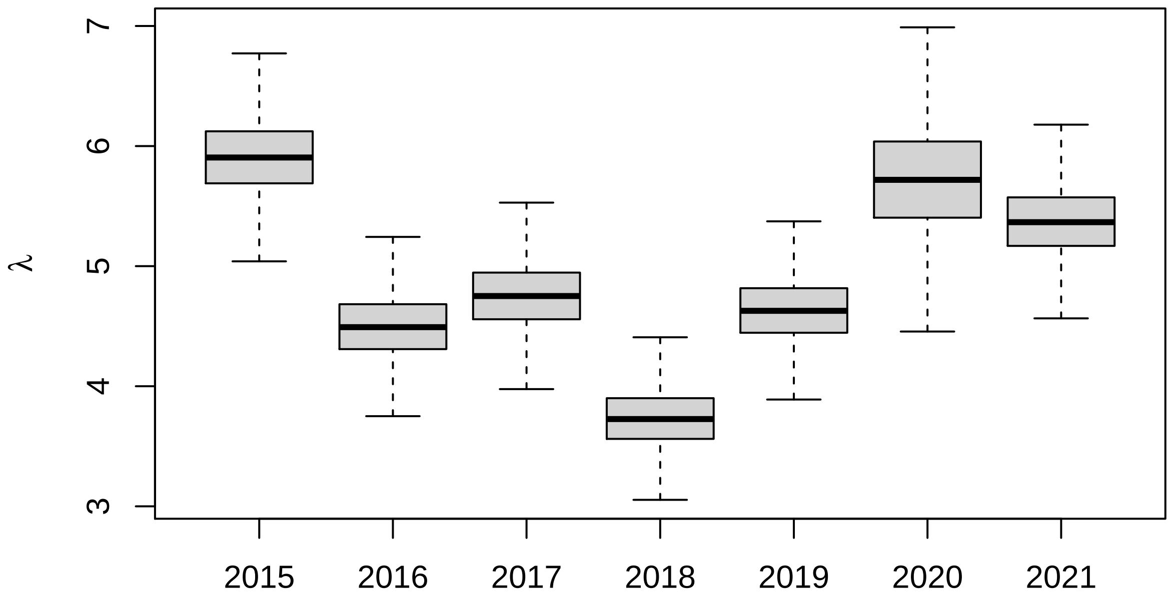
(c)

All model parameters appear to converge almost instantly as shown by the trace plots, likely because this is a relatively simple model with all parameters highly connected to the data. Therefor, the burn-in period will be very small if anything, and we could cut out the first few hundred samples if desired.



(d)

2015 appears to have the highest strikeout rate per game, whereas 2018 looks to have the lowest. 2015 may have been an anomaly, as the strikeout rate had a sudden drop-off and stayed pretty consistent for the 4 years after. There was a steady rise in strikeout rate form 2018 to 2020, which looks to have leveled-out in 2021. Also, 2020 has a noticeably wider confidence interval compared to all the others years, likely since so few games were played that year in comparison to the rest.



(e)

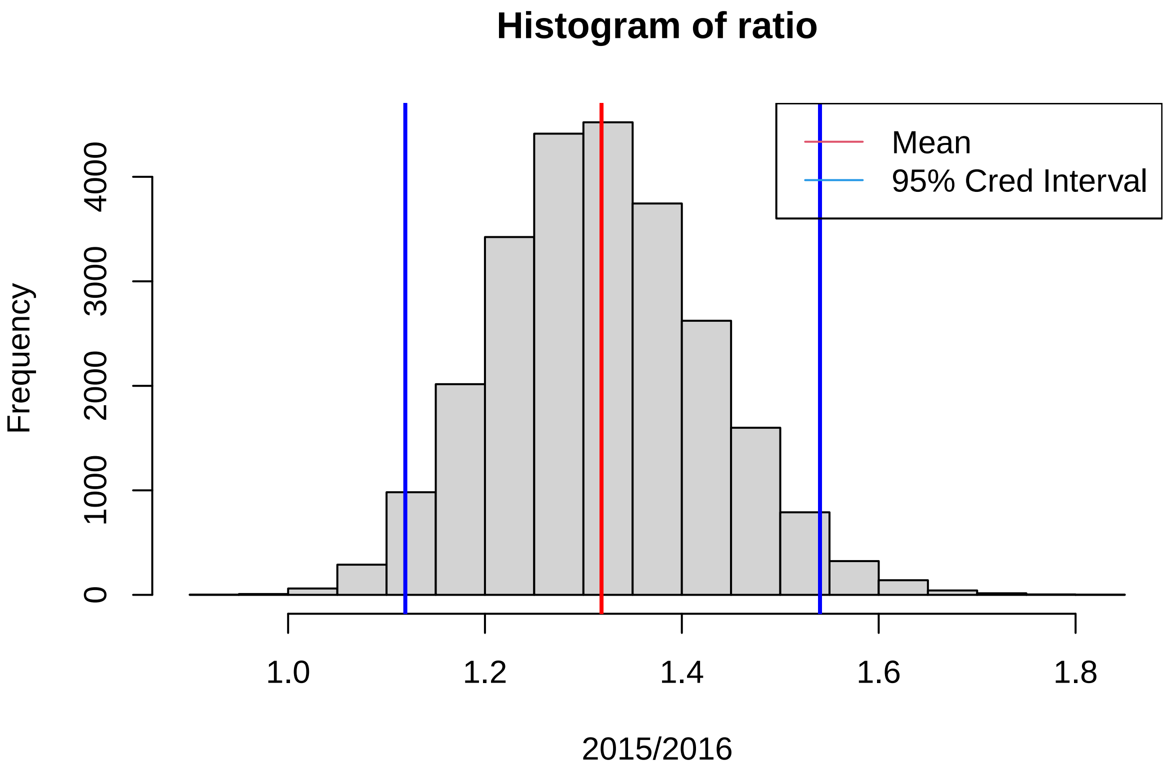
```
##                       lamb 1   lamb 2   lamb 3   lamb 4   lamb 5   lamb 6
## Posterior mean:     5.909615 4.499404 4.756351 3.732872 4.634347 5.727382
## 2.5th Percentile:   5.303919 3.977087 4.208039 3.253040 4.108062 4.844219
## 97.5th Percentile:  6.548702 5.056417 5.333440 4.246318 5.196841 6.675926
##                       lamb 7
## Posterior mean:     5.373292
## 2.5th Percentile:   4.795984
## 97.5th Percentile:  5.991898
```

(f)

The strikeout rate in 2020 was slightly higher than in 2021, but this result likely isn't very significant because of 2020's wide 95% credible interval (which actually includes all of 2021's 95% credible interval).

(g)

Given the results of the hypothesis test, it seems likely that the ratio of strikeout rates for these two seasons is greater than one. Viewing the histogram of 2015/2016, the posterior mean estimate is about 1.33, and the 95% credible interval is strictly greater than 1.0.



3)

(a)

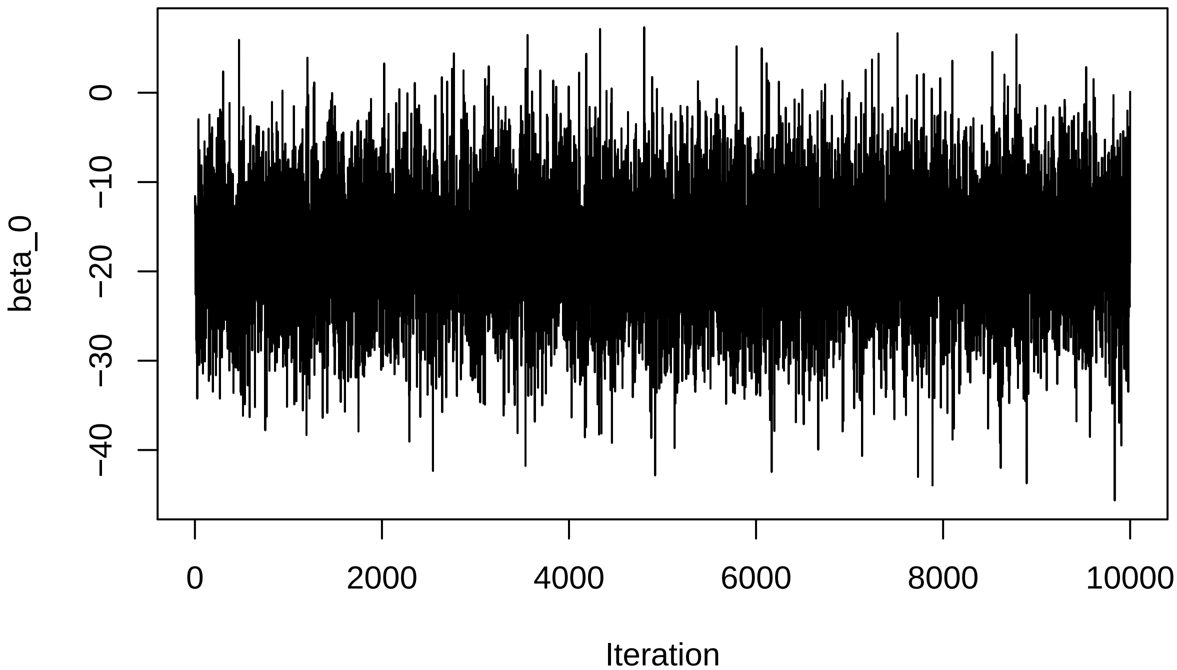
For my beta vector I chose a Normal with mean 0 and standard deviation of 0.001, and for my variance I chose an uninformative Inverse Gamma (i.e. a = b = 0.1).

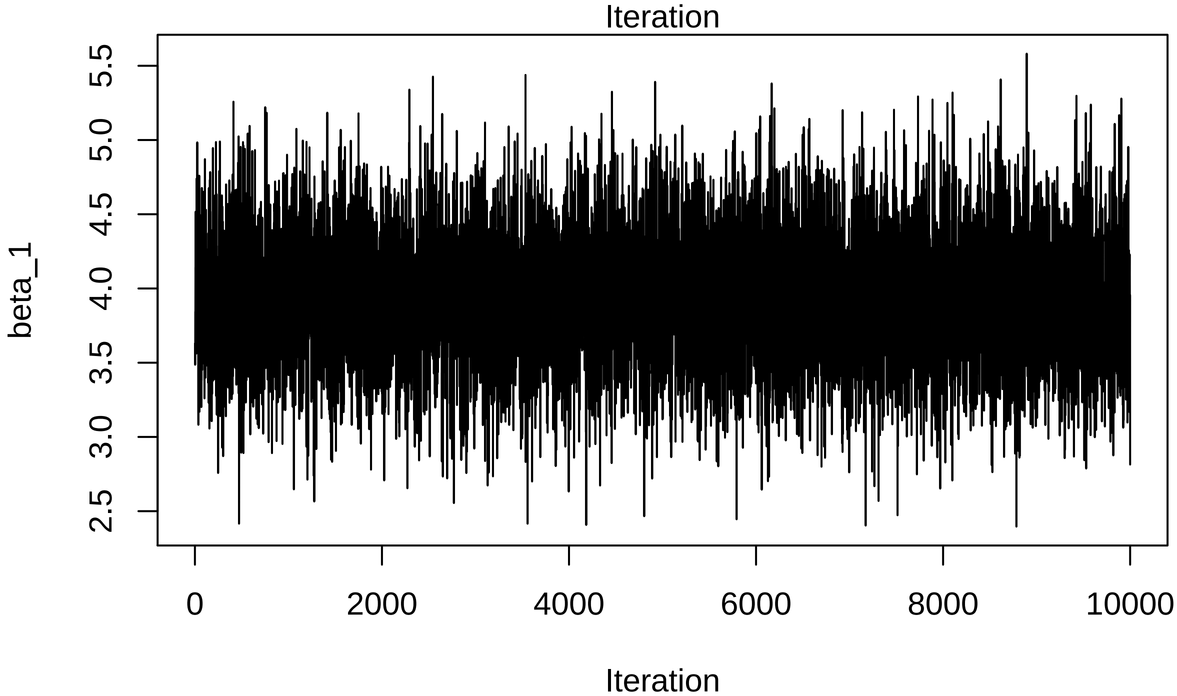
$\beta \sim N(0, 0.001)$

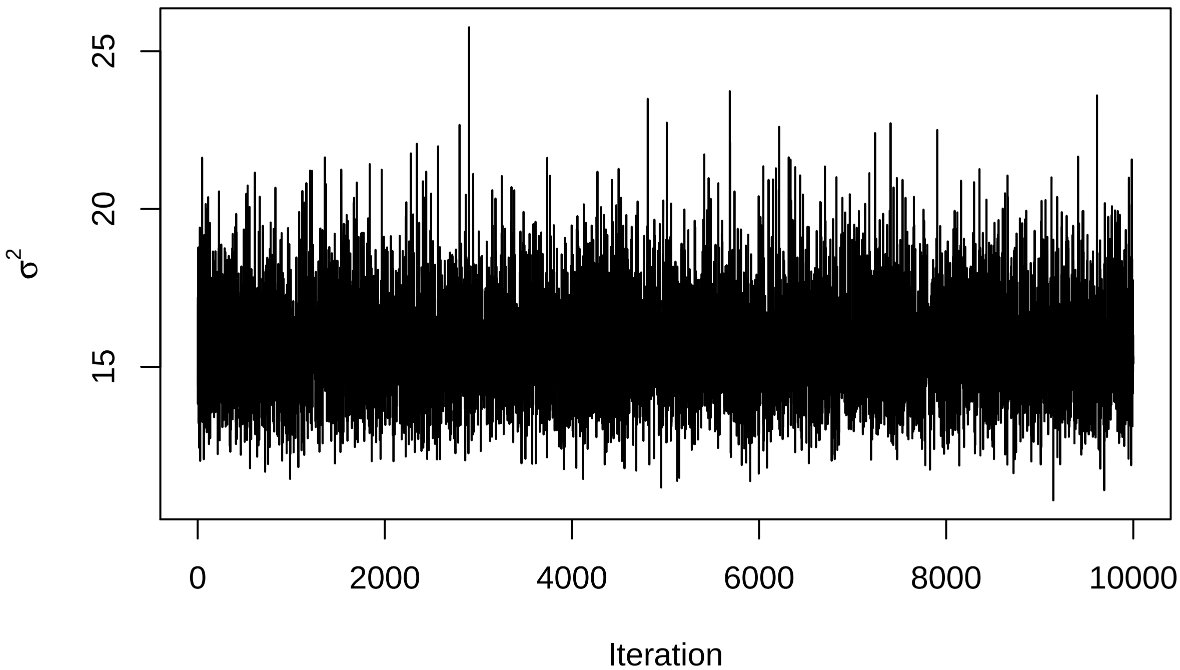
$\sigma \sim InvGamma(0.1, 0.1)$

(b)

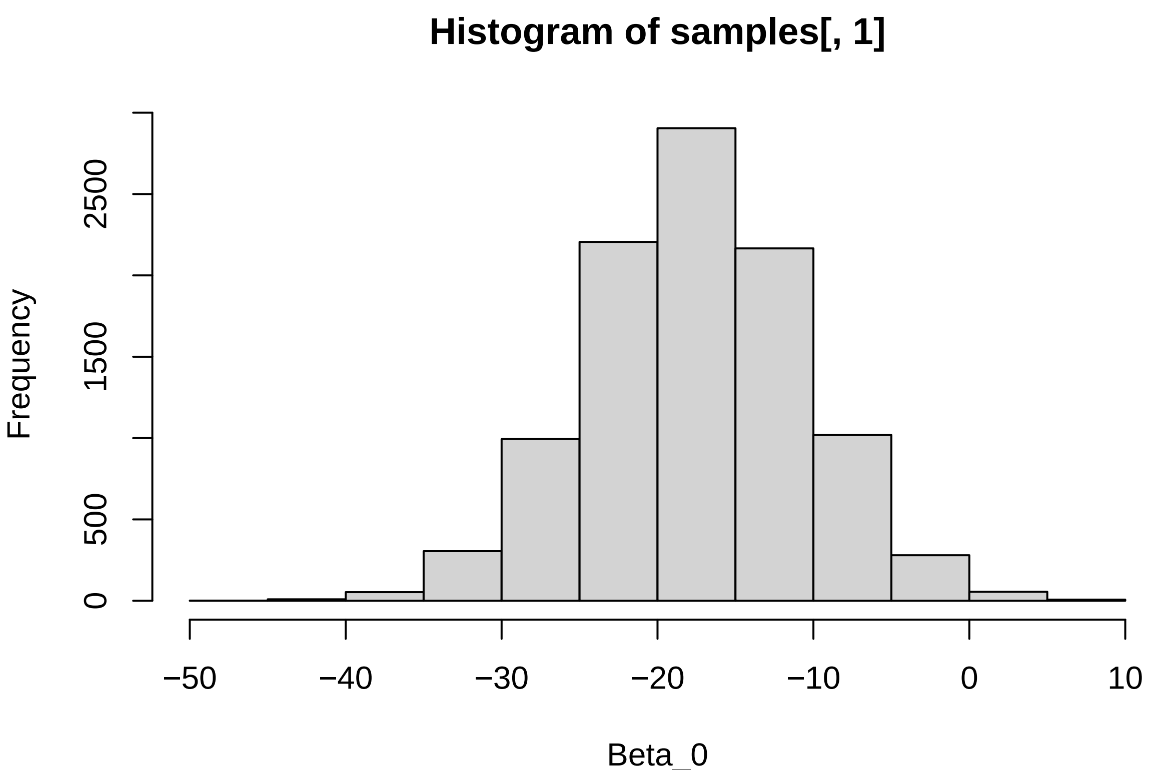
Again, all model parameters appear to converge almost instantly as shown by the trace plots. For burn-in, we could truncate the first few hundred values if desired.

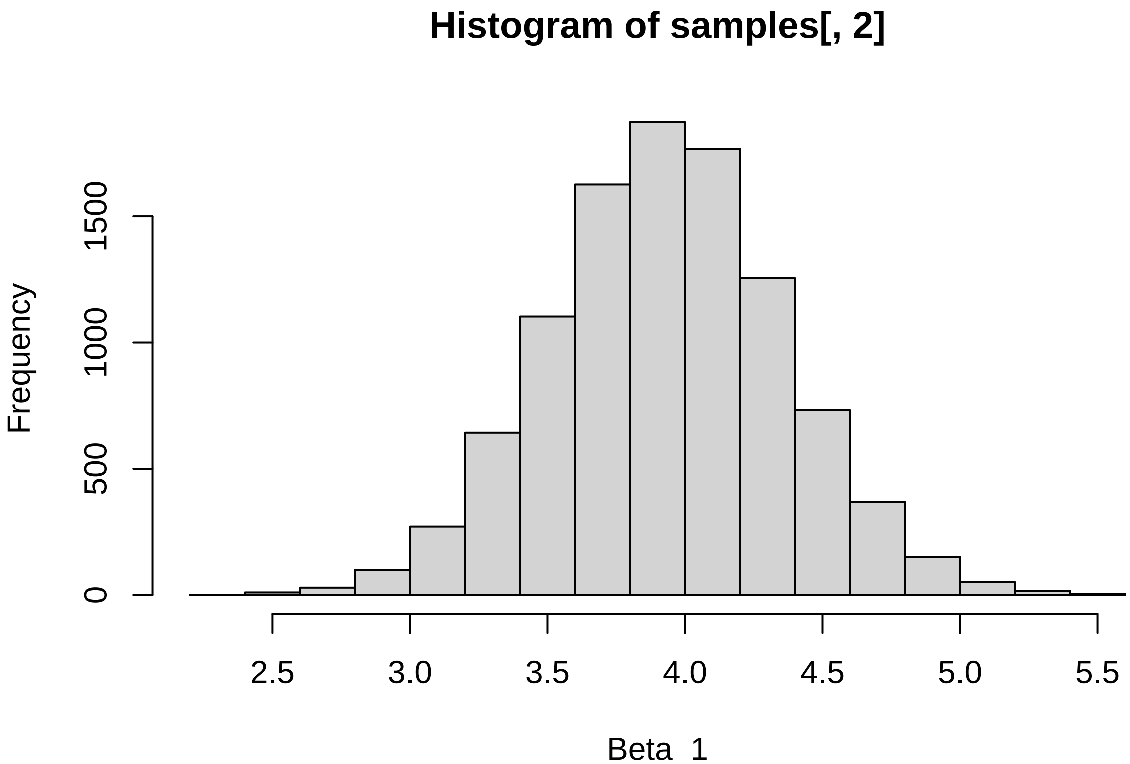

beta_0
0
−10
−20
−30
−40
0
2000
4000
6000
8000
10000
Iteration

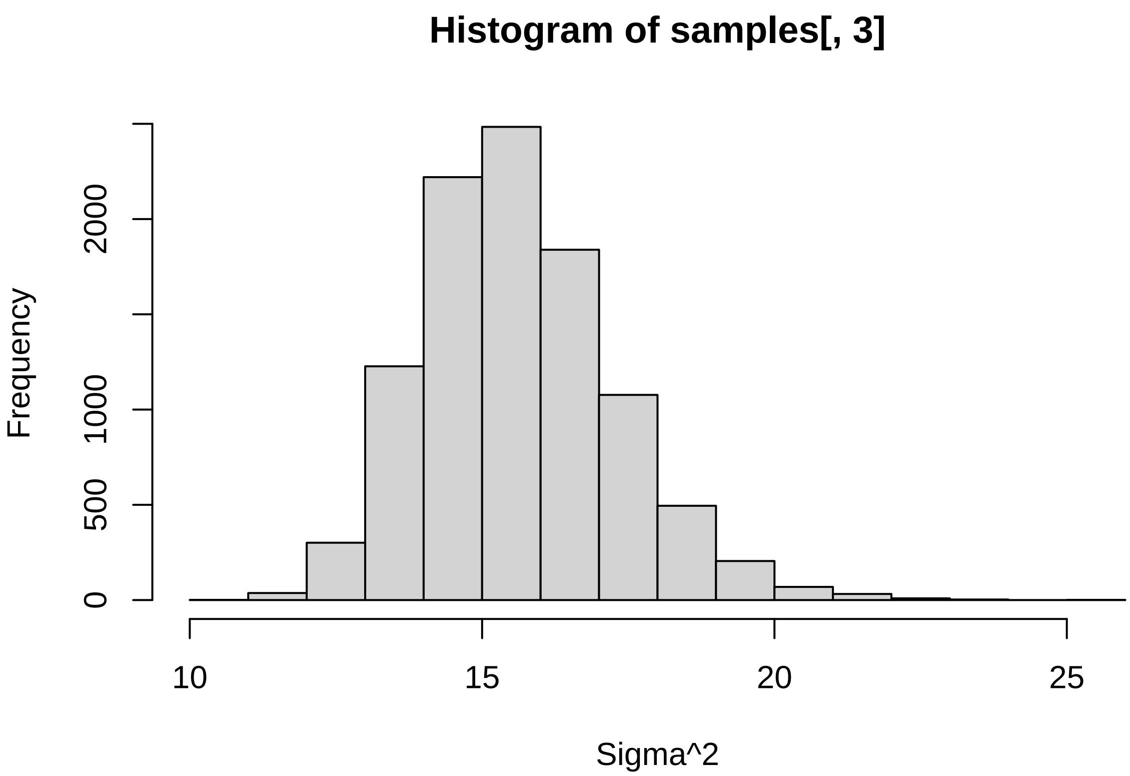

beta_1
5.5
5.0
4.5
4.0
3.5
3.0
2.5
0
2000
4000
6000
8000
10000
Iteration


σ2
25
20
15
0
2000
4000
6000
8000
10000
Iteration

(c)


Histogram of samples[, 1]
Frequency
0
500
1500
2500
−50
−40
−30
−20
−10
0
10
Beta_0





(d)

The Bayesian estimates for the parameters are extremely similar to those obtained via least squares / maximum likelihood estimation, only being off by less than 0.5 for all parameters.

```
##                            beta_0    beta_1   sigma^2
```

```
## Post. mean:          -17.547296 3.9315323 15.614263
## Post. std dev:         6.913664 0.4252464  1.643271
## 2.5th Percentile:   -31.216300 3.1072008 12.818886
## 97.5th Percentile:   -3.972887 4.7804577 19.242605
## Least Squares est.: -17.579095 3.9324088 15.221843
```

(e)

There is a negative correlation between β_0 and β_1. (i.e., as β_0 increases, β_1 decreases.)