

# STAT 578 - Advanced Bayesian Modeling - Fall 2019

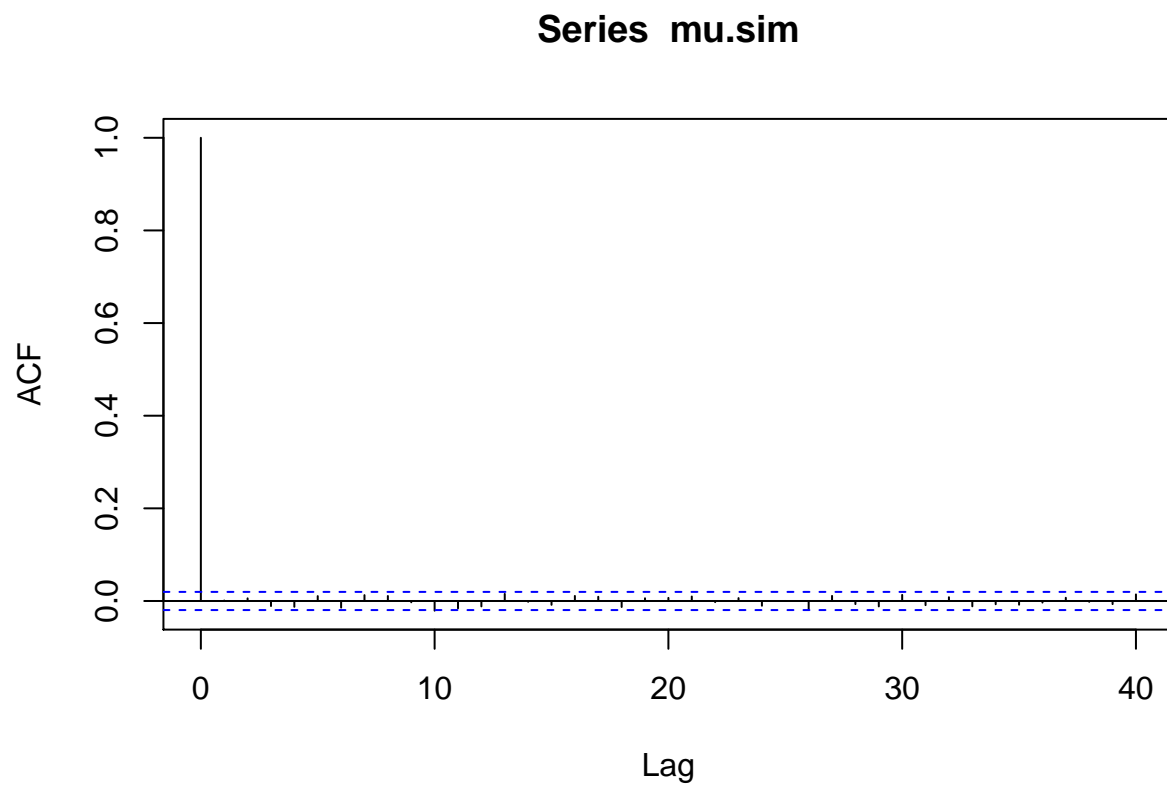
## Assignment 3

*Xiaoming Ji*

### Solution for Problem 1

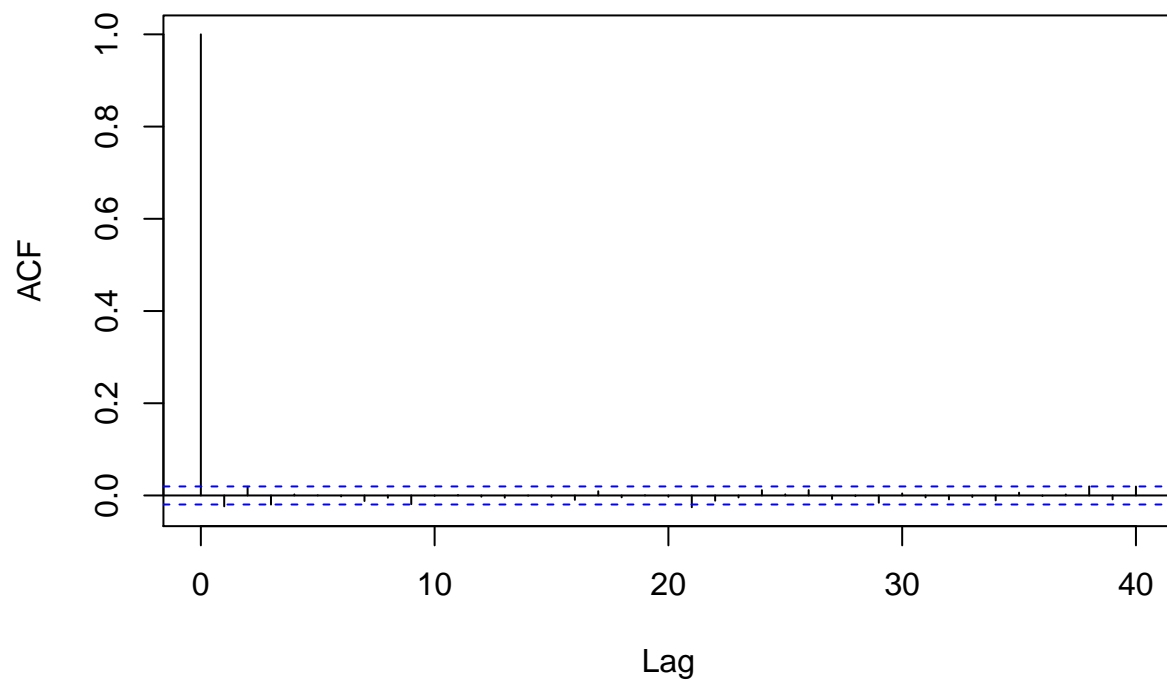
(a)

```
source("FlintGibbs.R")  
acf(mu.sim)
```



```
acf(sigma.2.sim)
```

### Series sigma.2.sim



(b)

(i)

```
rho <- 0.03  
source("FlintMetropolis.R")
```

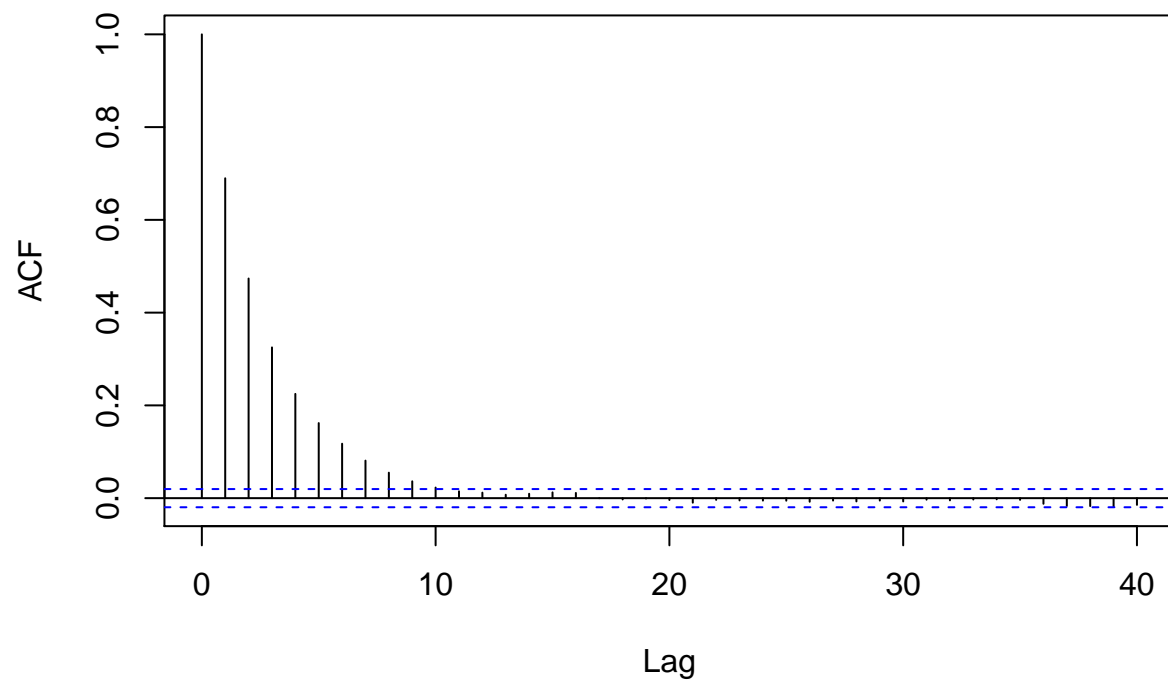
```
## [1] 0.3522545
```

$\rho$  takes 0.03 will give acceptance rate of about 0.35.

(ii)

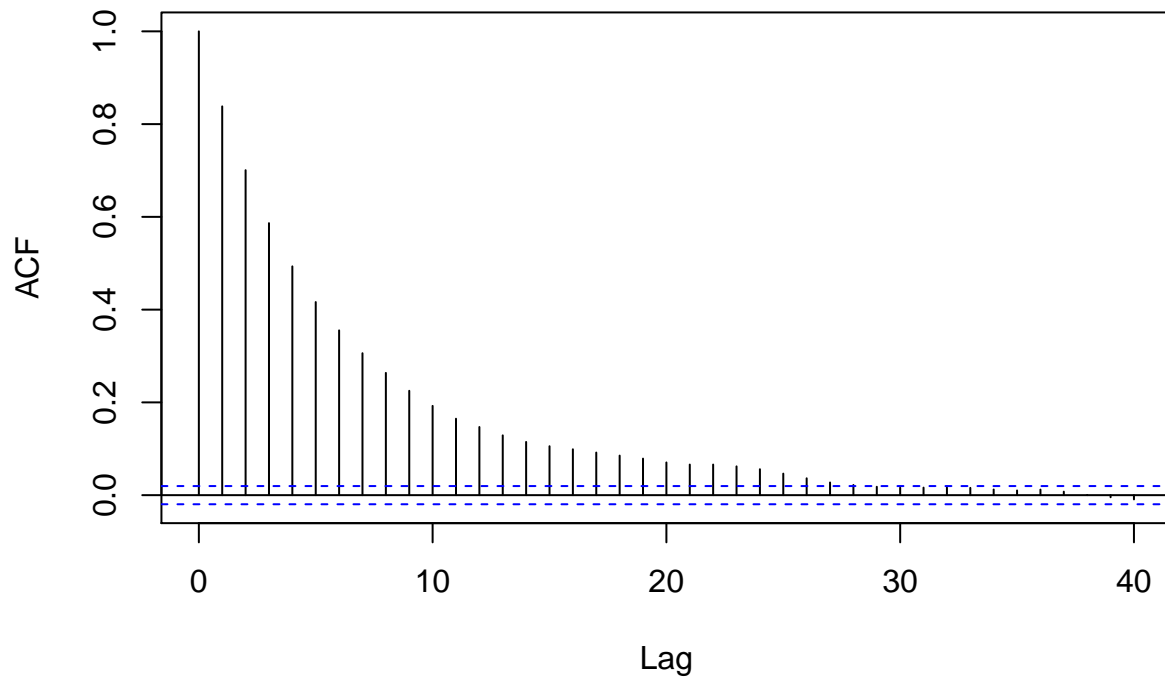
```
acf(mu.sim)
```

### Series mu.sim



```
acf(sigma.2.sim)
```

## Series sigma.2.sim



(c)

The autocorrelation plot for Gibbs sampler decays much faster than Metropolis sampler's. Thus, Gibbs sampler exhibited faster mixing.

## Solution for Problem 1

(a)

```
library(rjags)

## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
polls2016_df <- read.table("polls2016.txt", header=TRUE)
polls2016_df$sigma <- polls2016_df$ME/2
polls2016_df <- subset(polls2016_df, select = -c(poll, ME) )
```

(i)

```
initial_vals <- list(list(mu = 100, tau = 100,
                          .RNG.name="base::Wichmann-Hill", .RNG.seed=123),
```

```

list(mu = 100, tau = 0.01,
     .RNG.name="base::Wichmann-Hill", .RNG.seed=123),
list(mu = -100, tau = 100,
     .RNG.name="base::Wichmann-Hill", .RNG.seed=123),
list(mu = -100, tau = 0.01,
     .RNG.name="base::Wichmann-Hill", .RNG.seed=123))

poll_model <- jags.model("polls20161.bug", polls2016_df, initial_vals, n.chains = 4)

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 7
##   Unobserved stochastic nodes: 9
##   Total graph size: 42
##
## Initializing model

```

(ii)

```

update(poll_model, 2500)
x <- coda.samples(poll_model, c("mu", "tau"), n.iter = 5000)

```

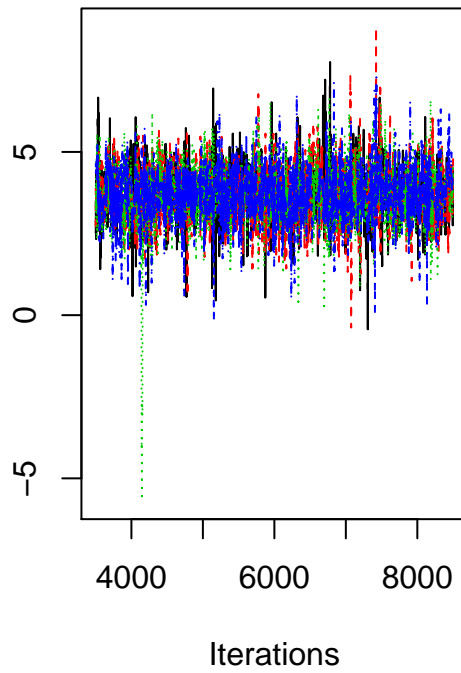
(iii)

```

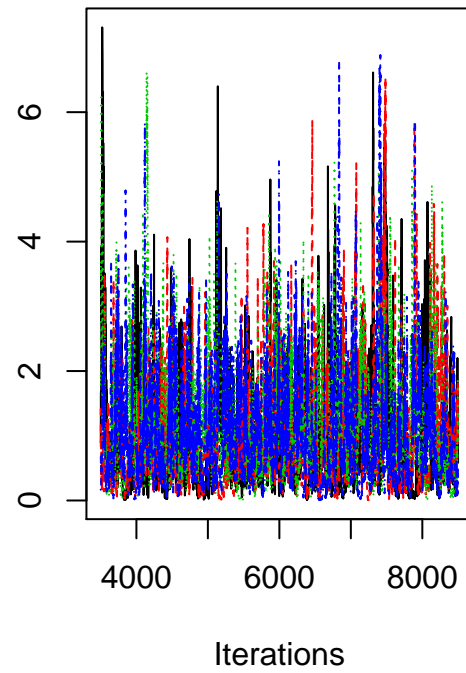
plot(x, smooth=FALSE, density = FALSE)

```

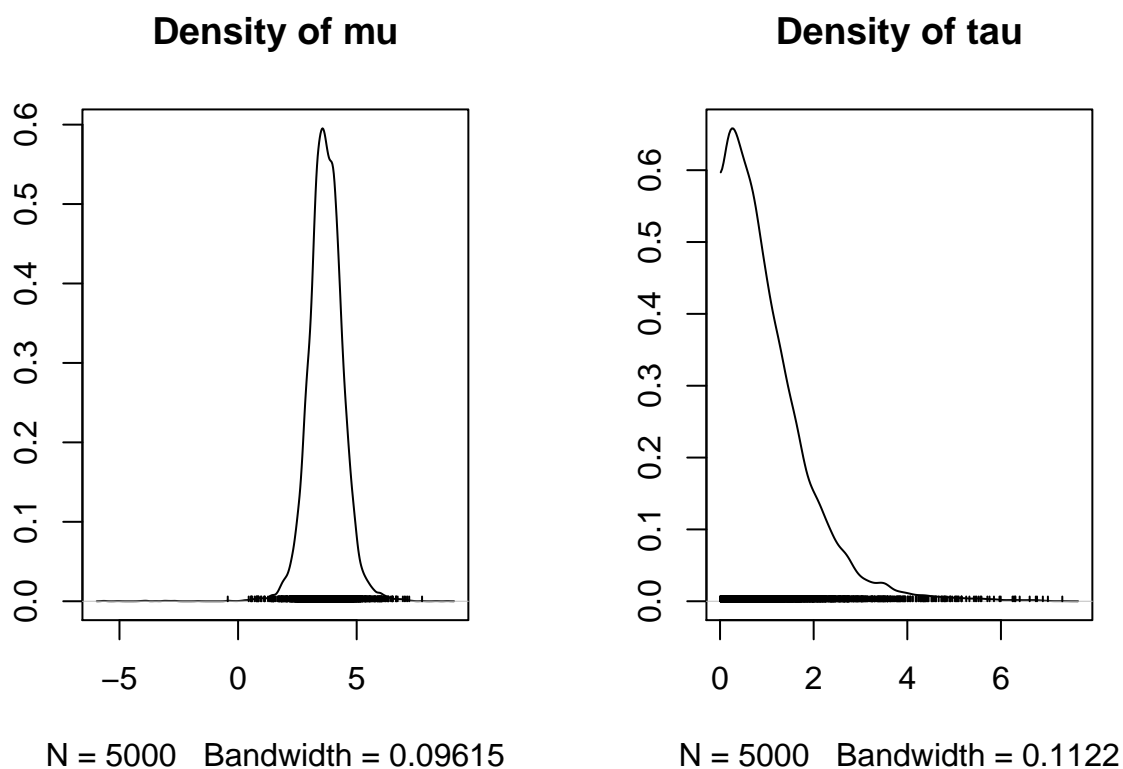
**Trace of mu**



**Trace of tau**



```
plot(x, smooth=FALSE, trace = FALSE)
```



The trace plot shows 4 chains for mu and tau span the similar region and we can't observe obvious convergence problem.

(iv)

```
gelman.diag(x, autoburnin=FALSE)
```

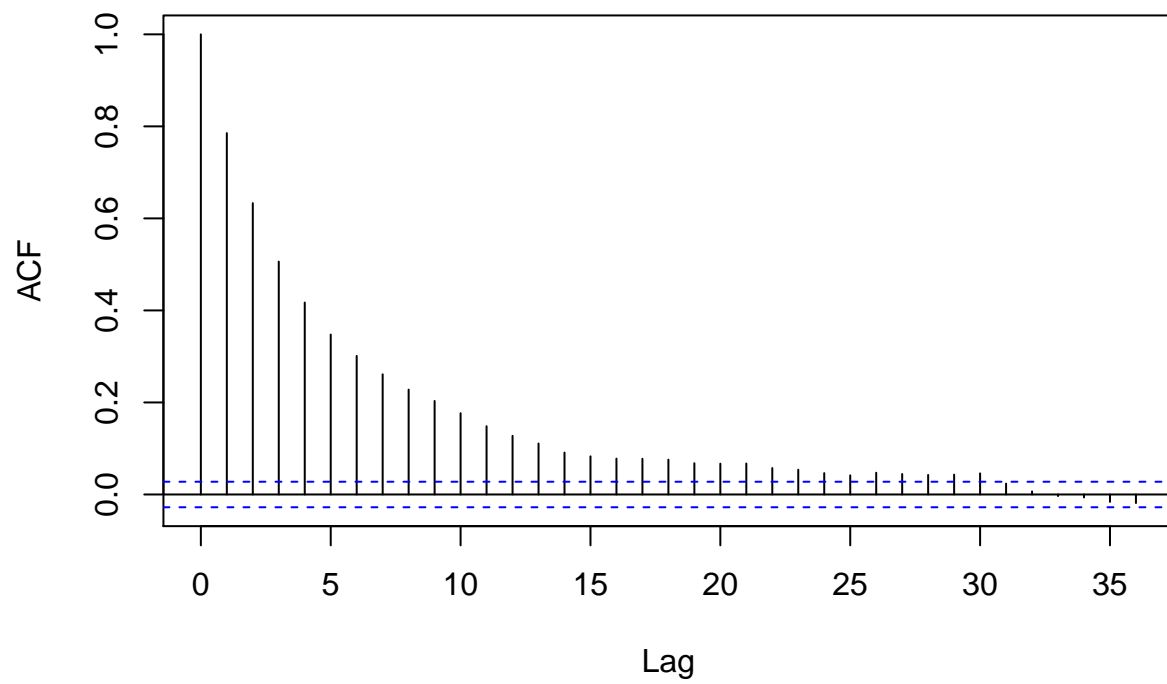
```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## mu      1.00      1.00
## tau      1.01      1.02
##
## Multivariate psrf
##
## 1.01
```

Gelman-Rubin statistics (Potential scale reduction factor) for mu and tau are close to 1 with upper confidence limits close to 1, thus there don't appear to have any convergence problem.

(v)

```
mu_chain_1 = x[[1]][,1]
tau_chain_1 = x[[1]][,2]
acf(mu_chain_1)
```

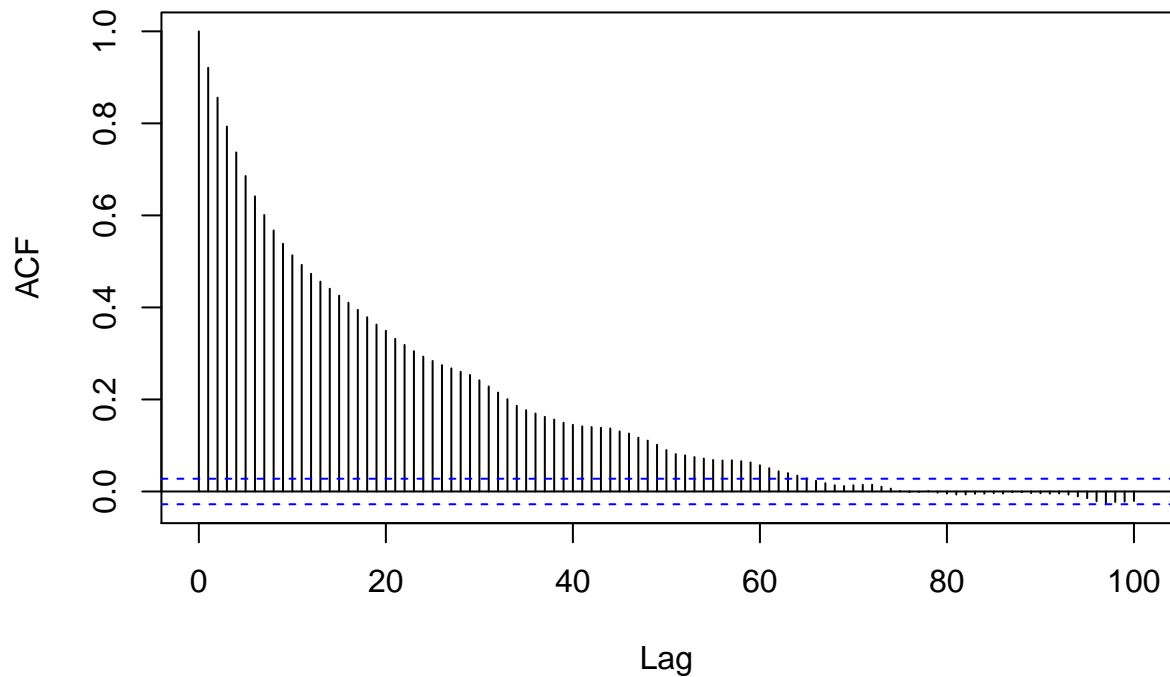
### Series mu\_chain\_1



```
acf(tau_chain_1, lag.max = 100)
```



## Series tau\_chain\_1



We see autocorrelation (of Chain 1) goes zero for  $\mu$  by lag around 30 and for  $\tau$  by lag around 70. Thus mixing of  $\mu$  is faster than  $\tau$ 's.

(vi)

```
effectiveSize(x)
```

```
##      mu      tau
## 2041.3237 939.8571
```

The effective sample size for  $\mu$  and  $\tau$  is less than 2500, thus our sample size of 5000 is adequate.

(b)

(i)

```
model {
  for (j in 1:length(y)) {
    y[j] ~ dnorm(theta[j], 1/sigma[j]^2)
    theta[j] ~ dnorm(mu, 1/tau^2)
  }

  mu ~ dunif(-1000,1000)
  logtau ~ dunif(-100, 100)
  tau <- exp(logtau)
}
```

(ii)

```
initial_vals_new <- list(list(mu = 100, logtau = log(100),  
                             .RNG.name="base::Wichmann-Hill", .RNG.seed=123),  
                          list(mu = 100, logtau = log(0.01),  
                             .RNG.name="base::Wichmann-Hill", .RNG.seed=123),  
                          list(mu = -100, logtau = log(100),  
                             .RNG.name="base::Wichmann-Hill", .RNG.seed=123),  
                          list(mu = -100, logtau = log(0.01),  
                             .RNG.name="base::Wichmann-Hill", .RNG.seed=123))  
  
poll_model_new <- jags.model("polls20161_new.bug", polls2016_df, initial_vals_new, n.chains = 4)  
  
## Compiling model graph  
##   Resolving undeclared variables  
##   Allocating nodes  
## Graph information:  
##   Observed stochastic nodes: 7  
##   Unobserved stochastic nodes: 9  
##   Total graph size: 44  
##  
## Initializing model
```

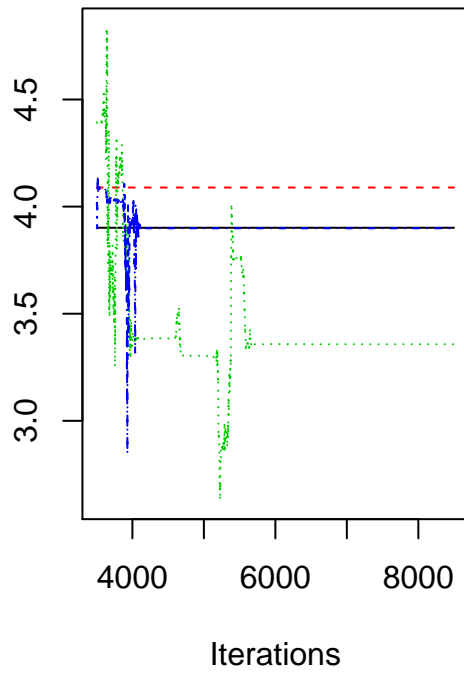
(iii)

```
update(poll_model_new, 2500)  
x_new <- coda.samples(poll_model_new, c("mu", "tau"), n.iter = 5000)
```

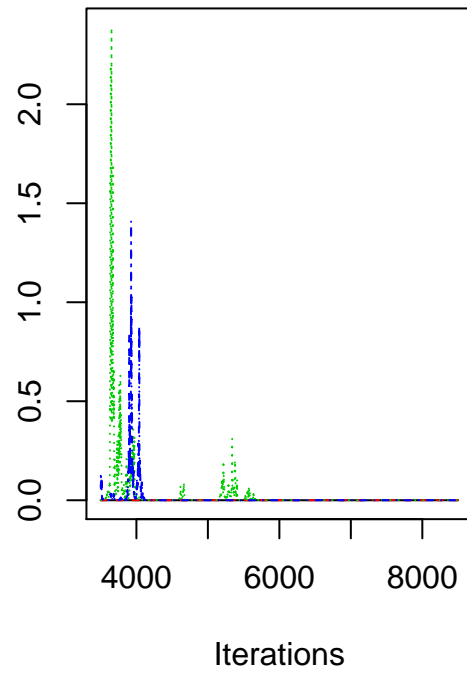
(iv)

```
plot(x_new, smooth=FALSE, density = FALSE)
```

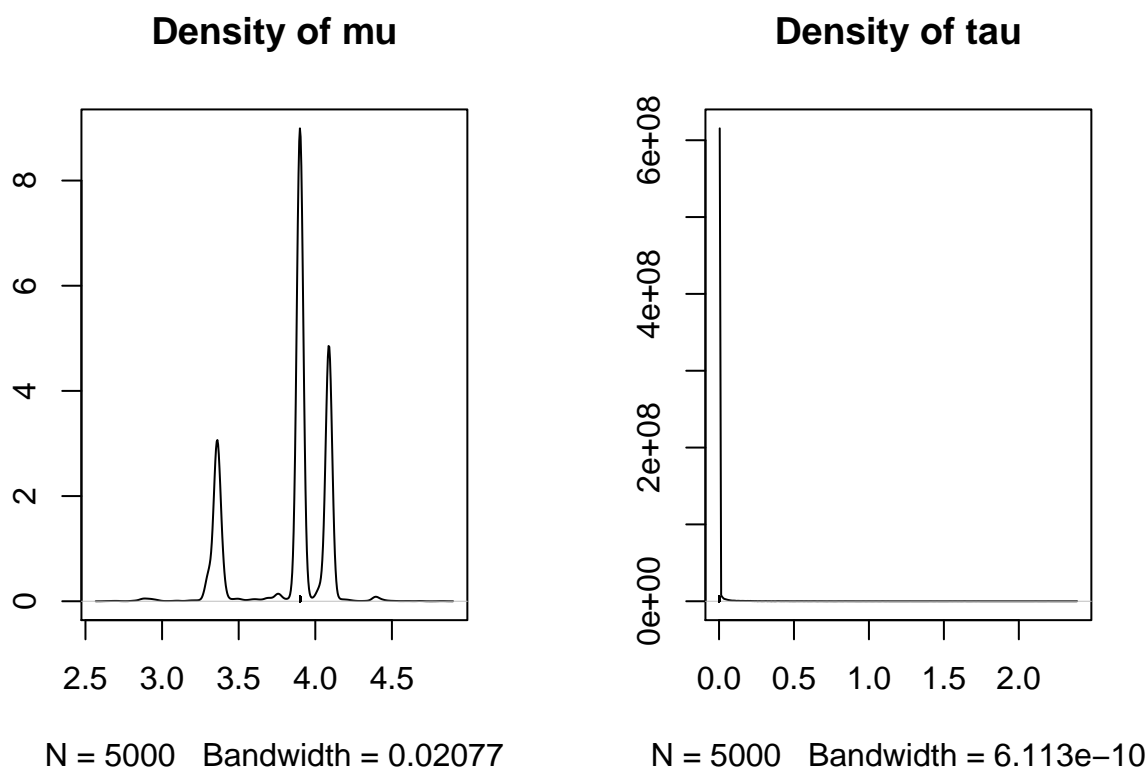
**Trace of mu**



**Trace of tau**



```
plot(x_new, smooth=FALSE, trace = FALSE)
```



The trace plots for mu and tau clearly show that the values range covered by 4 chains are quite different, thus this model *don't converge* on mu and tau based on these plots.

(v)

```
gelman.diag(x_new, autoburnin=FALSE)
```

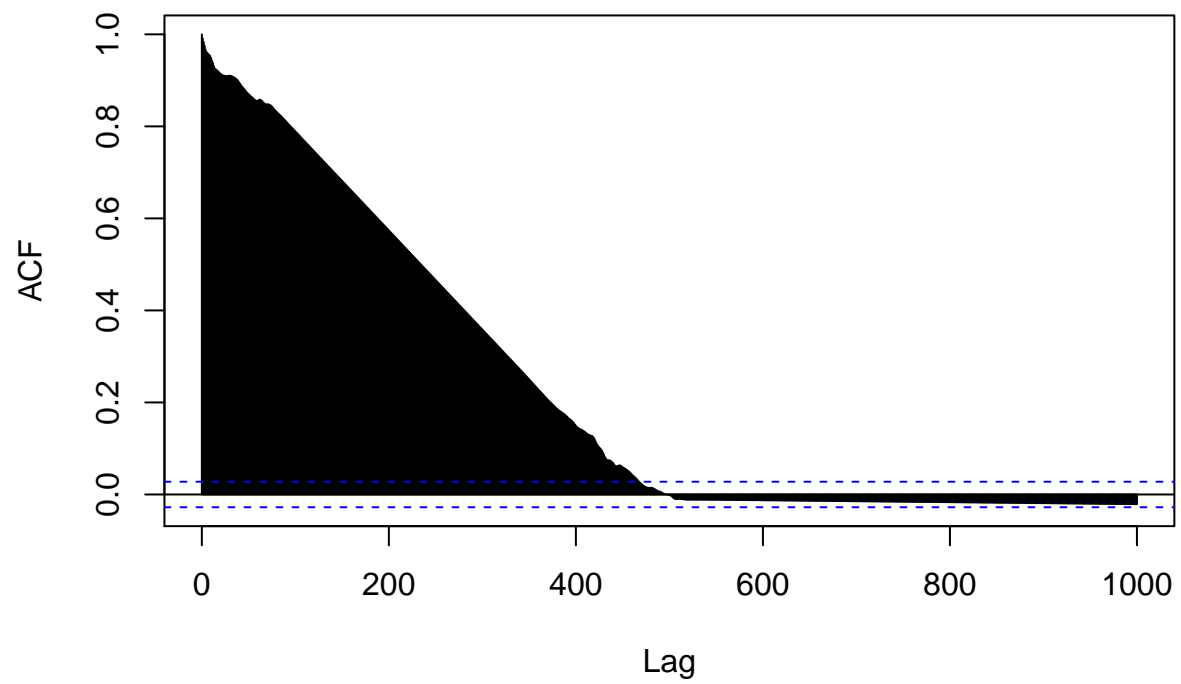
```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## mu      3.20      14.99
## tau      1.22      1.41
##
## Multivariate psrf
##
## 2.99
```

Gelman-Rubin statistics (Potential scale reduction factor) for mu and tau are all greater than 1.1, thus this model *don't converge* based on the Gelman-Rubin statistics.

(vi)

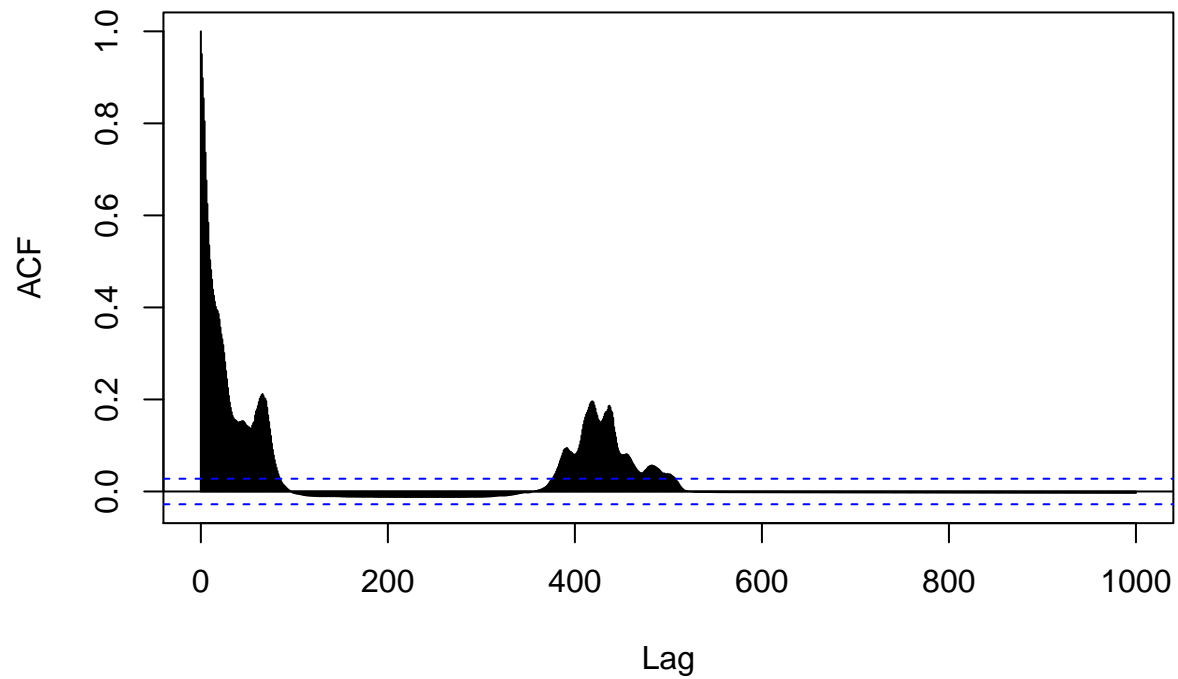
```
mu_chain_1_new = x_new[[1]][,1]
tau_chain_1_new = x_new[[1]][,2]
acf(mu_chain_1_new, lag.max = 1000)
```

### Series mu\_chain\_1\_new



```
acf(tau_chain_1_new, lag.max = 1000)
```

### Series tau\_chain\_1\_new



autocorrelation (of Chain 1) goes zero for mu by lag around 500 and for tau by lag around 100. Thus mixing of mu is slower than tau's.

(vii)

The fundamental problem here is: the chosen of improper flat prior for logtau ( $\text{logtau} \sim U(-100, 100)$ ) gives an improper posterior. When we choose the prior distribution, we should always check whether this prior can give us a proper posterior even the JAGS can run without any error. In our case, the simulation doesn't converge and gives any meaningful results.