

Stat251 Final Project

Emily Liu

3/25/2022

Introduction

Alzheimer's is a brain disease where cells degenerate and cause memory loss. 40 million people worldwide suffer from this disease and a cure does not exist. Although there is no definitive cause of Alzheimer's, scientists speculate that genetics, aging, and environmental influences may affect the probability of developing the disease. Some specialists have found that the larger the brain, the more the brain may combat against the effects of cognitive atrophy. The total intracranial volume (TIV) is a way to quantify the size of the brain. TIV includes the volume of the cranium, brain, and spinal fluid. To discover how brain size relates to Alzheimer's, we will investigate the average TIV for patients 60 years and older of those with and without the disease.

```
# filepath <- paste0('~/Documents/Winter 2022/Stat251/alzheimer.csv')
filepath <- 'alzheimer.csv'
alz <- read.csv(filepath, stringsAsFactors = TRUE)
```

```
# measurements of total intracranial volume
demented <- alz %>% filter(Group == 'Demented')
nondemented <- alz %>% filter(Group == 'Nondemented')
```

```
# plots
ggplot(demented, aes(x = eTIV)) +
  geom_histogram() +
  theme_bw() +
  labs(title = 'Histogram of TIV for Demented Patients')
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

```
ggplot(nondemented, aes(x = eTIV)) +
  geom_histogram() +
  theme_bw() +
  labs(title = 'Histogram of TIV for Nondemented Patients')
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

Likelihood: $x_i|\mu, \sigma^2 \sim N(\mu, \sigma^2)$, $i = 1, \dots, n$

3. Priors for demented population: $\mu_D \sim N(\lambda, \tau_D^2)$

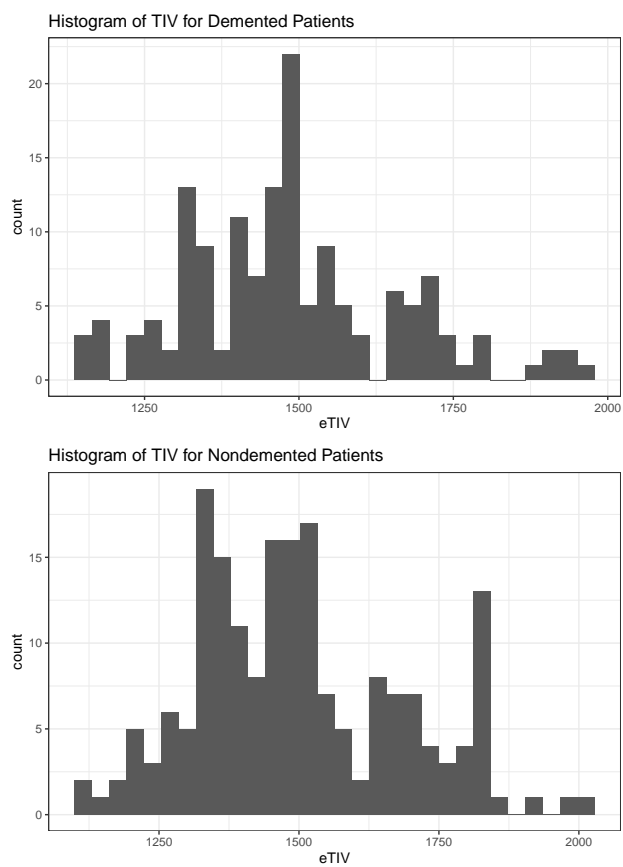


Figure 1: Figure 1

$$\sigma_D \sim IG(\gamma, \phi)$$

Priors for non-demented population: $\mu_N \sim N(\lambda, \tau\tau^2)$

$$\sigma_N \sim IG(\gamma, \phi)$$

4.

Data exploration

```
# Prior for mu
prior_mean <- 1500 # lambda
prior_var <- 31000 # Tau^2

# Prior for sigma2
prior_mean_sigma2 <- 1
prior_var_sigma2 <- 1
```

5.

```
demented <- alz %>% filter(Group == 'Demented') %>% select(eTIV)

# Find the posterior of mu using Gibbs sampling.
n <- length(demented)
# Starting values (This example has extreme starting values just for illustrative purposes)
lambda <- 1500 # Prior Mean for mu
tau2 <- 31000 # Prior Variance for mu
sigma2 <- 1 # Sample Variance
mu <- 1500 #????

gamma <- 1 # Prior Gamma for sigma2
phi <- 1 # Prior Phi for sigma2

# initializations for the Gibbs Sampling Algorithm
iters <- 10000
mu.save <- rep(0, iters)
mu.save[1] <- mu
sigma2.save <- rep(0, iters)
sigma2.save[1] <- sigma2

#Gibbs Sampling Algorithm
for(t in 2:iters){
  # Full conditional of mu (update the value of the parameters)
  lambda.p <- (tau2*sum(demented) + sigma2*lambda)/(tau2*n + sigma2) #posterior mean of mu given data a
  tau2.p <- sigma2*tau2/(tau2*n + sigma2) # posterior variance

  #sample a new value of mu from its full conditional
  mu <- rnorm(1, lambda.p, sqrt(tau2.p))

  #save the value of mu
```

```

mu.save[t] <- mu

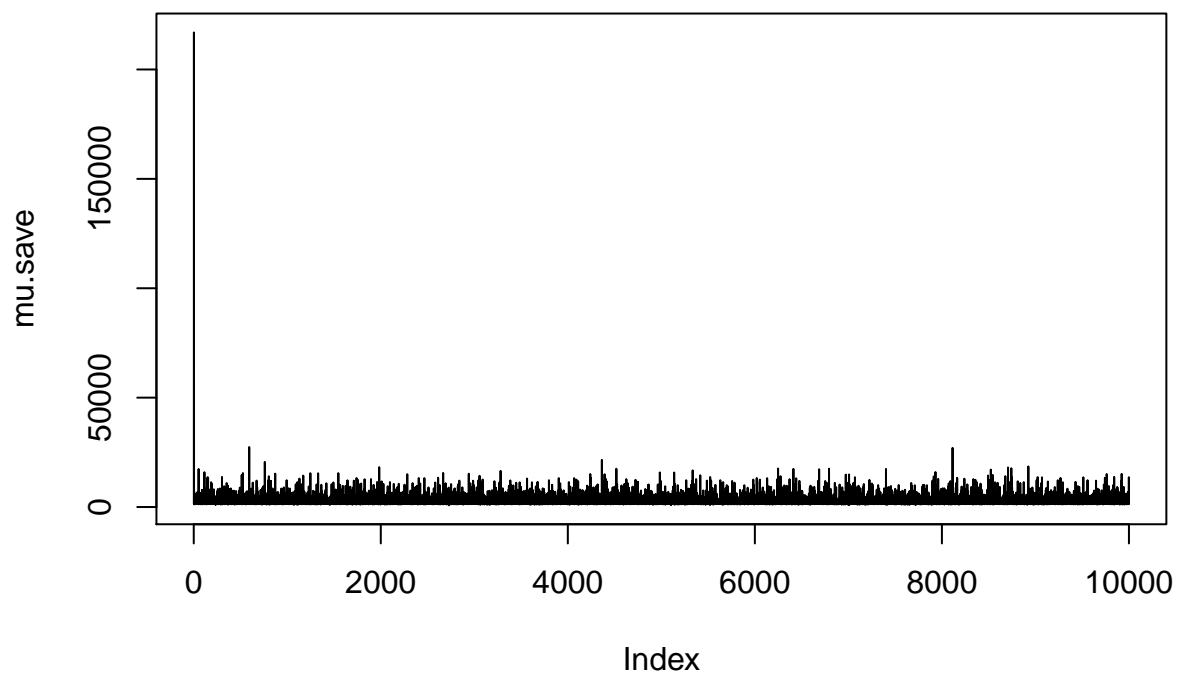
# full conditional of sigma2 (update the value of the parameters)
gamma.p <- gamma + n/2
phi.p <- phi + sum((demented - mu)^2)/2

#sample new value of sigma2 from its full conditional
sigma2 <- rinvgamma(1, gamma.p, phi.p)

#save the value of sigma2
sigma2.save[t] <- sigma2
}

# Trace plots (decide if we need to throw out the first few values)
plot(mu.save, type='l')

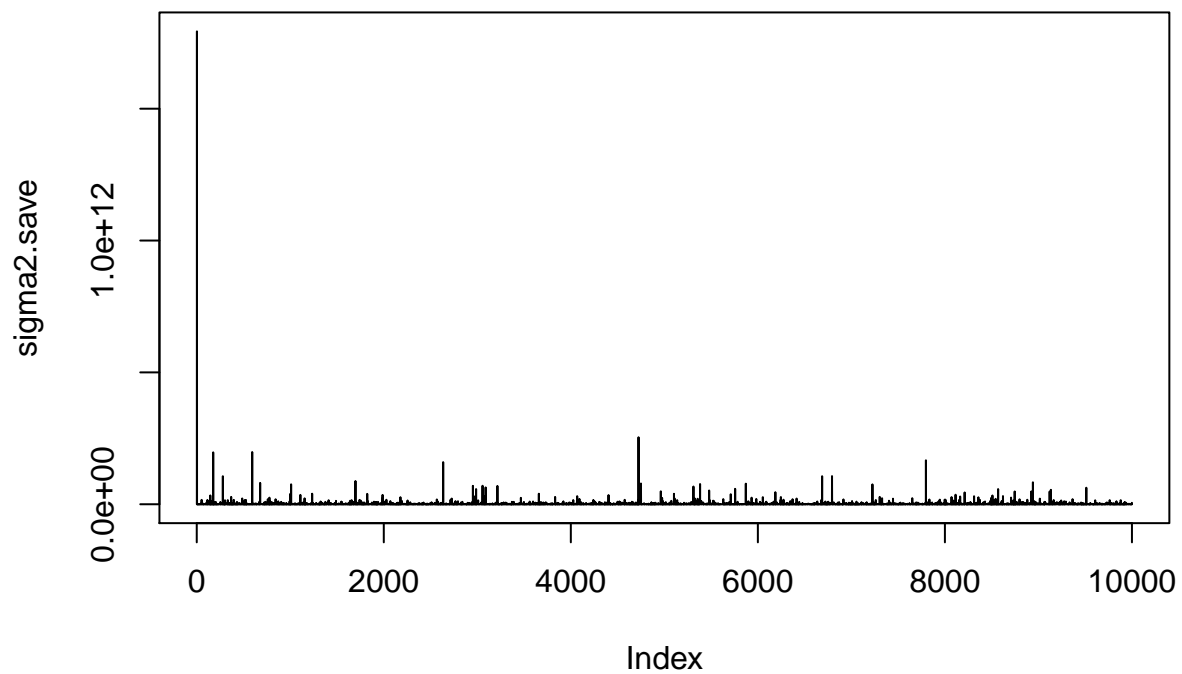
```



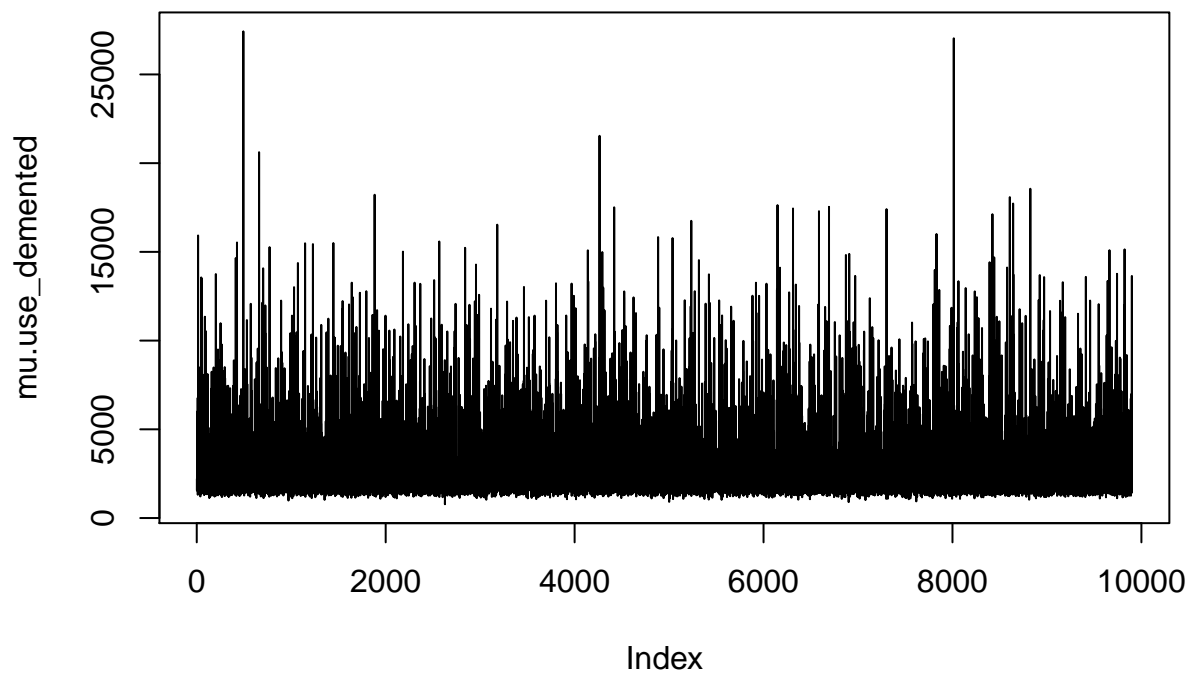
```

plot(sigma2.save, type='l')

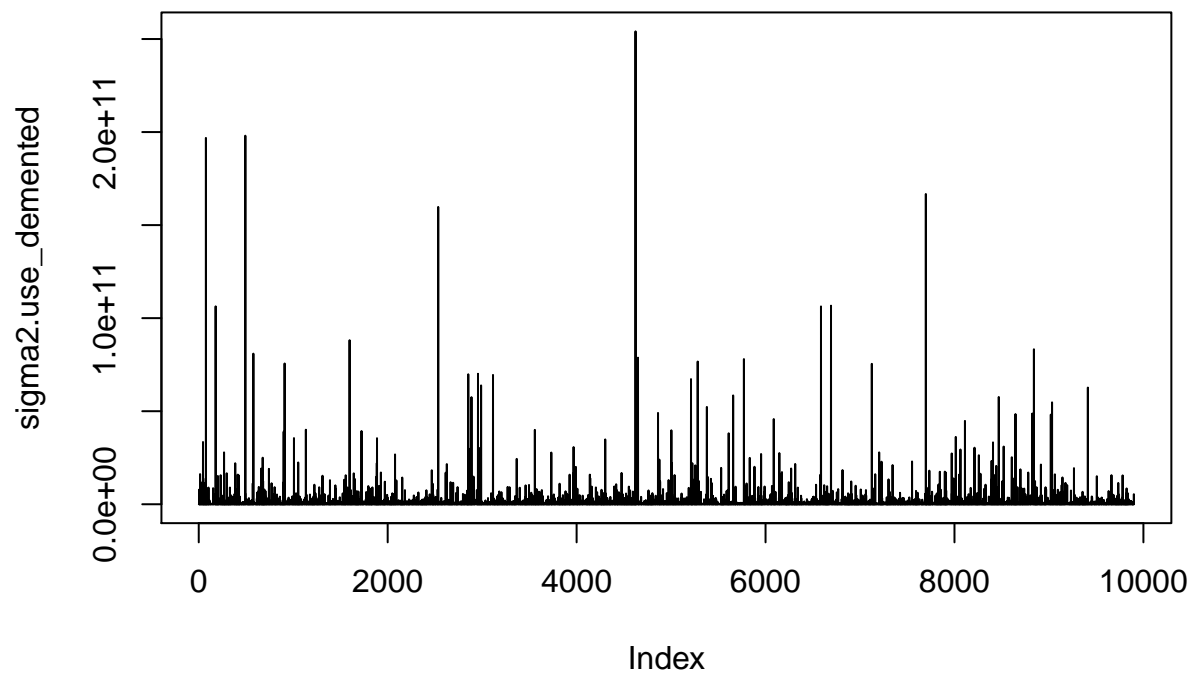
```



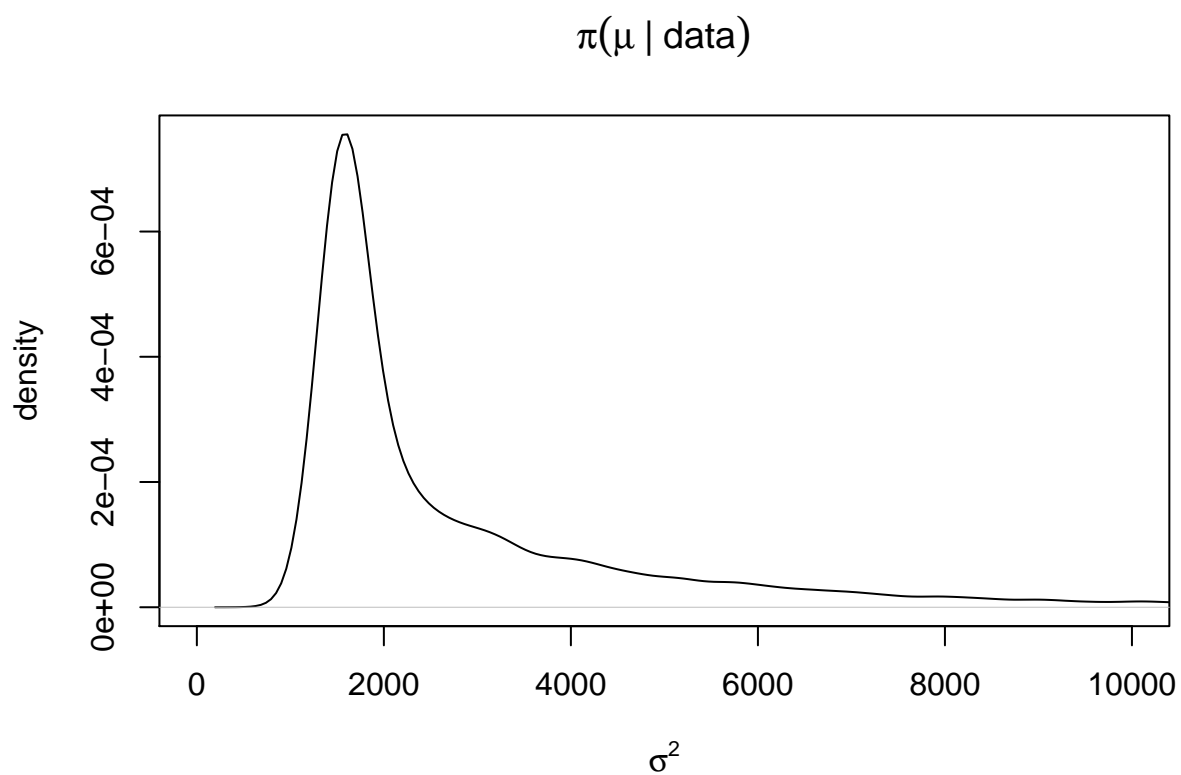
```
#throw out the first few values  
burn <- 100  
mu.use_demented <- mu.save[-(1:burn)]  
sigma2.use_demented <- sigma2.save[-(1:burn)]  
plot(mu.use_demented, type='l')
```



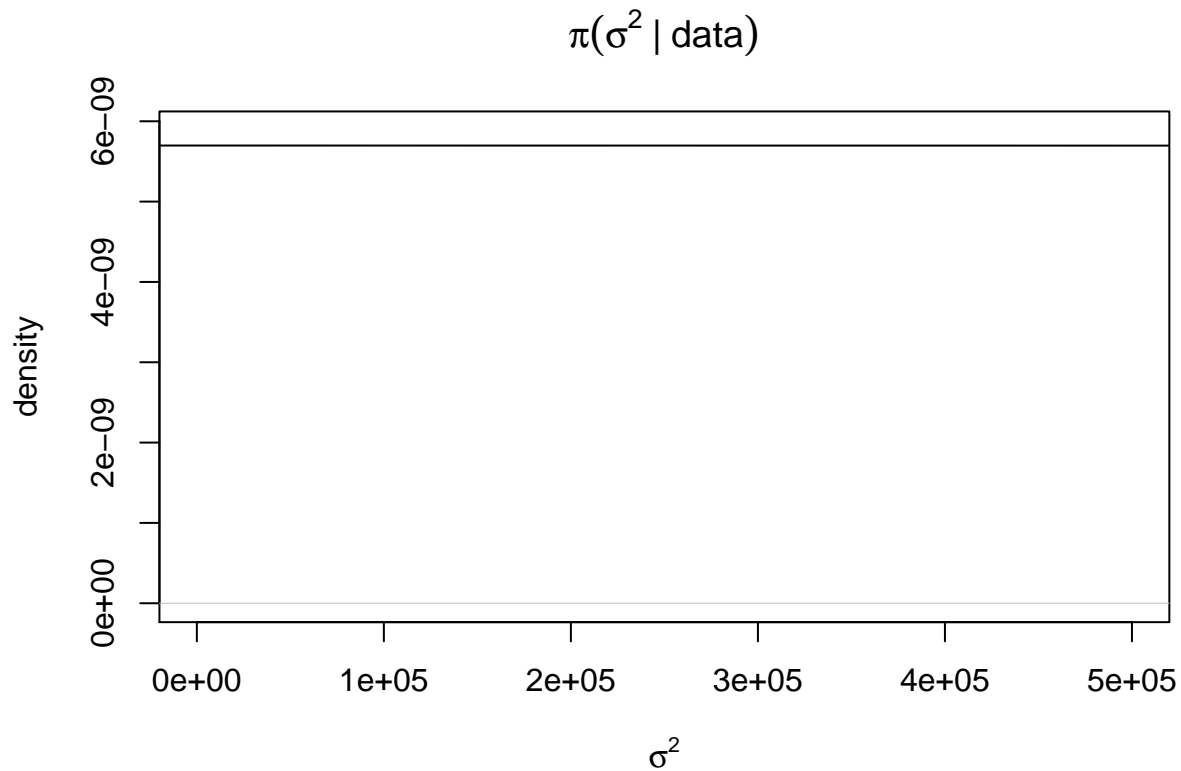
```
plot(sigma2.use_demented, type='l')
```



```
# marginal posterior distribution of mu (NOT conditional on sigma2)
plot(density(mu.use_demented), xlim=c(0, 10000), xlab=expression(sigma^2),
     ylab="density", main=expression(pi(mu~"|"~data)))
```



```
# marginal posterior distribution of sigma2 (NOT conditional on sigma2)
plot(density(sigma2.use_demented), xlim=c(0, 50000), xlab=expression(sigma^2),
     ylab="density", main=expression(pi(sigma^2~"|"~data)))
```

```
nondemented <- alz %>% filter(Group == 'Nondemented') %>% select(eTIV)

# Find the posterior of mu using Gibbs sampling.
n <- length(nondemented)
# Starting values (This example has extreme starting values just for illustrative purposes)
lambda <- 1500 # Prior Mean for mu
tau2 <- 31000 # Prior Variance for mu
sigma2 <- 1 # Sample Variance

gamma <- 1 # Prior Gamma for sigma2
phi <- 1 # Prior Phi for sigma2

# initializations for the Gibbs Sampling Algorithm
iters <- 10000
mu.save <- rep(0, iters)
mu <- 1500
mu.save[1] <- mu
sigma2.save <- rep(0, iters)
sigma2.save[1] <- sigma2

#Gibbs Sampling Algorithm
for(t in 2:iters){
  # Full conditional of mu (update the value of the parameters)
  lambda.p <- (tau2*sum(nondemented) + sigma2*lambda)/(tau2*n + sigma2) #posterior mean of mu given data
```

```

tau2.p <- sigma2*tau2/(tau2*n + sigma2) # posterior variance

#sample a new value of mu from its full conditional
mu <- rnorm(1, lambda.p, sqrt(tau2.p))

#save the value of mu
mu.save[t] <- mu

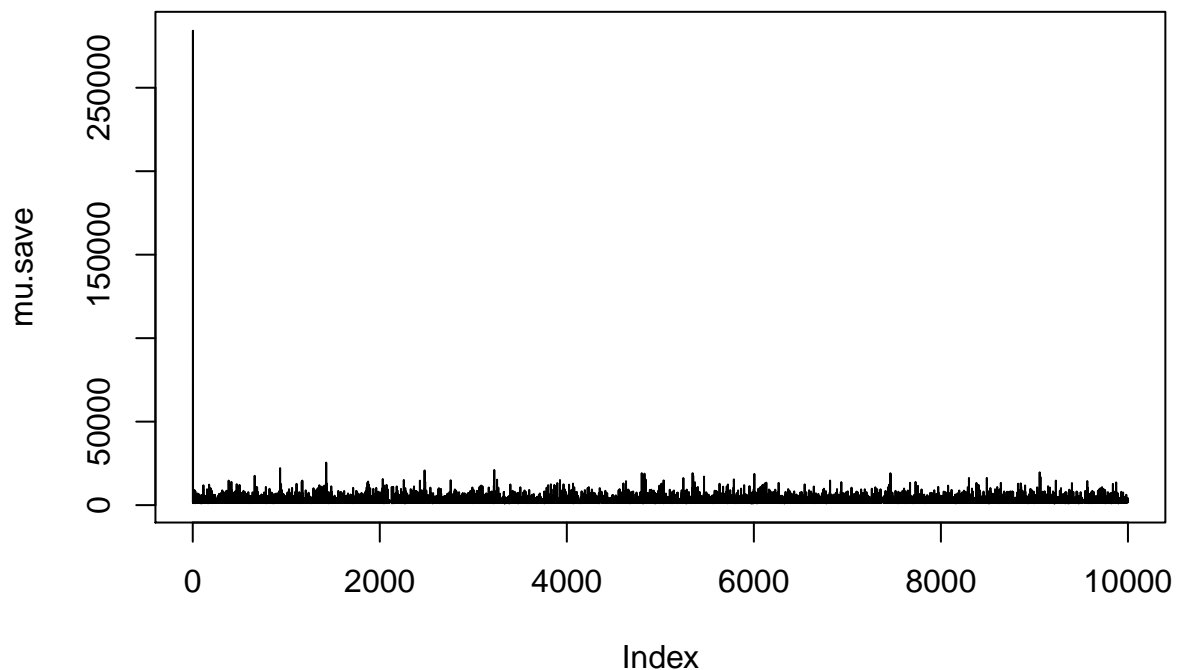
# full conditional of sigma2 (update the value of the parameters)
gamma.p <- gamma + n/2
phi.p <- phi + sum((nondemented - mu)^2)/2

#sample new value of sigma2 from its full conditional
sigma2 <- rinvgamma(1, gamma.p, phi.p)

#save the value of sigma2
sigma2.save[t] <- sigma2
}

# Trace plots (decide if we need to throw out the first few values)
plot(mu.save, type='l')

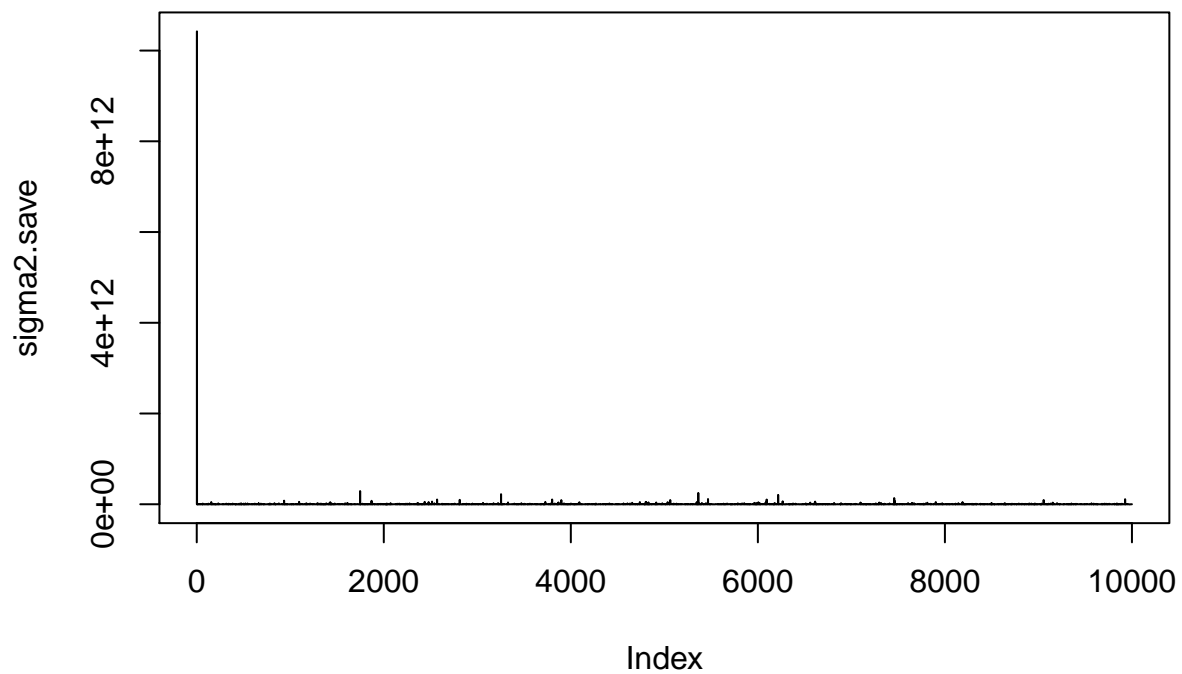
```



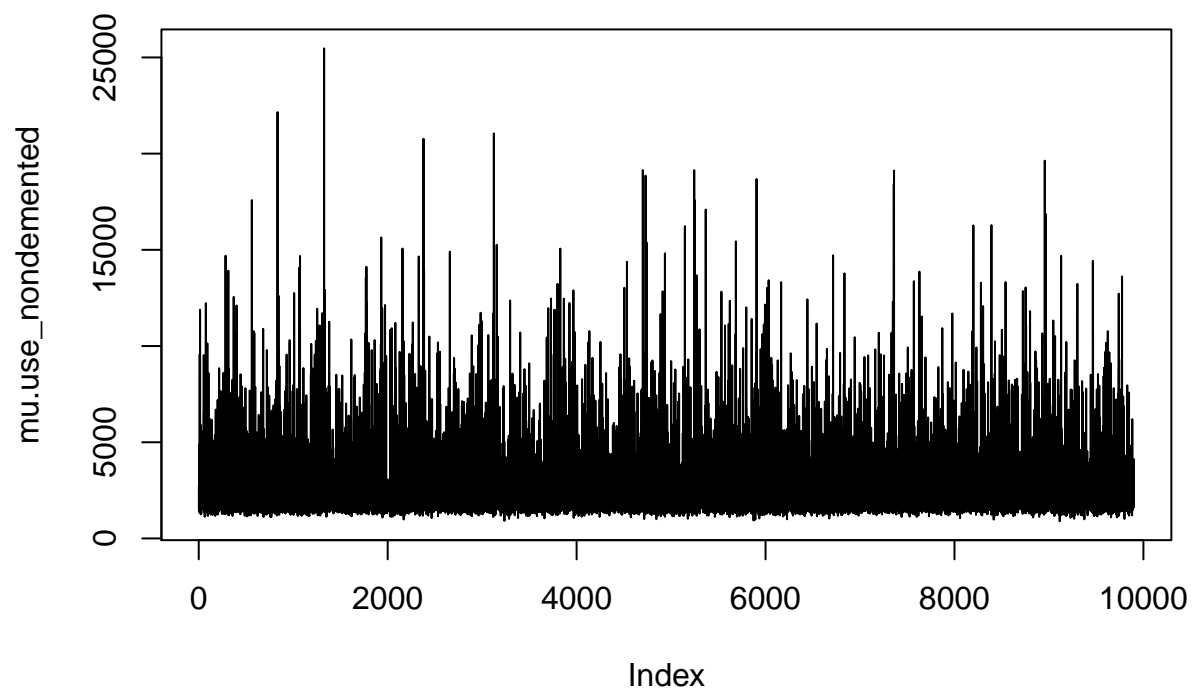
```

plot(sigma2.save, type='l')

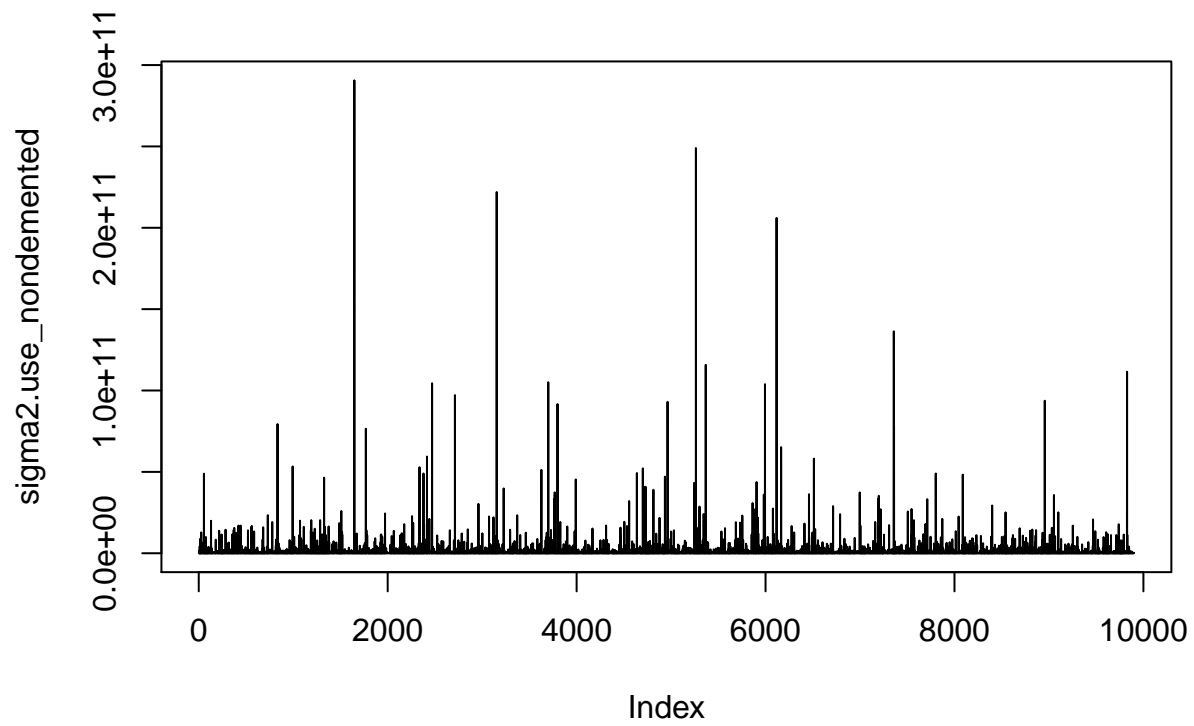
```



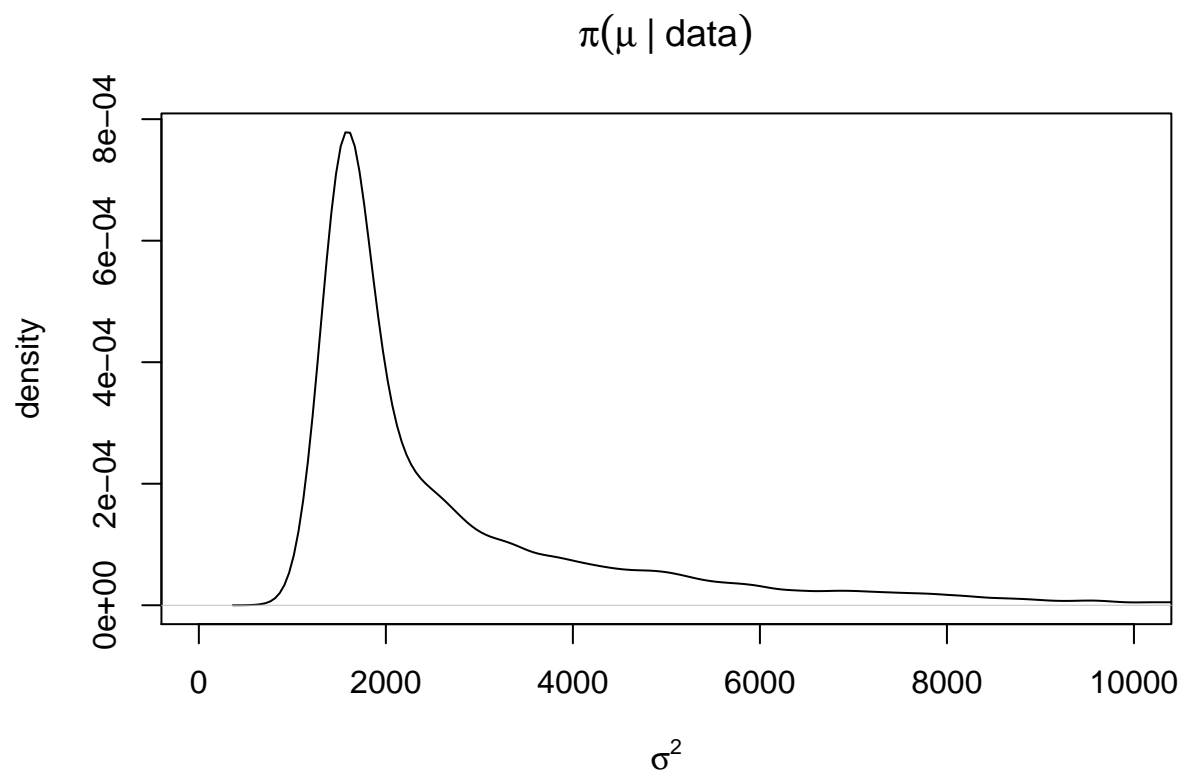
```
#throw out the first few values  
burn <- 100  
mu.use_nondemented <- mu.save[-(1:burn)]  
sigma2.use_nondemented <- sigma2.save[-(1:burn)]  
plot(mu.use_nondemented, type='l')
```



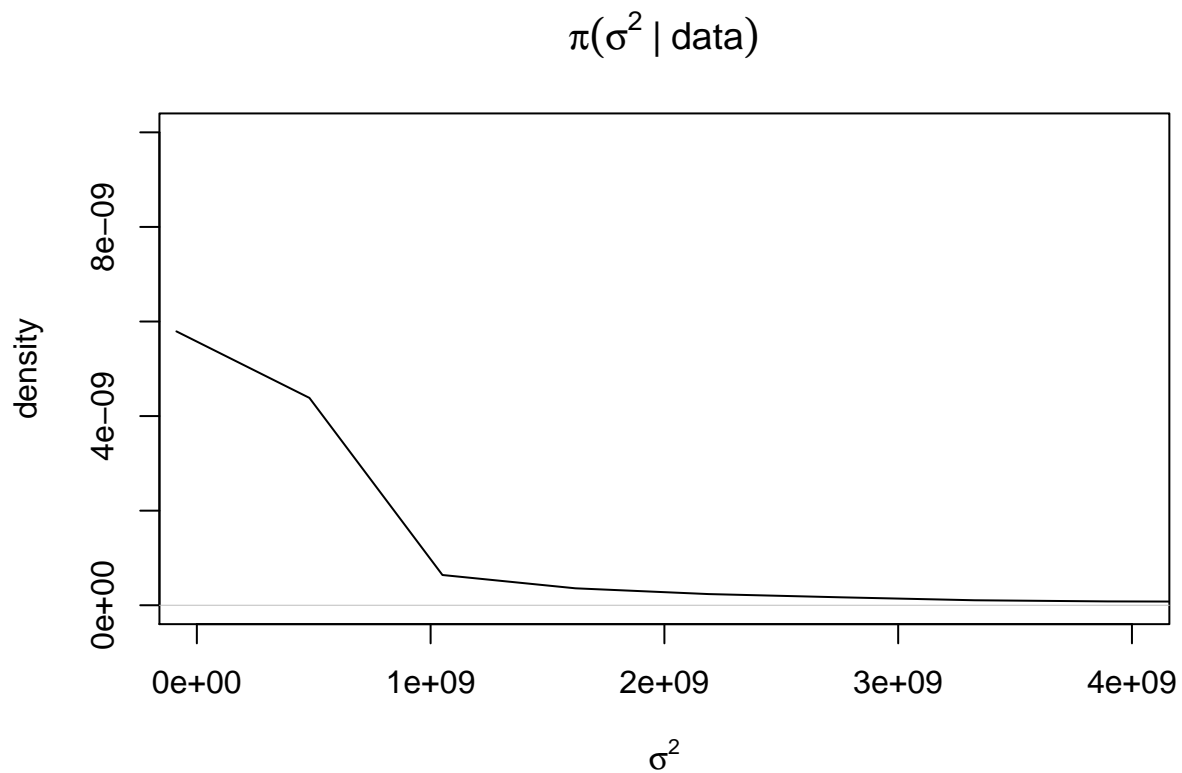
```
plot(sigma2.use_nondemented, type='l')
```



```
# marginal posterior distribution of mu (NOT conditional on sigma2)
plot(density(mu.use_nondemented), xlim=c(0, 10000), xlab=expression(sigma^2),
     ylab="density", main=expression(pi(mu~"|"~data)))
```



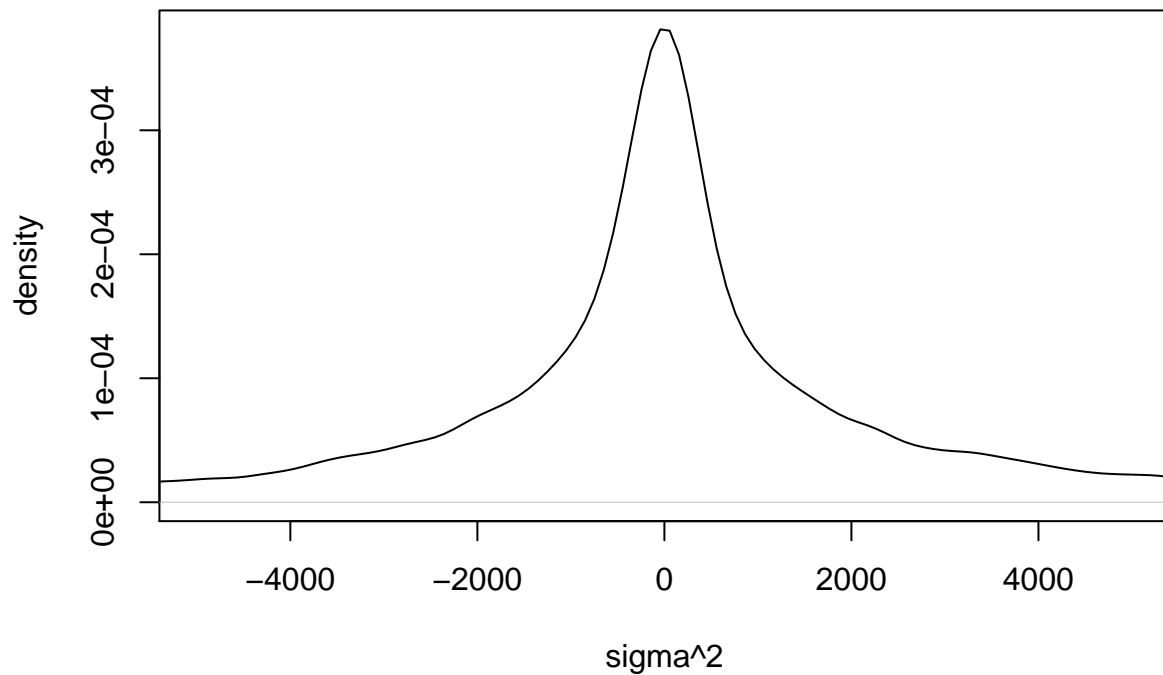
```
# marginal posterior distribution of sigma2 (NOT conditional on sigma2)
plot(density(sigma2.use_nondemented), xlim=c(0, 4000000000), xlab=expression(sigma^2),
     ylim=c(0, 0.00000001), ylab="density", main=expression(pi(sigma^2~"|"~data)))
```



We want to discover the difference between means

```
d <- mu.use_demented - mu.use_nondemented
# plot(density(d), xlim=c(-500000, 500000), xlab=expression(sigma^2), ylab="density", main=expression(p
plot(density(d), xlim=c(-5000, 5000), xlab="sigma^2", ylab="density",
     main="Posterior dist of urban-rural" )
```

Posterior dist of urban-rural



```
mean(d)
```

```
## [1] 123.3986
```

```
# 95% credible interval
```

```
quantile(d, c(.025, .975)) # The difference between TIV size between demented and nondemented is insign
```

```
##      2.5%      97.5%
```

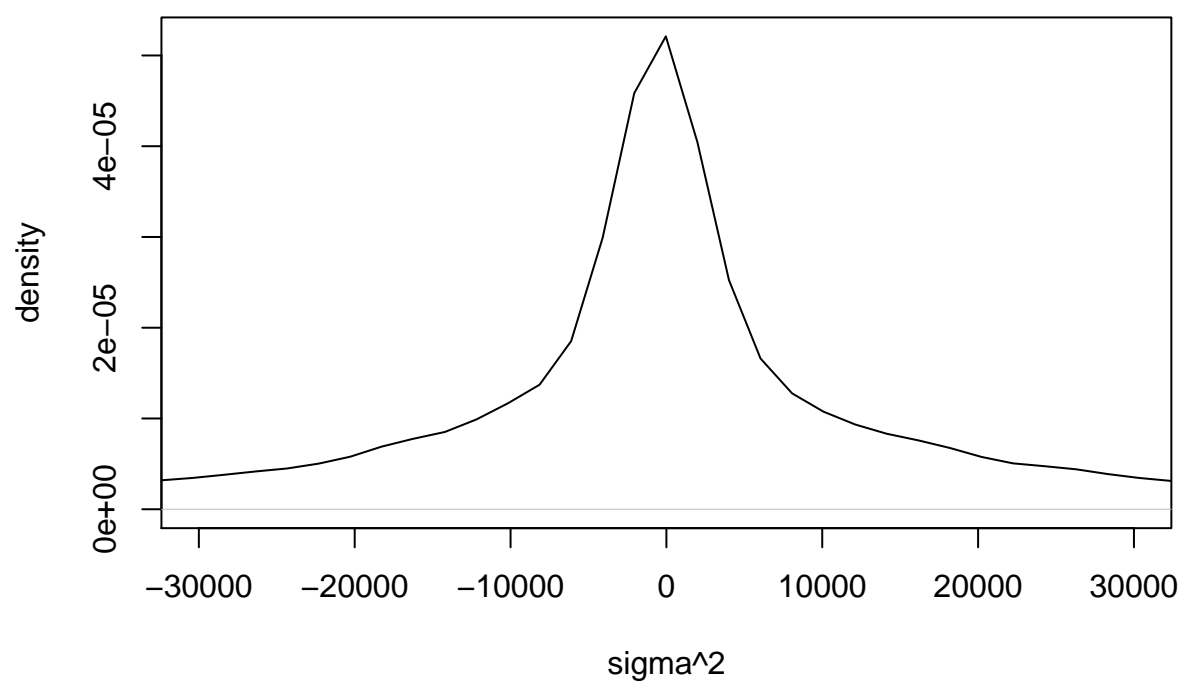
```
## -6401.807  7461.925
```

We want to discover the difference between standard deviations

```
d <- sqrt(sigma2.use_demented) - sqrt(sigma2.use_nondemented)
```

```
# plot(density(d), xlim=c(-500000, 500000), xlab=expression(sigma^2), ylab="density", main=expression(p  
plot(density(d), xlim=c(-30000, 30000), xlab="sigma^2", ylab="density",  
     main="Posterior dist of urban-rural" )
```


Posterior dist of urban-rural



```
mean(d)
```

```
## [1] -687.4292
```

```
# 95% credible interval
```

```
quantile(d, c(.025, .975)) # The difference between the standard deviation of TIV size between demented
```

```
##      2.5%      97.5%
```

```
## -80954.28 74437.67
```