206hw4

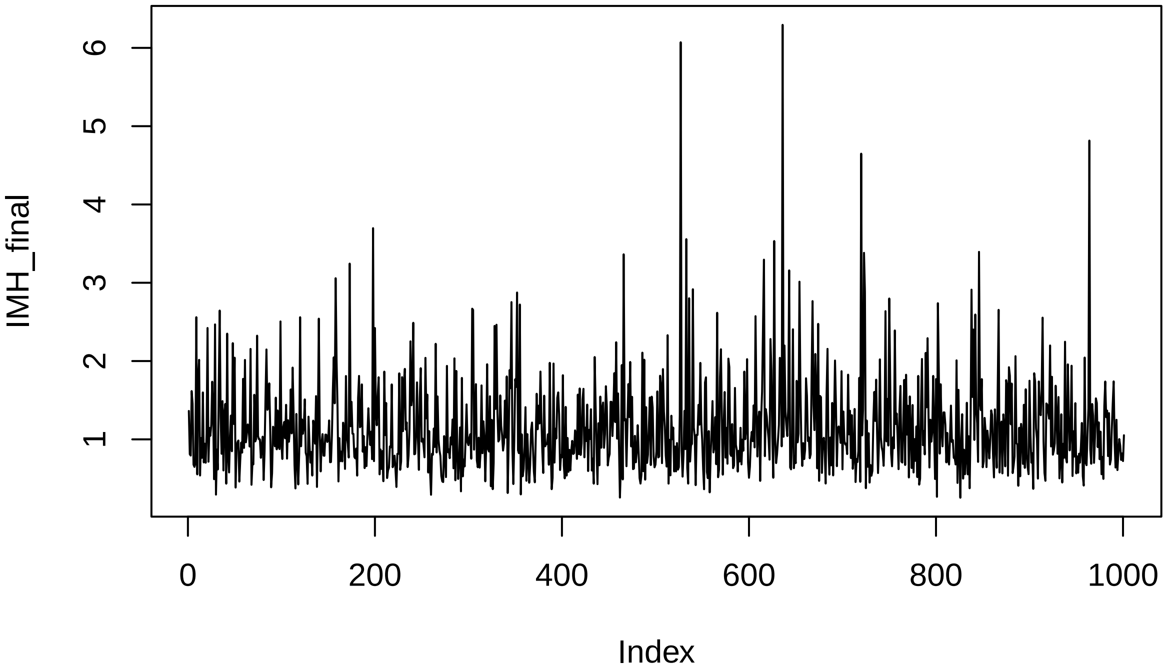
Qi Wang

2022/2/21

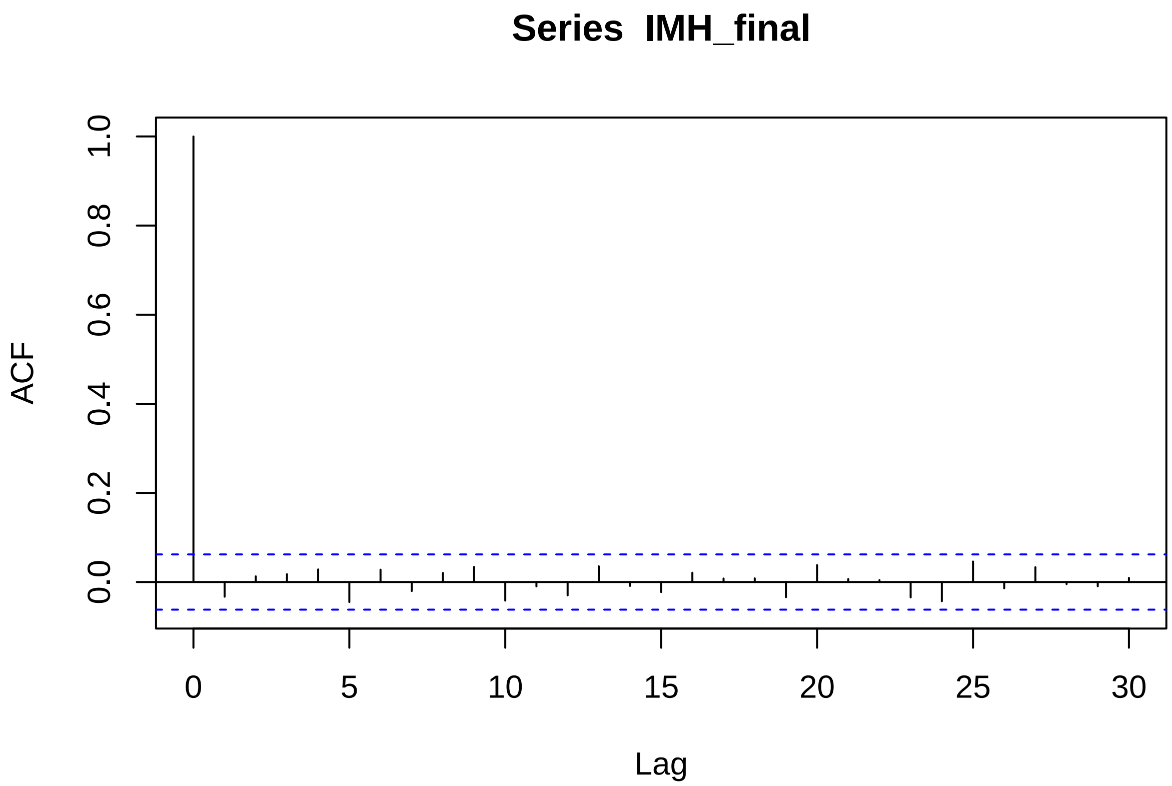
```
rm(list = ls())
```

#Question 1

```
theta_1 <- 1.5
theta_2 <- 2
f_invG <- function(x){
  fx <- x^(-3/2)*exp(-theta_1*x - theta_2/x)
  return(fx)
}
alpha <- 1
beta <- 1
pp <- function(x){
  fx <- dgamma(x, shape = alpha, scale = beta)
}

IMH <- rep(NA, 10000)
IMH[1] <- 0.1
for (i in 1:length(IMH)) {
  new <- rgamma(1, shape = alpha, scale = beta)
  p <-  f_invG(new)*pp(IMH[i]) / (f_invG(IMH[i])*pp(new))
  index <- runif(1)
  if(index <= p){
    IMH[i+1] <- new
  }else{
    IMH[i+1] <- IMH[i]
  }
}
thin <- seq(from = 2000, to = 10000, by = 8)
IMH_final <- IMH[thin]
plot(IMH_final, type = 'l')
```



```
acf(IMH_final)
```



```
mean(IMH_final)
```

```
## [1] 1.158764
```

```
mean(1/IMH_final)
```

```
## [1] 1.09085
```

```
sqrt(theta_2/theta_1)
```

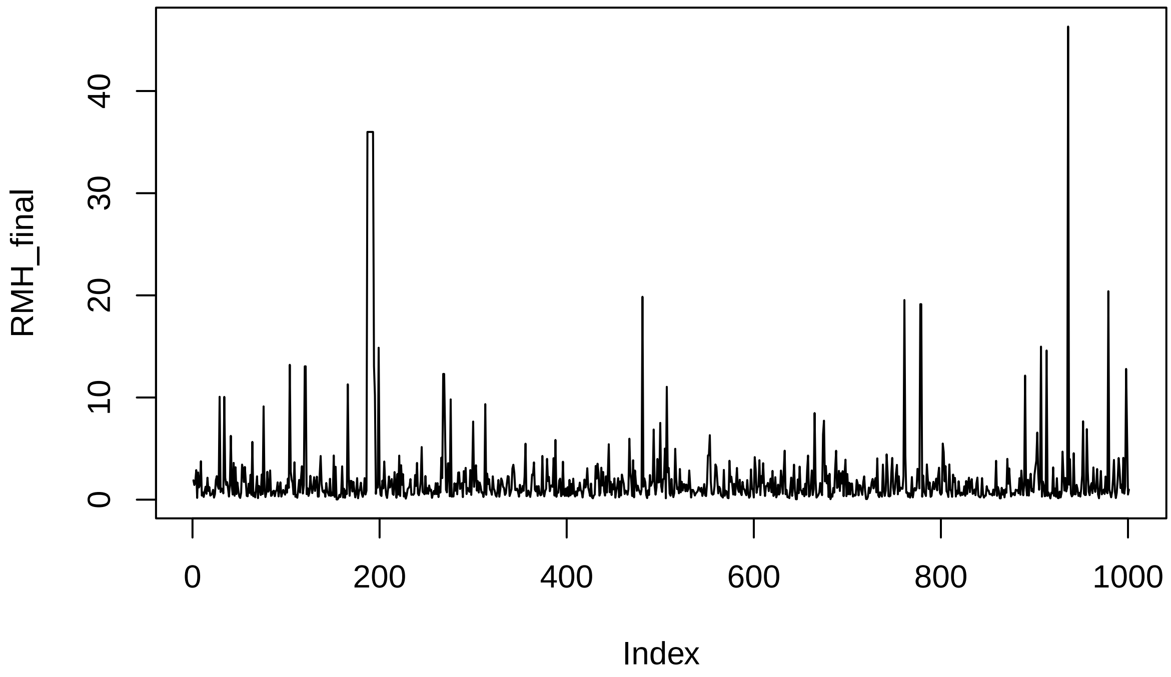
```
## [1] 1.154701
```

```
sqrt(theta_1/theta_2) + 1/(2*theta_2)
```

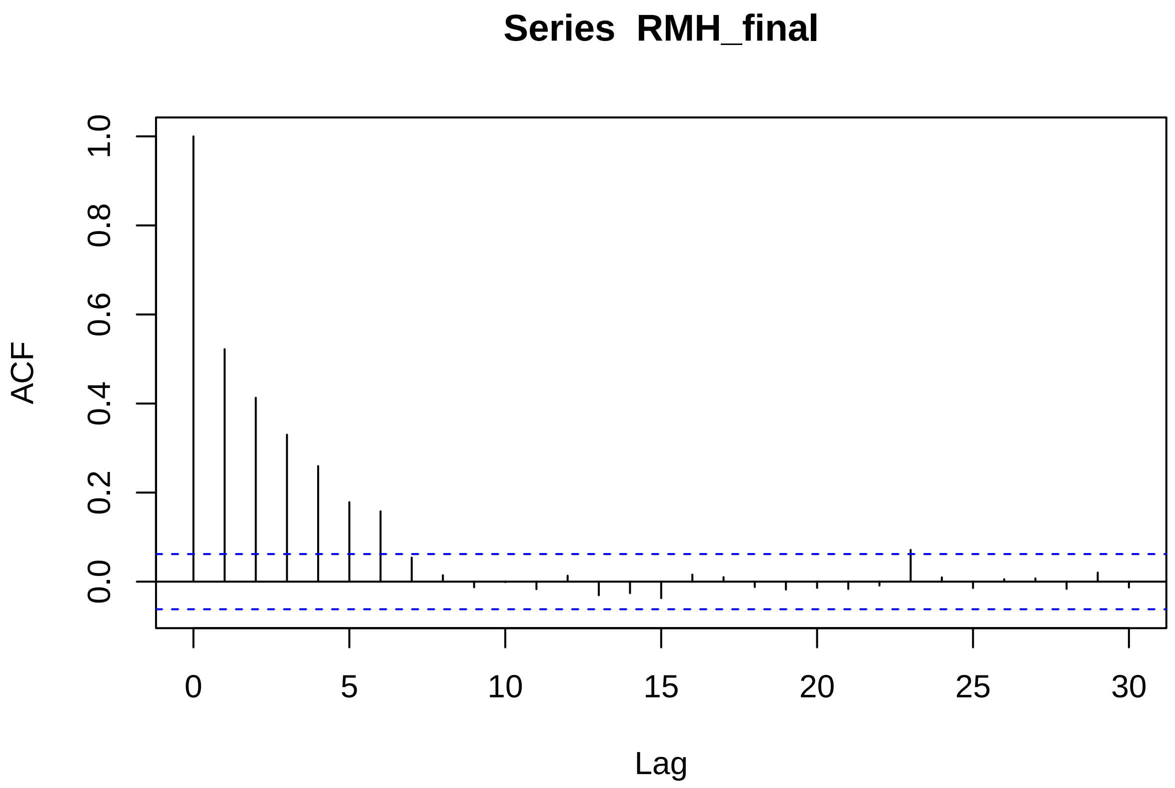
```
## [1] 1.116025
```

```
theta_1 <- 1.5
theta_2 <- 2
log_invG <- function(x){
  fx <- exp(-x/2) * exp(  -theta_1*exp(x) - theta_2/exp(x)  )
  return(fx)
}
```

```
sigma <- 2

RMH <- rep(NA, 10000)
RMH[1] <- 1
for (i in 1:length(RMH)) {
  new <- rnorm(1, mean = RMH[i], sd = sqrt(sigma))
  p <-  log_invG(new) / log_invG(IMH[i])
  index <- runif(1)
  if(index <= p){
    RMH[i+1] <- new
  }else{
    RMH[i+1] <- RMH[i]
  }
}
thin <- seq(from = 2000, to = 10000, by = 8)
RMH_final_log <- RMH[thin]

RMH_final <- exp(RMH_final_log)
plot(RMH_final, type = 'l')
```



```
acf(RMH_final)
```



```
mean(RMH_final)
```

```
## [1] 1.970528
```

```
mean(1/RMH_final)
```

```
## [1] 1.601439
```

```
sqrt(theta_2/theta_1)
```

```
## [1] 1.154701
```

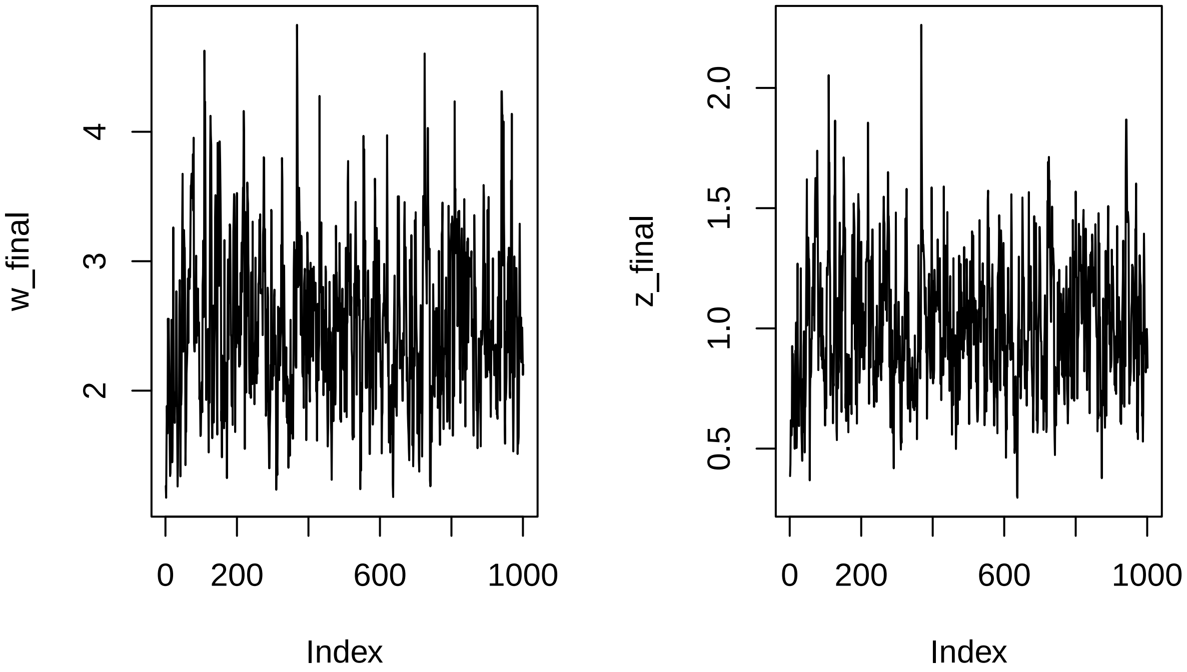
```
sqrt(theta_1/theta_2) + 1/(2*theta_2)
```

```
## [1] 1.116025
```

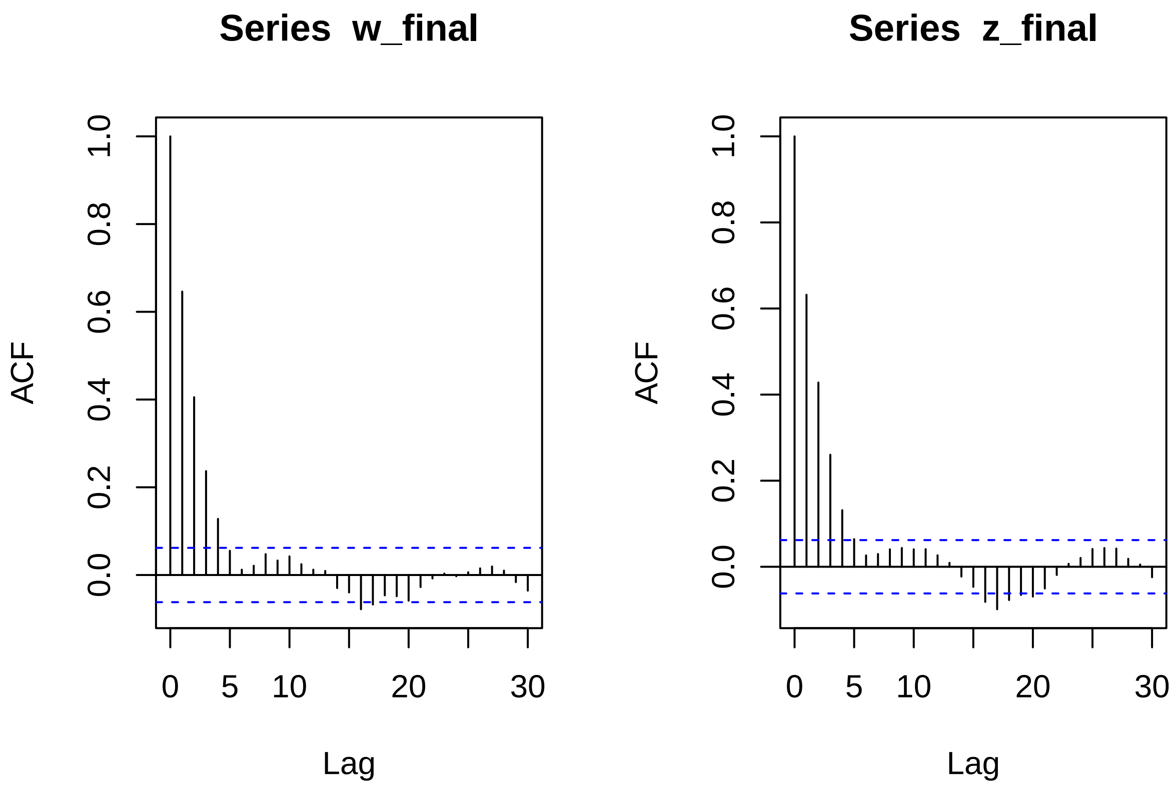
Question 2(a): MH-within-Gibbs

```
X <- read.table("D:/77/UCSC/study/206b/hw/my-data.txt")
X <- as.vector(X)
n <- nrow(X)
con_w <- function(w, beta){

  fx <- exp(3*w)/ gamma(exp(w))^n * beta^(n*exp(w)) * prod(X)^(exp(w)-1) * exp(-exp(w))
  return(fx)
}

con_z <- function(alpha, z){

  fx <- exp(z)^(n*alpha + 1) * exp(-exp(z)*sum(X) - 2*exp(z))
  return(fx)
}

sigma <- 0.2


w <- rep(NA, 10000)
z <- rep(NA, 10000)

w[1] <- 1
z[1] <- 1
a = 1
for (i in 1:length(w)) {
  z_new <- rnorm(1, z[i], sd = sqrt(sigma))
  p <- con_z(alpha = exp(w[i]), z = z_new)/con_z(alpha = exp(w[i]), z = z[i])
  k <- runif(1)
  if(k <= p ){
    z[i+1] <- z_new
  }else{
    z[i+1] <- z[i]
  }

  w_new <- rnorm(1, w[i], sd = sqrt(sigma))
  p <- con_w(w = w_new, beta = exp(z[i+1]))/con_w(w = w[i], beta = exp(z[i+1]))
  k <- runif(1)
  if(k <= p){
    w[i+1] <- w_new
  }else{
    w[i+1] <- w[i]
  }
  a = a+1

}
```

```
w_final <- exp(w[thin])
z_final <- exp(z[thin])

par(mfrow = c(1,2))
plot(w_final, type = 'l' )
plot(z_final, type = 'l' )
```



```
acf(w_final)
acf(z_final)
```



```
CI_a <- c(mean(w_final), mean(z_final), quantile(w_final, 0.025), quantile(z_final, 0.025),quantile(w_fi
CI_a <- matrix(CI_a, 4, 2, byrow = T)
colnames(CI_a) <- c("Alpha", "Beta")
rownames(CI_a) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_a)
```

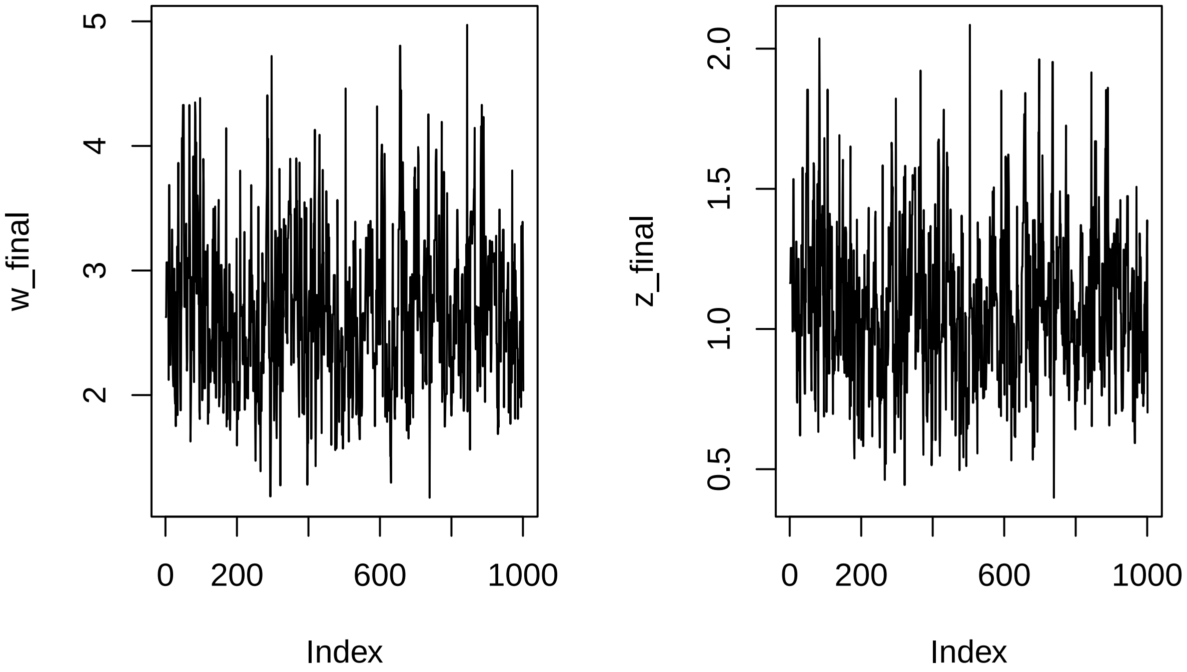
	Alpha	Beta
Mean	2.501284	1.0038527
2.5% Quantile	1.460755	0.5578775
97.5% Quantile	3.914277	1.5800860
Effective Sample Size	200.050576	188.9312738

Question 2(b): R-MH

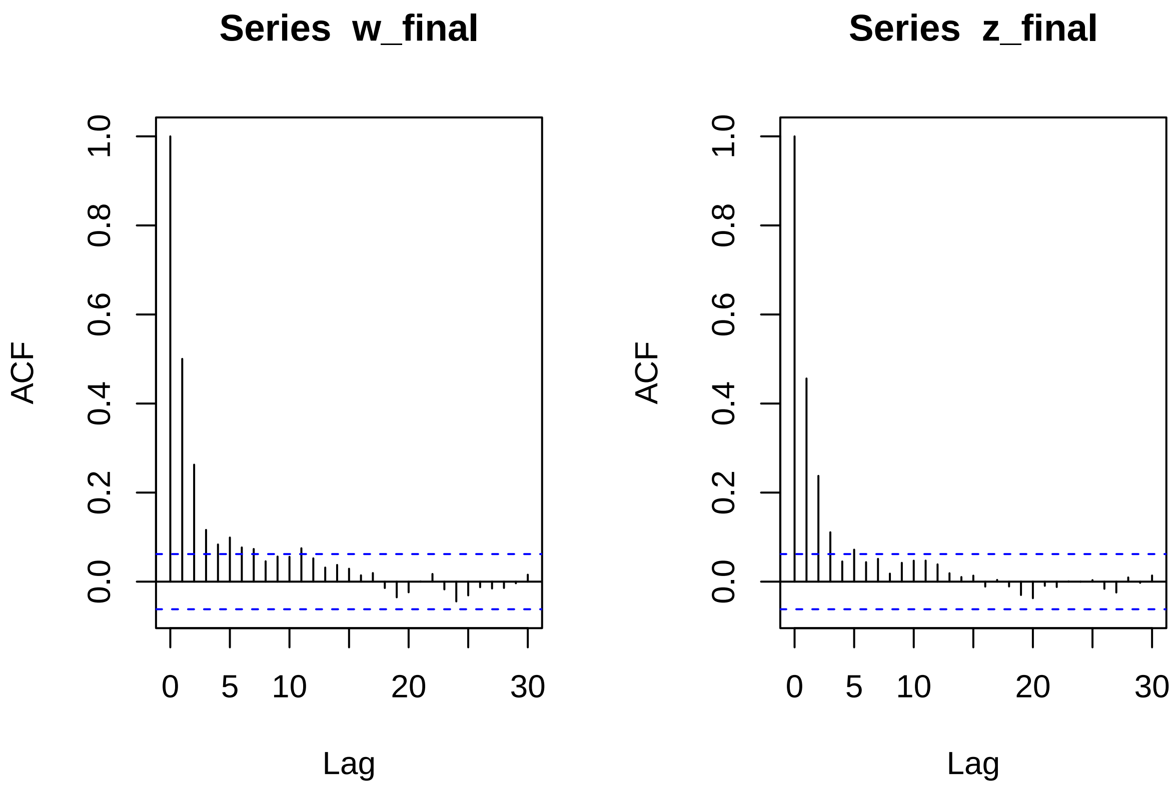
```
pos_joint_log <- function(w, z){
  f <- exp(w)^2 / gamma(exp(w))^n * exp(z*( n*exp(w) + 1 )) * prod(X)^(exp(w)-1)*
    exp(-exp(z)*sum(X)-exp(w)-2*exp(z))*exp(w)*exp(z)
  return(f)
}

sigma <- 0.2
```

```
Sigma <- diag(sigma, 2)

param = matrix(NA, nrow = 2, ncol = 10000)

param[1,1] <- 1
param[2,1] <- 1


for (i in 1: (ncol(param)-1) ) {

  new <- mvtnorm::rmvnorm(1, mean = param[,i], sigma = Sigma)
  p <- pos_joint_log(w = new[1], z = new[2]) / pos_joint_log(w = param[1,i], z = param[2,i])
  k <- runif(1)
  if (k <= p){
    param[,i+1] <- new
  }else{
    param[,i+1] <- param[, i]
  }

}
```

```
param_final <- param[,thin]


w_final <- exp(param_final[1,])
z_final <- exp(param_final[2,])

par(mfrow = c(1,2))
plot(w_final, type = 'l' )
plot(z_final, type = 'l' )
```



```
acf(w_final)
acf(z_final)
```



```
CI_b <- c(mean(w_final), mean(z_final), quantile(w_final, 0.025), quantile(z_final, 0.025),quantile(w_f
CI_b <- matrix(CI_b, 4, 2, byrow = T)
colnames(CI_b) <- c("Alpha", "Beta")
rownames(CI_b) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_b)
```

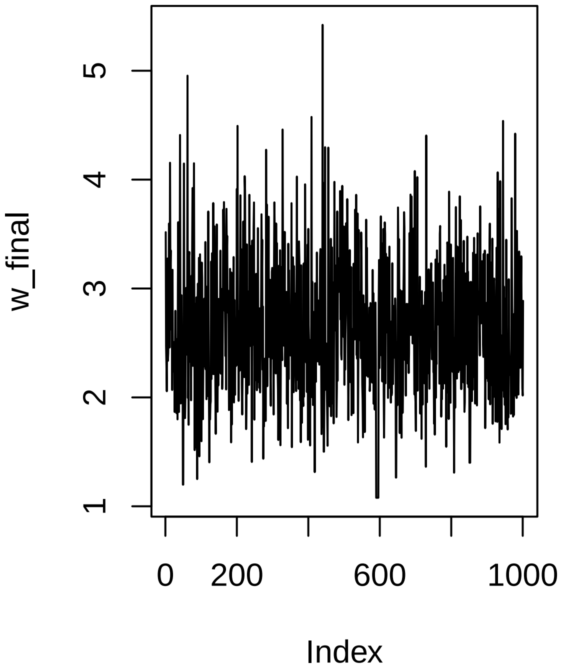
	Alpha	Beta
Mean	2.669832	1.0824115
2.5% Quantile	1.646393	0.6032106
97.5% Quantile	4.063334	1.6761819
Effective Sample Size	274.496995	254.1009009

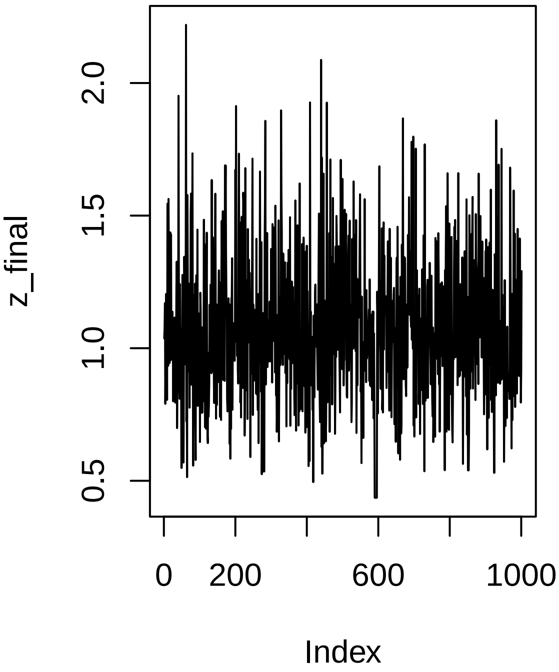
Question 2(c): I-MH_Laplace

```
log_pos <- function(a){
  f <- 2*a[1] - n*log(gamma(exp(a[1])))+a[2]*(n*exp(a[1])+1)+(exp(a[1])-1)*sum(log(X))-exp(a[2])*sum(X)
  g <- -f
  return(g)
}

opt <- optim(par = c(0.8,0), fn = log_pos, hessian = TRUE)
```

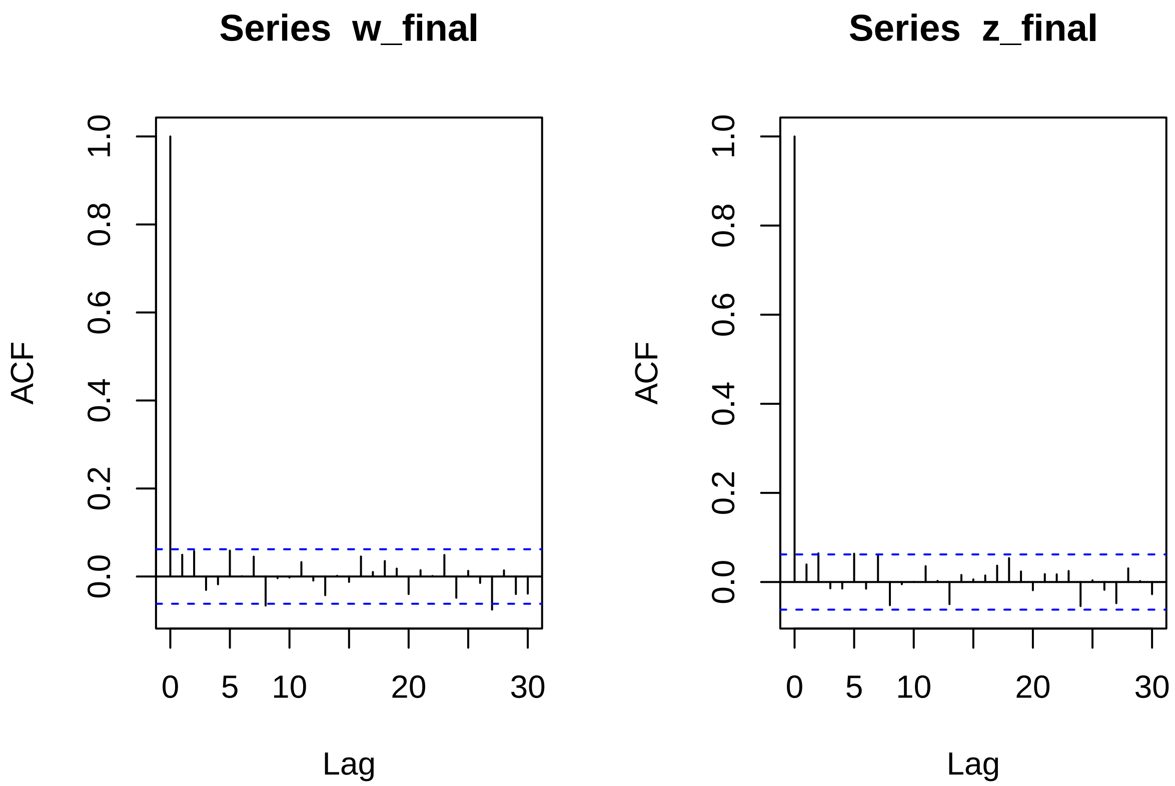
```
M <- opt$par
S <- solve(opt$hessian)

pos_joint_log <- function(w, z){
  f <- exp(w)^2 / gamma(exp(w))^n * exp(z*( n*exp(w) + 1 )) * prod(X)^(exp(w)-1)*
    exp(-exp(z)*sum(X)-exp(w)-2*exp(z))*exp(w)*exp(z)
  return(f)
}


param = matrix(NA, nrow = 2, ncol = 10000)

param[1,1] <- 1
param[2,1] <- 1


for (i in 1: (ncol(param)-1) ) {

  new <- mvtnorm::rmvnorm(1, mean = M, sigma = S)
  p <- pos_joint_log(w = new[1], z = new[2]) * mvtnorm::dmvnorm(param[,i], mean = M, sigma = S) /
    ( pos_joint_log(w = param[1,i], z = param[2,i]) * mvtnorm::dmvnorm(new, mean = M, sigma = S) )

  k <- runif(1)
  if (k <= p){
    param[,i+1] <- new
  }else{
    param[,i+1] <- param[, i]
  }

}
```

```
param_final <- param[,thin]


w_final <- exp(param_final[1,])
z_final <- exp(param_final[2,])

par(mfrow = c(1,2))
plot(w_final, type = 'l' )
plot(z_final, type = 'l' )
```





```
acf(w_final)
acf(z_final)
```



```
CI_c <- c(mean(w_final), mean(z_final), quantile(w_final, 0.025), quantile(z_final, 0.025),quantile(w_f
CI_c <- matrix(CI_c, 4, 2, byrow = T)
colnames(CI_c) <- c("Alpha", "Beta")
rownames(CI_c) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_c)
```

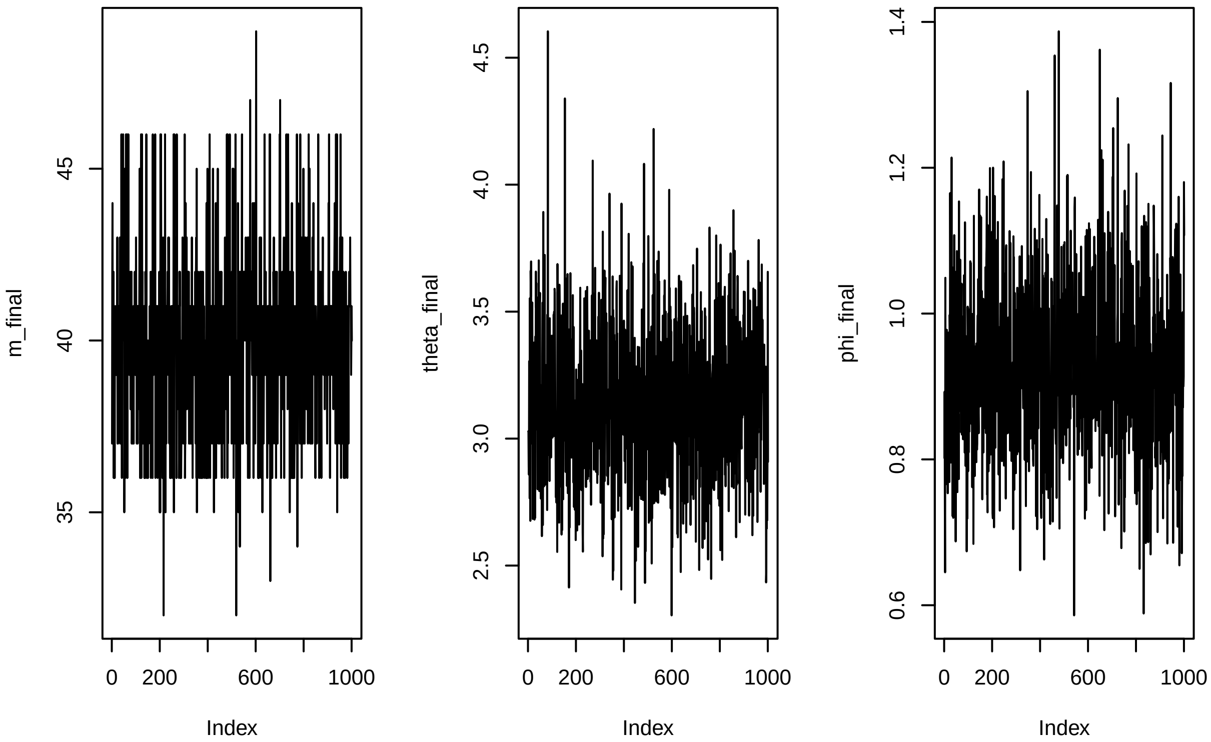
	Alpha	Beta
Mean	2.632236	1.0701328
2.5% Quantile	1.541476	0.5571119
97.5% Quantile	3.922255	1.6812000
Effective Sample Size	1001.000000	1001.0000000

Question 4: (alpha = gamma = 1, beta = delta = 0.1)

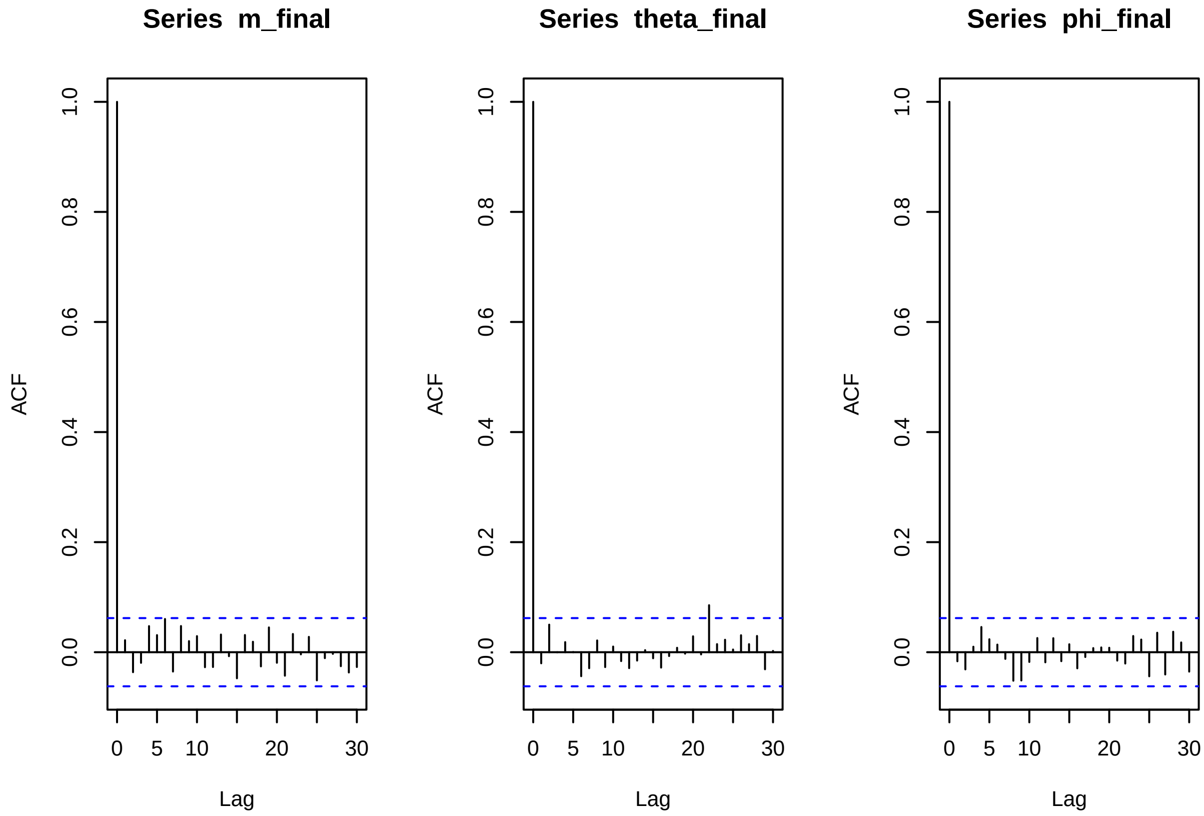
```
y <-
c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,
2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,
0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,
0,1)
n <- length(y)
```

```r
con_m <- function(m, theta, phi){
  pm_raw <- rep(NA, n)
  for (j in 1:n) {
    pm_raw[j] <- theta^(sum(y[1:j])) * exp(-j*theta)* phi^(sum(y[j+1:n])) * exp(j*phi)
  }
  pm_norm <- pm_raw/sum(pm_raw)
  return(pm_norm)
}
m_pos <- rep(NA, 10000)
theta_pos <- rep(NA, 10000)
phi_pos <- rep(NA, 10000)

alpha <- 1
gam <- 1
beta <- 0.1
delta <- 0.1

m_pos[1] <- 10
theta_pos[1] <- 1
phi_pos[1] <- 1

for (i in 1:9999) {
  theta_pos[i+1] <- rgamma(1, shape = sum(y[1:m_pos[i]]) + alpha, rate = m_pos[i] + beta )
  phi_pos[i+1] <- rgamma(1, shape = sum( y[ (m_pos[i]+1) : n] ) + gam, rate = n - m_pos[i]+ delta)
  pm_raw <- rep(NA, n)
  for (j in 1:(n-1)) {
    pm_raw[j] <- theta_pos[i+1]^(sum(y[1:j])) * exp(-j*theta_pos[i+1])* phi_pos[i+1]^(sum(y[(j+1):n])) *
  }
  pm_raw[n] <- theta_pos[i+1]^(sum(y)) * exp(-n*theta_pos[i+1]) * exp(n*phi_pos[i+1])
  pm_norm <- pm_raw/sum(pm_raw)
  m_pos[i+1] <- which.max(rmultinom(n = 1, size = 1, prob = pm_norm))

}
```

```r
m_final <- m_pos[thin]
theta_final <- theta_pos[thin]
phi_final <- phi_pos[thin]
par(mfrow = c(1,3))

plot(m_final, type = 'l')
plot(theta_final, type = 'l')
plot(phi_final, type = 'l')
```



```
acf(m_final)
acf(theta_final)
acf(phi_final)
```



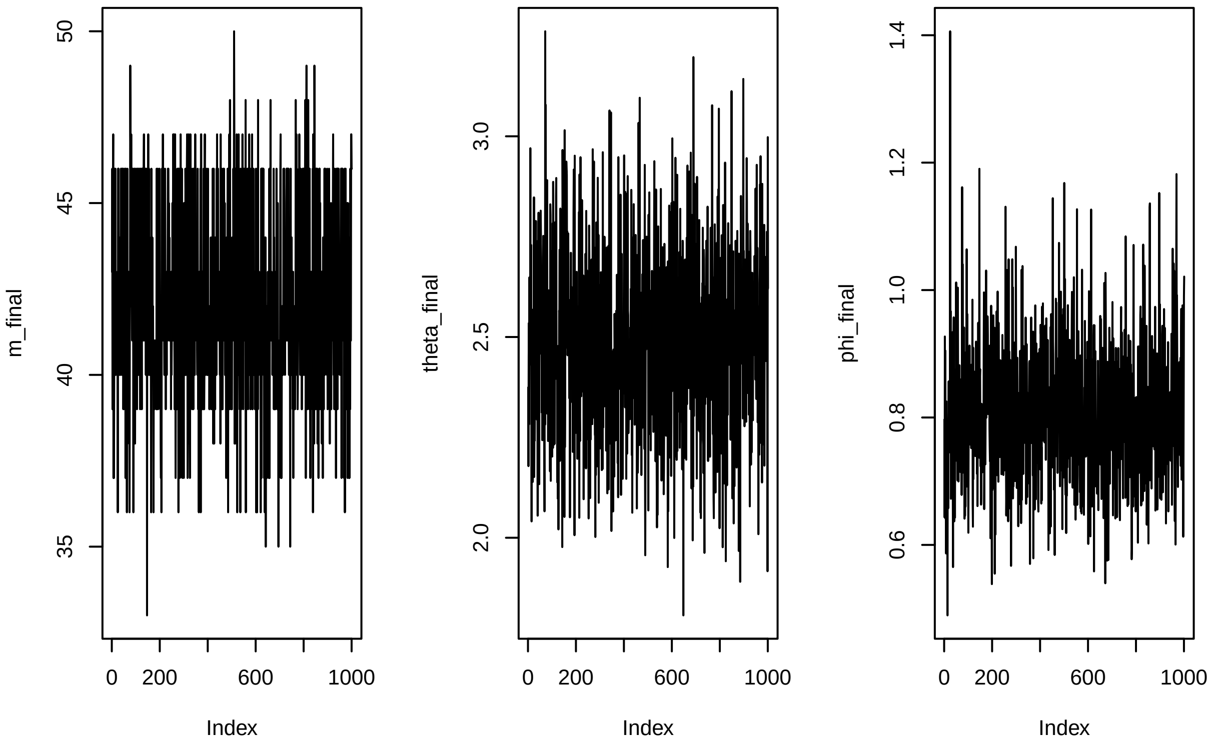
```r
CI_4b <- matrix(c(
  mean(m_final), mean(theta_final), mean(phi_final),
  quantile(m_final, 0.025),quantile(theta_final, 0.025),quantile(phi_final, 0.025),
  quantile(m_final, 0.975),quantile(theta_final, 0.975),quantile(phi_final, 0.975),
  mcmcse::ess(m_final), mcmcse::ess(theta_final),mcmcse::ess(phi_final)
), nrow = 4, ncol = 3, byrow = TRUE)
colnames(CI_4b) <- c("M", "Theta", "Phi")
rownames(CI_4b) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_4b)
```

	M	Theta	Phi
Mean	40.00799	3.141653	0.929278
2.5% Quantile	36.00000	2.600286	0.711369
97.5% Quantile	46.00000	3.699646	1.168680
Effective Sample Size	1001.00000	1001.000000	1001.000000

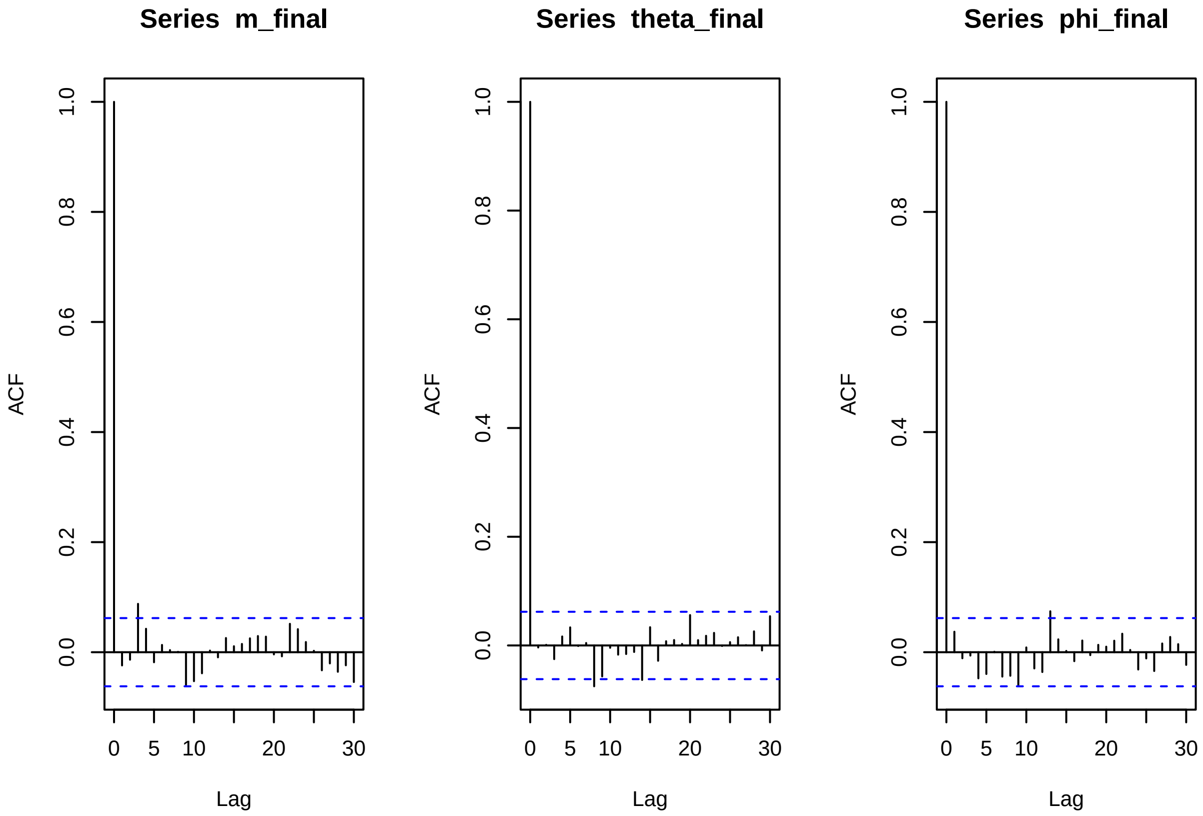
Question 4: (alpha = gamma = 1, beta = delta = 10)

```r
y <-
c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,
2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,
0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,
0,1)
n <- length(y)

con_m <- function(m, theta, phi){
  pm_raw <- rep(NA, n)
  for (j in 1:n) {
    pm_raw[j] <- theta^(sum(y[1:j])) * exp(-j*theta)* phi^(sum(y[j+1:n])) * exp(j*phi)
  }
  pm_norm <- pm_raw/sum(pm_raw)
  return(pm_norm)
}
m_pos <- rep(NA, 10000)
theta_pos <- rep(NA, 10000)
phi_pos <- rep(NA, 10000)

alpha <- 1
gam <- 1
beta <- 10
delta <- 10

m_pos[1] <- 10
theta_pos[1] <- 1
phi_pos[1] <- 1

for (i in 1:9999) {
  theta_pos[i+1] <- rgamma(1, shape = sum(y[1:m_pos[i]]) + alpha, rate = m_pos[i] + beta )
  phi_pos[i+1] <- rgamma(1, shape = sum( y[ (m_pos[i]+1) : n] ) + gam, rate = n - m_pos[i]+ delta)
  pm_raw <- rep(NA, n)
  for (j in 1:(n-1)) {
    pm_raw[j] <- theta_pos[i+1]^(sum(y[1:j])) * exp(-j*theta_pos[i+1])* phi_pos[i+1]^(sum(y[(j+1):n])) 
  }
  pm_raw[n] <- theta_pos[i+1]^(sum(y)) * exp(-n*theta_pos[i+1]) * exp(n*phi_pos[i+1])
  pm_norm <- pm_raw/sum(pm_raw)
  m_pos[i+1] <- which.max(rmultinom(n = 1, size = 1, prob = pm_norm))

}
```

```
m_final <- m_pos[thin]
theta_final <- theta_pos[thin]
phi_final <- phi_pos[thin]
par(mfrow = c(1,3))

plot(m_final, type = 'l')
plot(theta_final, type = 'l')
plot(phi_final, type = 'l')
```



```
acf(m_final)
acf(theta_final)
acf(phi_final)
```



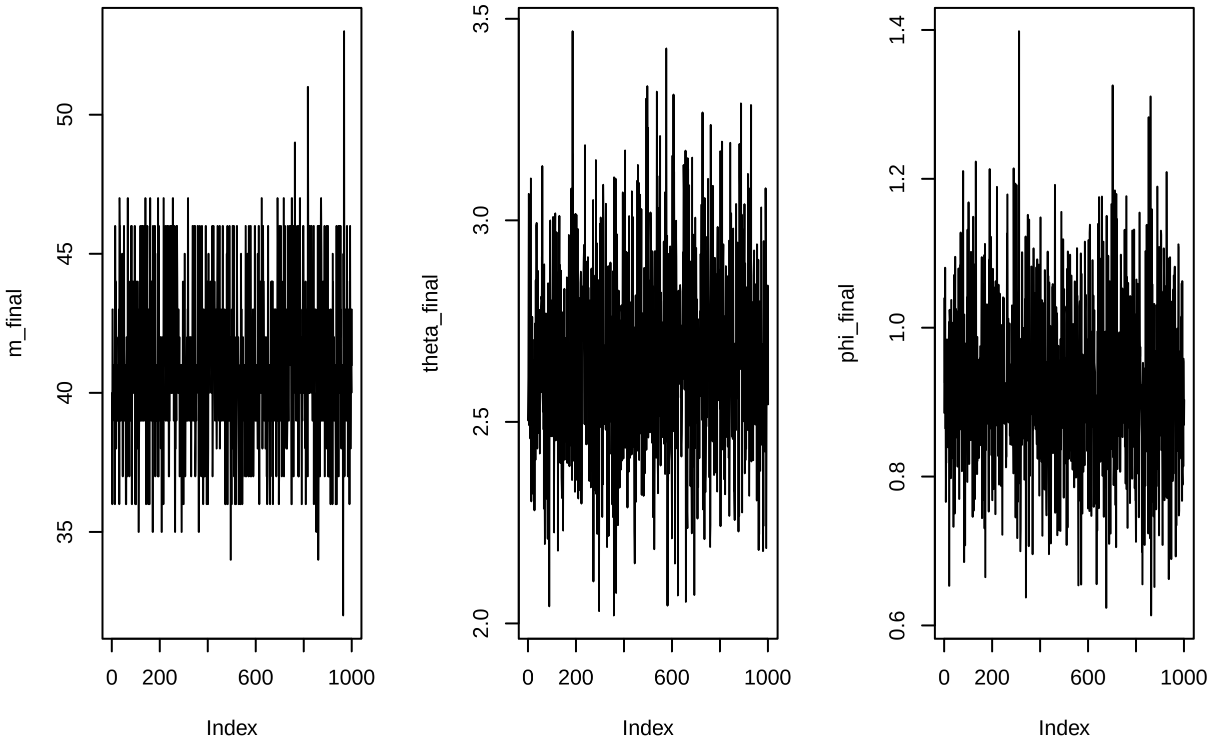
```
CI_4b <- matrix(c(
  mean(m_final), mean(theta_final), mean(phi_final),
  quantile(m_final, 0.025),quantile(theta_final, 0.025),quantile(phi_final, 0.025),
  quantile(m_final, 0.975),quantile(theta_final, 0.975),quantile(phi_final, 0.975),
  mcmcse::ess(m_final), mcmcse::ess(theta_final),mcmcse::ess(phi_final)
), nrow = 4, ncol = 3, byrow = TRUE)
colnames(CI_4b) <- c("M", "Theta", "Phi")
rownames(CI_4b) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_4b)
```

	M	Theta	Phi
Mean	42.08192	2.486429	0.8094665
2.5% Quantile	37.00000	2.051021	0.6186348
97.5% Quantile	47.00000	2.946450	1.0376872
Effective Sample Size	1001.00000	1001.000000	1001.0000000

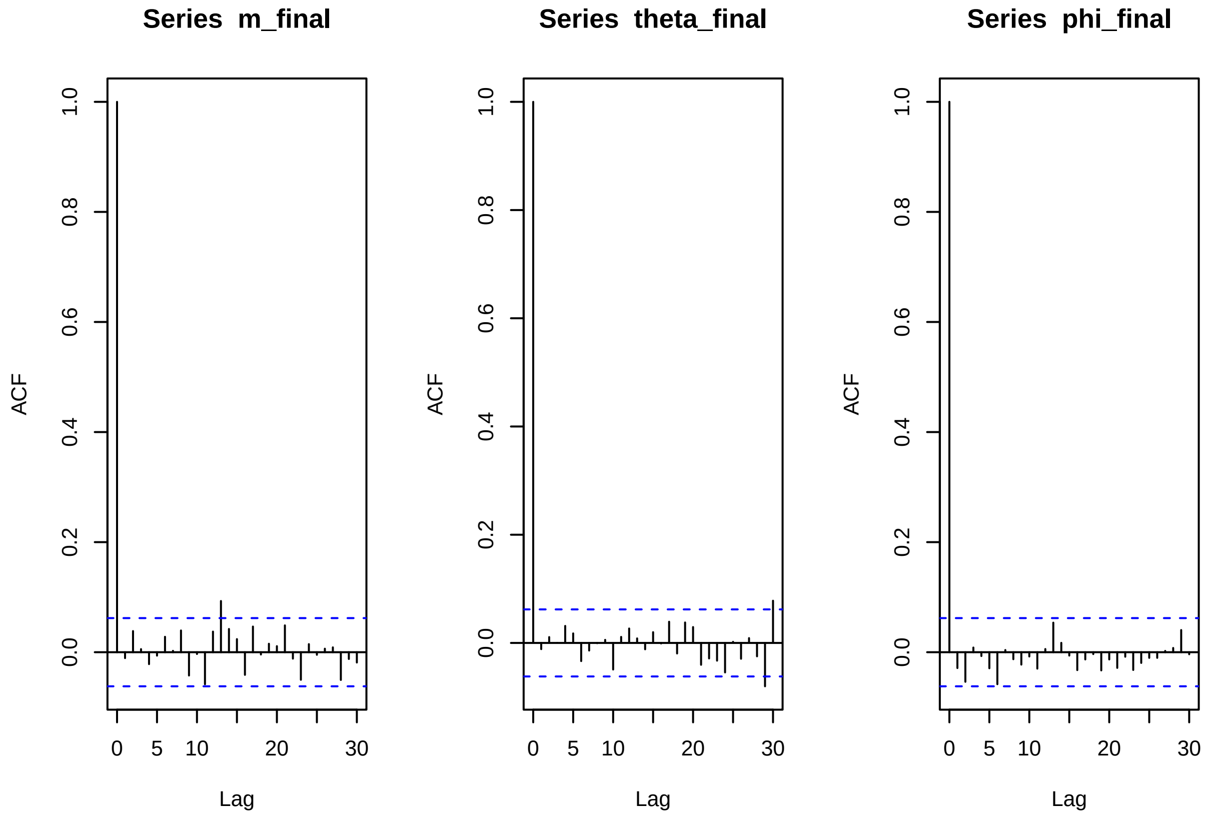
Question 4: (alpha = gamma = 10, beta = delta = 10)

```
y <-
c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,
2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,
0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,
0,1)
n <- length(y)

con_m <- function(m, theta, phi){
  pm_raw <- rep(NA, n)
  for (j in 1:n) {
    pm_raw[j] <- theta^(sum(y[1:j])) * exp(-j*theta)* phi^(sum(y[j+1:n])) * exp(j*phi)
  }
  pm_norm <- pm_raw/sum(pm_raw)
  return(pm_norm)
}
m_pos <- rep(NA, 10000)
theta_pos <- rep(NA, 10000)
phi_pos <- rep(NA, 10000)

alpha <- 10
gam <- 10
beta <- 10
delta <- 10

m_pos[1] <- 10
theta_pos[1] <- 1
phi_pos[1] <- 1

for (i in 1:9999) {
  theta_pos[i+1] <- rgamma(1, shape = sum(y[1:m_pos[i]]) + alpha, rate = m_pos[i] + beta )
  phi_pos[i+1] <- rgamma(1, shape = sum( y[ (m_pos[i]+1) : n] ) + gam, rate = n - m_pos[i]+ delta)
  pm_raw <- rep(NA, n)
  for (j in 1:(n-1)) {
    pm_raw[j] <- theta_pos[i+1]^(sum(y[1:j])) * exp(-j*theta_pos[i+1])* phi_pos[i+1]^(sum(y[(j+1):n])) *
  }
  pm_raw[n] <- theta_pos[i+1]^(sum(y)) * exp(-n*theta_pos[i+1]) * exp(n*phi_pos[i+1])
  pm_norm <- pm_raw/sum(pm_raw)
  m_pos[i+1] <- which.max(rmultinom(n = 1, size = 1, prob = pm_norm))

}
```

```
m_final <- m_pos[thin]
theta_final <- theta_pos[thin]
phi_final <- phi_pos[thin]
par(mfrow = c(1,3))

plot(m_final, type = 'l')
plot(theta_final, type = 'l')
plot(phi_final, type = 'l')
```



```
acf(m_final)
acf(theta_final)
acf(phi_final)
```



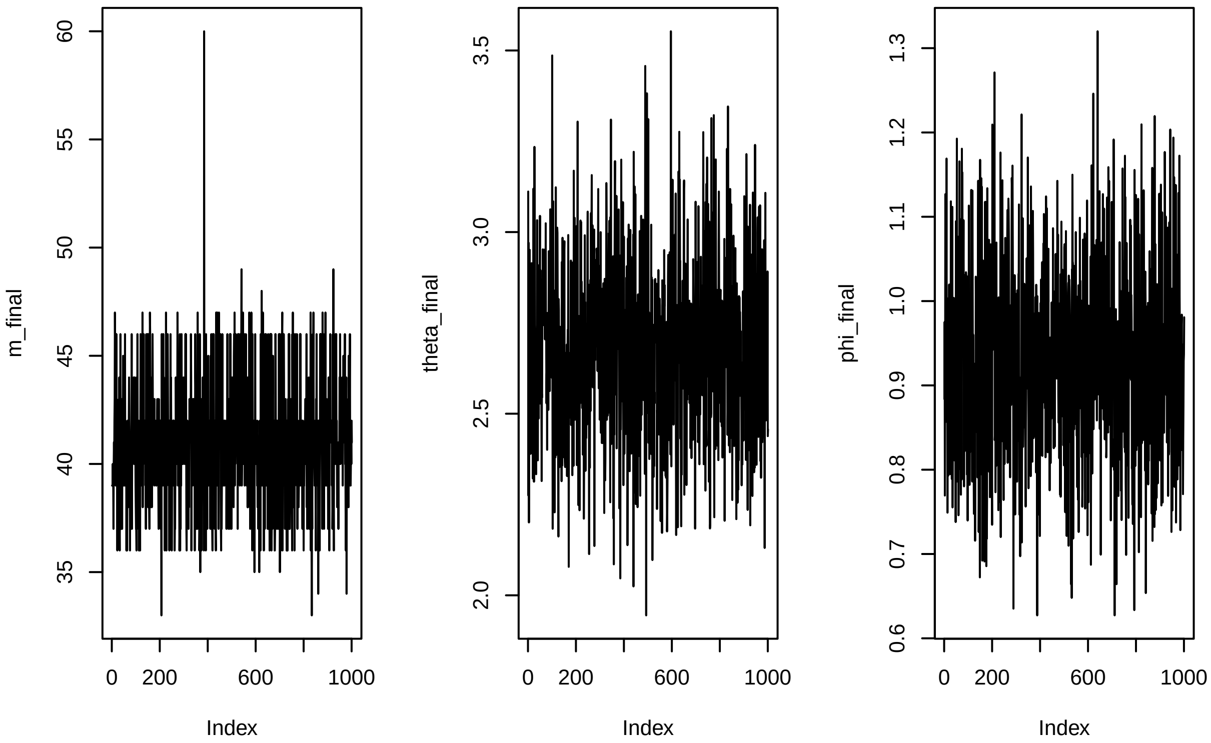
```
CI_4b <- matrix(c(
  mean(m_final), mean(theta_final), mean(phi_final),
  quantile(m_final, 0.025),quantile(theta_final, 0.025),quantile(phi_final, 0.025),
  quantile(m_final, 0.975),quantile(theta_final, 0.975),quantile(phi_final, 0.975),
  mcmcse::ess(m_final), mcmcse::ess(theta_final),mcmcse::ess(phi_final)
), nrow = 4, ncol = 3, byrow = TRUE)
colnames(CI_4b) <- c("M", "Theta", "Phi")
rownames(CI_4b) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_4b)
```

	M	Theta	Phi
Mean	40.97203	2.664739	0.9230648
2.5% Quantile	36.00000	2.222403	0.7102387
97.5% Quantile	46.00000	3.137233	1.1545735
Effective Sample Size	1001.00000	1001.000000	1001.0000000

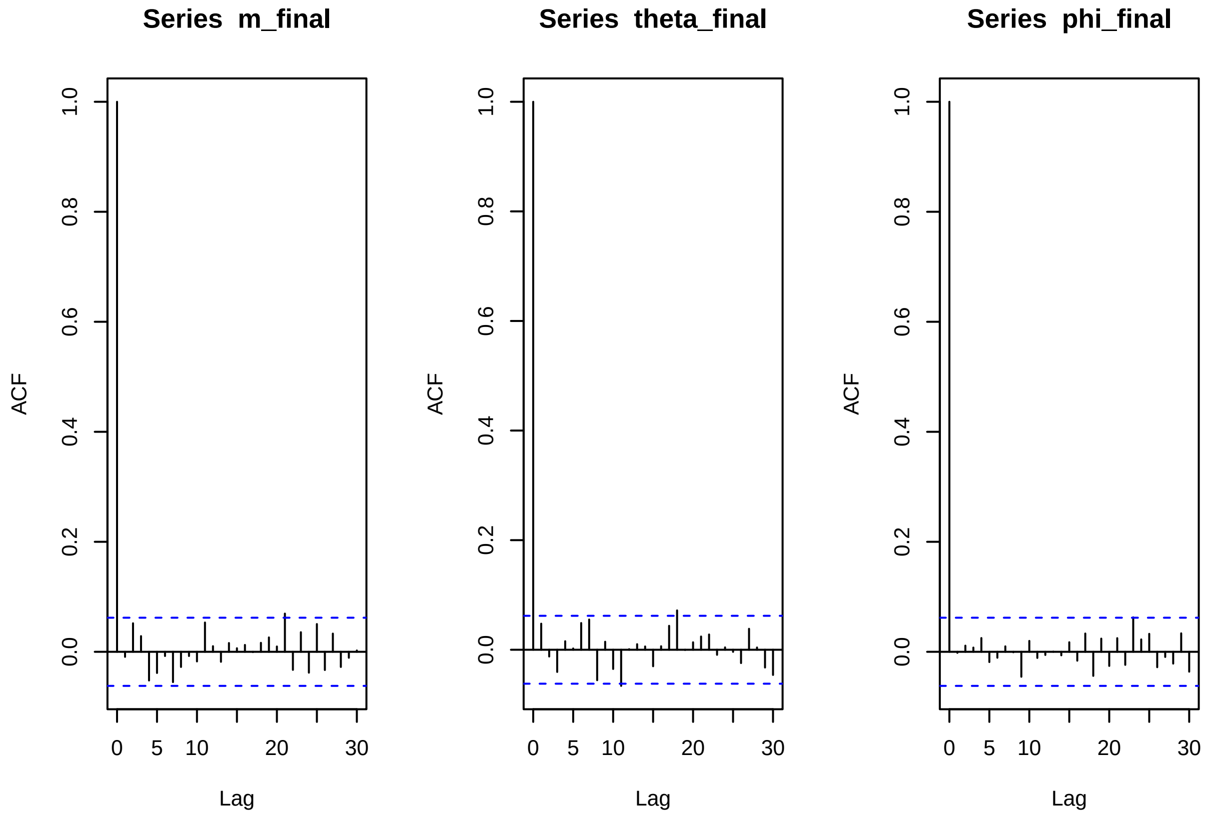
Question 4: (alpha = gamma = 100, beta = delta = 10)

```
y <-
c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,
2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,
0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,
0,1)
n <- length(y)

con_m <- function(m, theta, phi){
  pm_raw <- rep(NA, n)
  for (j in 1:n) {
    pm_raw[j] <- theta^(sum(y[1:j])) * exp(-j*theta)* phi^(sum(y[j+1:n])) * exp(j*phi)
  }
  pm_norm <- pm_raw/sum(pm_raw)
  return(pm_norm)
}
m_pos <- rep(NA, 10000)
theta_pos <- rep(NA, 10000)
phi_pos <- rep(NA, 10000)

alpha <- 10
gam <- 10
beta <- 10
delta <- 10

m_pos[1] <- 10
theta_pos[1] <- 1
phi_pos[1] <- 1

for (i in 1:9999) {
  theta_pos[i+1] <- rgamma(1, shape = sum(y[1:m_pos[i]]) + alpha, rate = m_pos[i] + beta )
  phi_pos[i+1] <- rgamma(1, shape = sum( y[ (m_pos[i]+1) : n] ) + gam, rate = n - m_pos[i]+ delta)
  pm_raw <- rep(NA, n)
  for (j in 1:(n-1)) {
    pm_raw[j] <- theta_pos[i+1]^(sum(y[1:j])) * exp(-j*theta_pos[i+1])* phi_pos[i+1]^(sum(y[(j+1):n])) *
  }
  pm_raw[n] <- theta_pos[i+1]^(sum(y)) * exp(-n*theta_pos[i+1]) * exp(n*phi_pos[i+1])
  pm_norm <- pm_raw/sum(pm_raw)
  m_pos[i+1] <- which.max(rmultinom(n = 1, size = 1, prob = pm_norm))

}
```

```
m_final <- m_pos[thin]
theta_final <- theta_pos[thin]
phi_final <- phi_pos[thin]
par(mfrow = c(1,3))

plot(m_final, type = 'l')
plot(theta_final, type = 'l')
plot(phi_final, type = 'l')
```



```
acf(m_final)
acf(theta_final)
acf(phi_final)
```



```
CI_4b <- matrix(c(
  mean(m_final), mean(theta_final), mean(phi_final),
  quantile(m_final, 0.025),quantile(theta_final, 0.025),quantile(phi_final, 0.025),
  quantile(m_final, 0.975),quantile(theta_final, 0.975),quantile(phi_final, 0.975),
  mcmcse::ess(m_final), mcmcse::ess(theta_final),mcmcse::ess(phi_final)
), nrow = 4, ncol = 3, byrow = TRUE)
colnames(CI_4b) <- c("M", "Theta", "Phi")
rownames(CI_4b) <- c("Mean", "2.5% Quantile", "97.5% Quantile", "Effective Sample Size")
knitr::kable(CI_4b)
```

	M	Theta	Phi
Mean	40.98701	2.675120	0.9291854
2.5% Quantile	36.00000	2.210989	0.7214093
97.5% Quantile	46.00000	3.144163	1.1555482
Effective Sample Size	1001.00000	1001.000000	1001.0000000

Question 5

```
dat_q5 <- read.csv("D:/77/UCSC/study/206b/hw/q5.csv", header = T)
dat <- reshape(dat_q5, v.names=c("weight","time"), timevar="visit",idvar="patient", direction="wide")
dat <- as.matrix(dat)
```

```
alpha = beta = tau_alpha = tau_beta = sigma <- rep(NA, 10000)
alpha_all = beta_all <- matrix(NA, nrow = max(dat_q5$patient), ncol = 10000)

#Initialize

alpha[1] <-  160

beta[1] <- 1

tau_alpha[1] <- 1

tau_beta[1] <- 1

sigma[1] <- 1

alpha_all[,1] <- 1

beta_all[,1] <- 1


#hyper:

a_sigma <- 1

b_sigma <- 0.1

a_alpha <- 1

b_alpha <- 0.1

a_beta <- 1

b_beta <- 0.1

P_alpha <- 100

P_beta <- 1

attach(dat_q5)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     X
```

```
I <- max(dat_q5$patient)
n_i <- as.vector(table(dat_q5$patient))
N <- nrow(dat_q5)
```

```
set.seed(999)
for (m in 1:9999) {

# Generate new Alpha_i Vector
```

```
  yij <- dat[,seq(from = 3, to = 15, by = 2)]
  tij <- dat[,seq(from = 4, to = 16, by = 2)]
  mean_vec_alpha <- ( rowSums ( (yij - tij * beta_all[,m]) / sigma[m], na.rm = TRUE ) + alpha[m]/tau_al
  alpha_all[,m+1] <- mvtnorm::rmvnorm(1, mean = mean_vec_alpha, sigma = diag(  1/( n_i/sigma[m] + 1/tau_

# Generate new Beta_i Vector
  mean_vec_beta <- ( rowSums( (yij - alpha_all[,m+1])*tij, na.rm = TRUE) / sigma[m] + beta[m]/tau_beta[
    ( rowSums(tij^2, na.rm = TRUE)/sigma[m] + 1/tau_beta[m] )
  beta_all[,m+1] <- mvtnorm::rmvnorm(1, mean = mean_vec_beta, sigma = diag(  1/( rowSums(tij^2, na.rm =

# Generate alpha and beta
  mu_alpha <- sum(alpha_all[,m+1])/ tau_alpha[m] / (I/tau_alpha[m] + 1/P_alpha)
  sig_alpha <- 1/( I/tau_alpha[m] + 1/P_alpha )
  alpha[m+1] <- rnorm(1, mean = mu_alpha, sd = sqrt(sig_alpha) )


  mu_beta <- sum(beta_all[,m+1]/ tau_beta[m]) / (I/tau_beta[m] + 1/P_beta)
  sig_beta <- 1 / (I/tau_beta[m] + 1/P_beta)
  beta[m+1] <- rnorm(1, mean = mu_beta, sd = sqrt(sig_beta))

# Generate Sigma
  sse <- sum( ( yij - alpha_all[,m+1] - tij * beta_all[,m+1] )^2, na.rm = TRUE )
  sigma[m+1] <- 1/ rgamma(1, shape = N/2 + a_sigma, rate = sse/2 + b_sigma)


# Generate tau_alpha and tau_beta

  tau_alpha[m+1] <- 1/ rgamma(1, shape = I/2 + a_alpha, rate = sum( (alpha_all[,m+1]- alpha[m+1])^2 )/2
  tau_beta[m+1] <- 1/ rgamma(1, shape = I/2 + a_beta, rate = sum(  (beta_all[,m+1] - beta[m+1])^2 )/2 +

}
```

```
alpha_i_estimate <- rowMeans(alpha_all[,thin])
alpha_lower <- apply(alpha_all[,thin], 1, quantile, 0.025)
alpha_upper <- apply(alpha_all[,thin], 1, quantile, 0.975)


beta_i_estimate  <- rowMeans(beta_all[,thin])
beta_lower <- apply(beta_all[,thin], 1, quantile, 0.025)
beta_upper <- apply(beta_all[,thin], 1, quantile, 0.975)

al_be_table <- cbind(c(1:I),alpha_i_estimate, alpha_lower, alpha_upper, beta_i_estimate, beta_lower, be
colnames(al_be_table) <- c("Patient", "Alpha_i(mean)", "Lower 95% CI of Alpha_i","Upper 95% CI of Alpha_
tab1 <- knitr::kable(al_be_table)
tab1
```

Patient	Alpha_i(mean)	Lower 95% CI of Alpha_i	Upper 95% CI of Alpha_i	Beta_i(mean)	Lower 95% CI of Beta_i	Upper 95% CI of Beta_i
1	174.3102	168.1024	180.8953	6.048707	5.105033	6.914313
2	167.9970	162.2953	173.6362	5.119530	4.252582	5.984225
3	170.5407	163.7297	177.8351	4.690904	3.659325	5.542255
4	173.6425	168.3729	178.8058	5.856221	5.073306	6.631359

Patient	Alpha_i(mean)	Lower 95% CI of Alpha_i	Upper 95% CI of Alpha_i	Beta_i(mean)	Lower 95% CI of Beta_i	Upper 95% CI of Beta_i
5	170.9582	164.3721	176.8484	6.497020	5.643945	7.480495
6	172.7511	166.6061	179.1483	7.413384	6.513594	8.292374
7	170.5911	164.8908	176.0705	6.249131	5.394134	7.177229
8	171.1108	165.2579	177.1511	4.944021	4.037598	5.826952
9	176.7652	171.2422	182.6087	6.798383	5.861177	7.816850
10	172.0033	166.1676	177.6873	5.824687	4.931373	6.740803
11	178.1504	171.8942	184.8241	6.705769	5.773563	7.598354
12	165.5230	159.5901	171.3530	4.490491	3.627277	5.375584
13	170.1026	164.3342	175.5919	5.308230	4.432369	6.198795
14	169.9022	163.6023	175.6835	3.647069	2.824194	4.614016
15	167.2047	162.0218	172.1501	4.755867	4.000821	5.551599
16	170.6376	164.1647	176.6377	5.040674	4.191091	5.976977
17	169.0524	163.2967	174.5920	4.719885	3.848057	5.521375
18	173.9206	167.7367	179.7523	5.683242	4.813107	6.594274
19	174.9726	169.8331	180.7749	5.325046	4.479267	6.133058
20	171.8167	166.4067	176.9661	4.903761	4.117604	5.683015
21	174.1780	168.7704	179.4815	5.714718	4.888741	6.546265
22	170.0606	163.5045	176.6285	4.954817	4.004264	5.853236
23	171.7829	166.4141	177.2948	5.327366	4.480903	6.093998
24	164.1087	157.4070	170.6059	3.343848	2.453975	4.262977
25	162.8911	156.5999	168.5201	1.992628	1.153174	2.886019
26	174.0547	168.6713	179.5048	6.259993	5.423166	7.068928
27	168.5723	161.7227	175.5256	3.744737	2.812982	4.647774
28	165.8469	160.2269	171.4999	5.657303	4.821402	6.455515
29	168.7521	162.2320	174.7799	6.299390	5.445421	7.188927
30	175.1972	169.1521	181.9110	6.825983	5.663948	7.931891
31	162.6142	156.0921	168.4905	3.908346	3.013308	4.866134
32	168.1691	162.5097	173.7722	5.281855	4.389046	6.148130
33	167.8671	162.1539	173.2411	6.092496	5.241722	6.901718
34	167.9171	161.8753	173.7247	5.254201	4.391917	6.091007
35	170.8892	165.6083	176.0871	6.023064	5.170705	6.848257
36	172.6417	167.5360	177.9514	5.494626	4.711911	6.267639
37	169.7444	164.4045	175.2974	4.413579	3.547228	5.209782
38	171.9450	165.7692	178.0990	4.973993	4.041121	5.910897
39	166.2054	160.5184	171.3680	4.345739	3.484725	5.255577
40	169.5105	163.2123	175.3487	5.629137	4.720358	6.583088
41	168.8721	163.6951	173.9858	4.971307	4.222187	5.735776
42	171.6203	165.7255	177.8961	5.193428	4.277441	6.011765
43	165.6352	160.0858	171.1232	5.193651	4.254223	6.074504
44	168.6021	163.0365	173.8995	3.657275	2.870973	4.570556
45	180.6049	174.0163	187.6693	8.495770	7.548644	9.432600
46	166.8691	161.3397	172.5809	2.944439	2.167751	3.773023
47	170.4068	164.7019	176.1938	5.316883	4.441087	6.140644
48	166.8317	160.7698	172.6876	3.279751	2.385324	4.129164
49	166.9716	160.9985	172.7818	5.367483	4.518835	6.273536
50	166.4206	161.0456	171.9528	5.413233	4.556038	6.294518
51	174.5782	169.0530	179.8277	5.529626	4.735749	6.348700
52	167.9522	162.2964	173.7216	4.162488	3.239056	5.082049
53	171.6506	166.2427	176.9677	4.854708	4.055776	5.676823
54	172.0866	166.7393	177.9369	5.026549	4.050342	5.936034
55	168.9562	163.4298	174.3116	4.879738	4.080317	5.712218

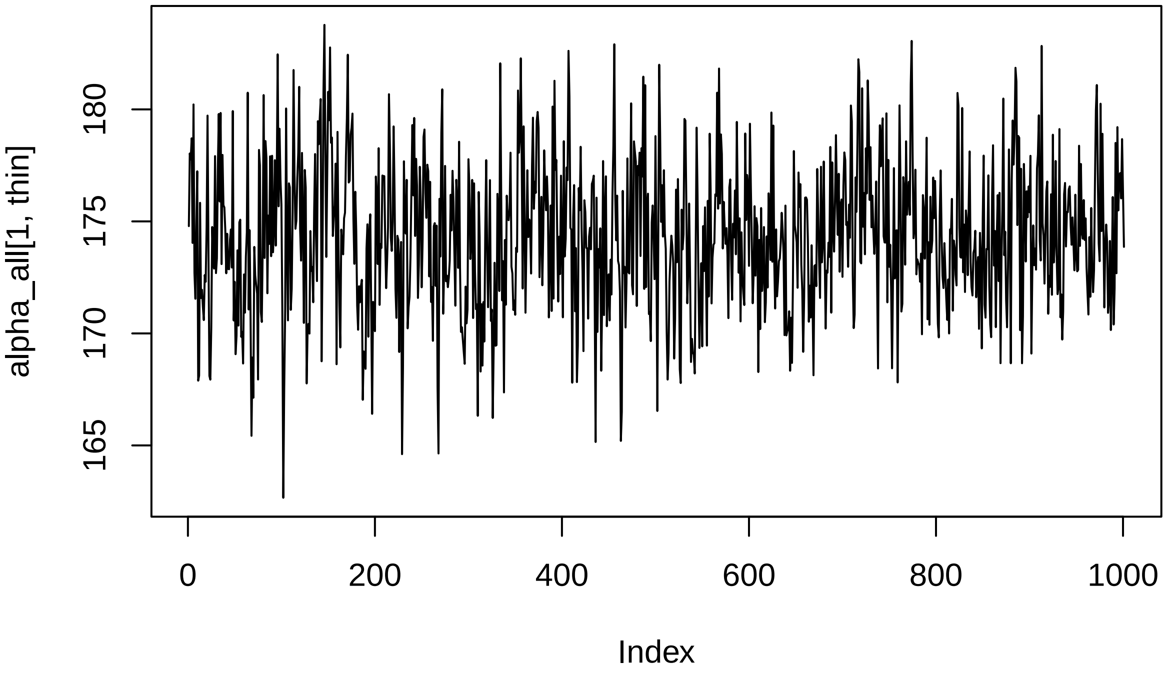
Patient	Alpha_i(mean)	Lower 95% CI of Alpha_i	Upper 95% CI of Alpha_i	Beta_i(mean)	Lower 95% CI of Beta_i	Upper 95% CI of Beta_i
56	168.2551	160.8357	174.8363	3.798964	2.746272	4.886358
57	165.0919	159.3155	170.6877	4.252413	3.389719	5.071873
58	169.0758	161.9456	176.3131	6.122126	5.160972	7.069776
59	167.6307	161.9226	173.2293	4.556321	3.652367	5.414666
60	168.1963	163.0645	173.2300	5.155193	4.395195	5.908433
61	171.4465	165.9215	177.3470	5.522304	4.666176	6.395399
62	165.3355	158.8832	171.3888	4.969304	4.017728	5.942911
63	170.2533	164.9410	175.2794	4.449010	3.682797	5.245487
64	168.4402	162.8592	174.2611	4.231184	3.348828	5.044183
65	170.6231	165.0501	175.9681	5.173174	4.389341	5.993922
66	161.2418	154.5340	167.5439	4.812064	3.873184	5.817365
67	168.9479	162.6434	175.0368	5.071343	4.213825	5.906394
68	168.4750	161.7496	174.6495	4.077161	3.092739	5.032608

```r
alpha_est <- mean(alpha[thin])
beta_est <- mean(beta[thin])
sigma_est <- mean(sigma[thin])
tau_alpha_est <- mean(tau_alpha[thin])
tau_beta_est <- mean(tau_beta[thin])
ALPHA_lower <- quantile(alpha[thin], 0.025)
ALPHA_upper <- quantile(alpha[thin], 0.975)
BETA_lower <- quantile(beta[thin], 0.025)
BETA_upper <- quantile(beta[thin], 0.975)
sigma_lower <- quantile(sigma[thin], 0.025)
sigma_upper <- quantile(sigma[thin], 0.975)
tau_alpha_lower <- quantile(tau_alpha[thin], 0.025)
tau_alpha_upper <- quantile(tau_alpha[thin], 0.975)
tau_beta_lower <- quantile(tau_beta[thin], 0.025)
tau_beta_upper <- quantile(tau_beta[thin], 0.975)

A <- c(alpha_est, ALPHA_lower, ALPHA_upper)
B <- c(beta_est, BETA_lower, BETA_upper)
C <- c(sigma_est, sigma_lower, sigma_upper)
TA <- c(tau_alpha_est, tau_alpha_lower, tau_alpha_upper)
TB <- c(tau_beta_est, tau_beta_lower, tau_beta_upper)

tab2 <- cbind(A, B, C, TA, TB)
colnames(tab2) <- c(
  "Alpha", "Beta", "Sigma","Tau_alpha", "Tau_beta"
)
rownames(tab2) <- c("Point Estimate", "95% CI Lower", "95% CI Upper")
knitr::kable(tab2)
```

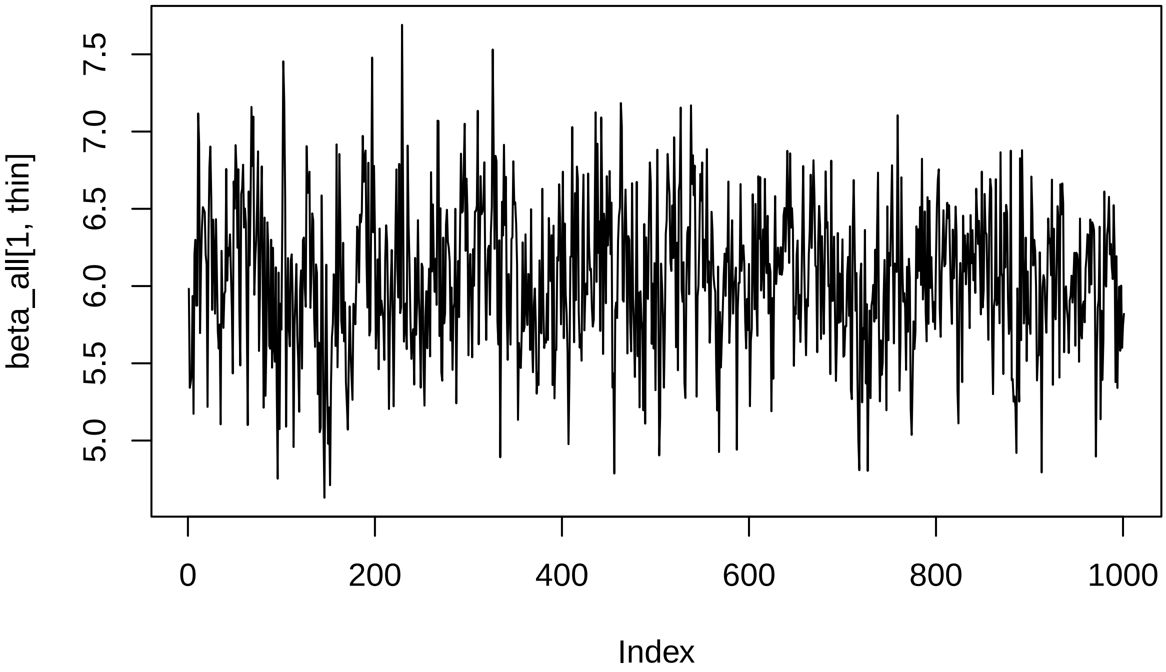
	Alpha	Beta	Sigma	Tau_alpha	Tau_beta
Point Estimate	169.2781	5.026451	9.424936	21.808482	1.3289224
95% CI Lower	167.7988	4.697103	8.038228	9.454264	0.8271465
95% CI Upper	170.6956	5.350685	11.078935	37.446560	2.0297705

Check for Convergence

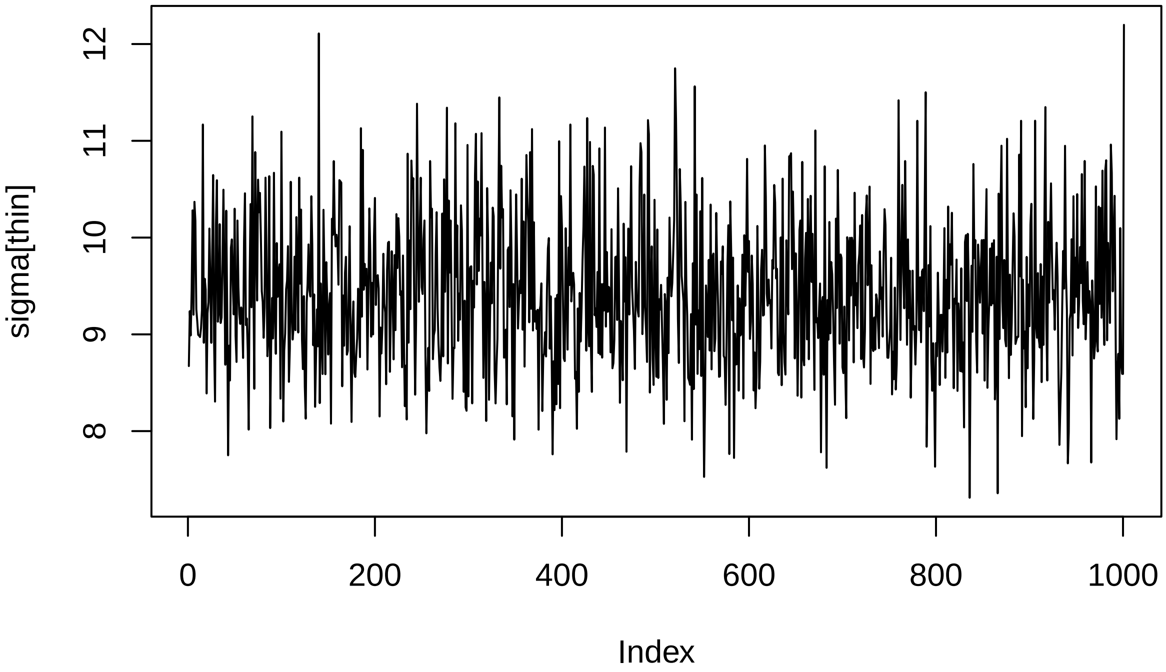
```
plot(alpha_all[1, thin], type = 'l')
```



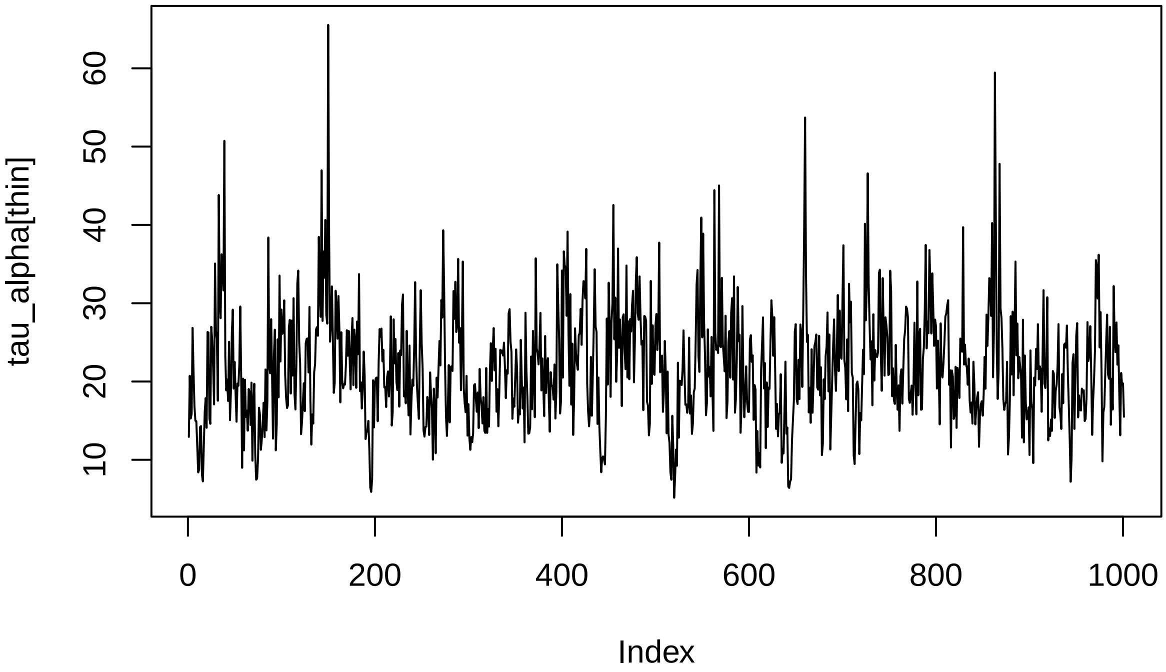
```
plot(beta_all[1, thin], type = 'l')
```



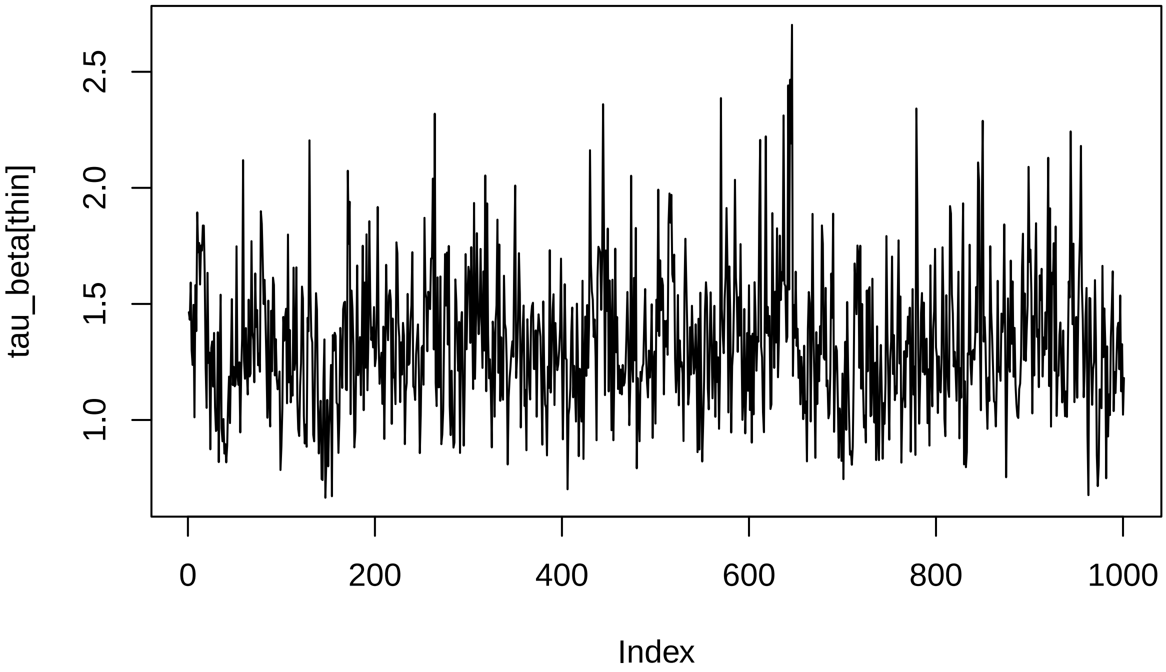
```r
plot(sigma[thin], type = 'l')
```



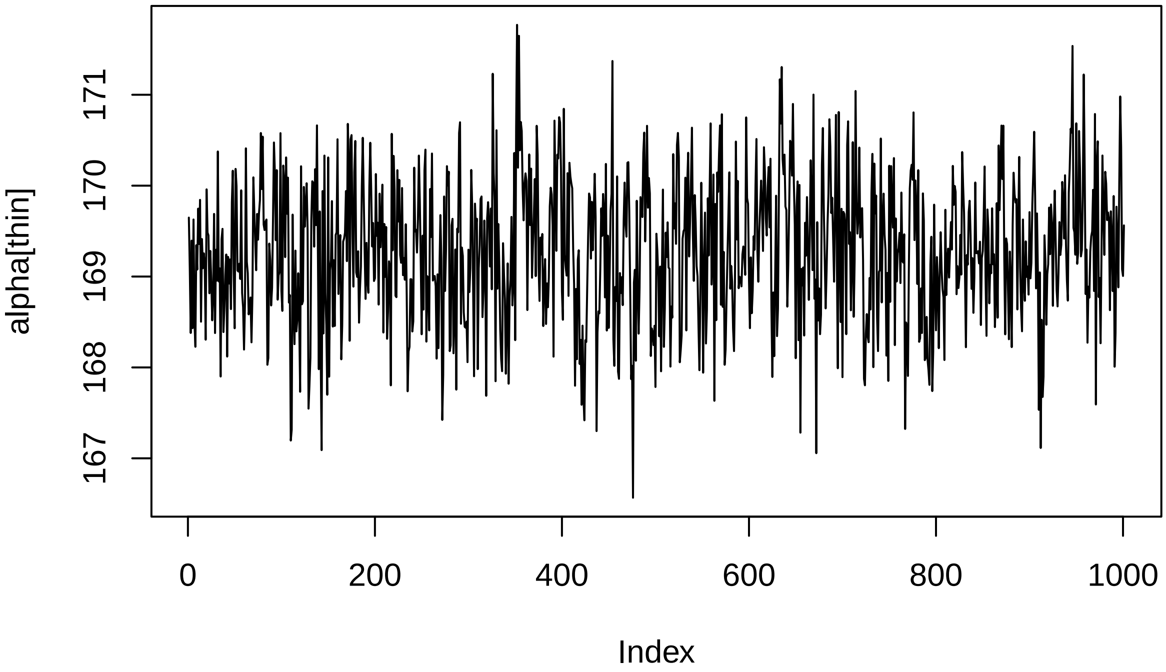
```
plot(tau_alpha[thin], type = 'l')
```



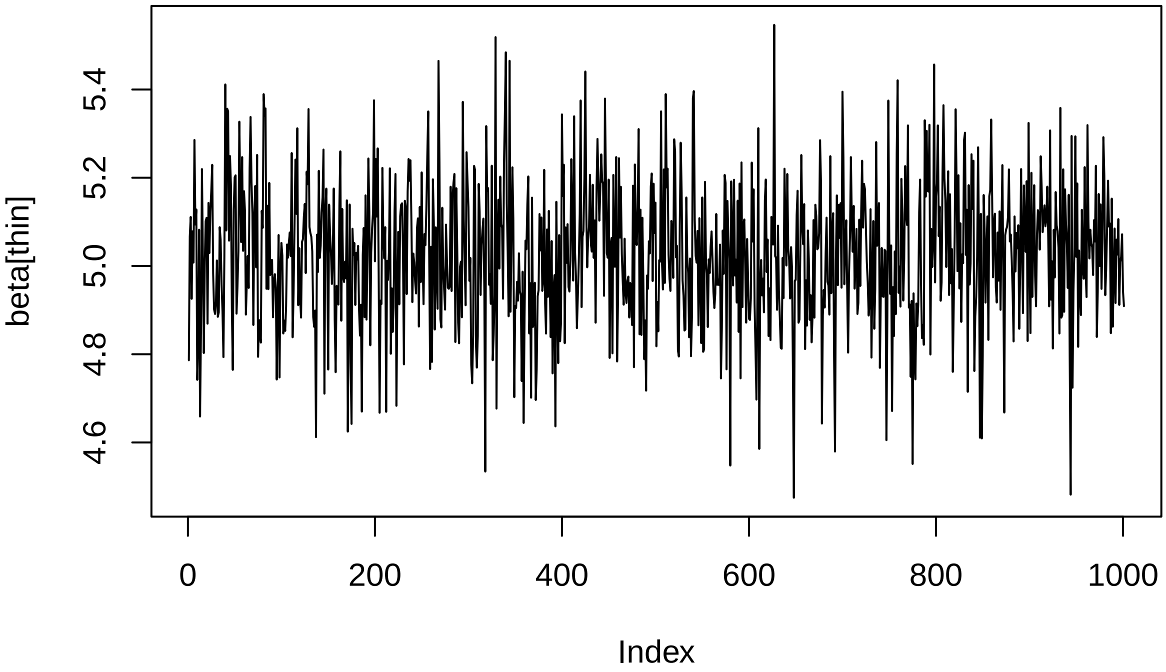
```r
plot(tau_beta[thin], type = 'l')
```



```
plot(alpha[thin], type = 'l')
```



```r
plot(beta[thin], type = 'l')
```


beta[thin]
4.6
4.8
5.0
5.2
5.4
0
200
400
600
800
1000
Index