```
#Q4
set.seed(10)
library(mcmc)
library(mvtnorm)
library(BayesLogit)
data(logit)
```

```
lupost_MHWG <- function(m, a1, b1, beta, T, lambda) {
  val <- (m+a1-1)*log(beta) + (beta-1)*sum(log(T)) -lambda*sum(T^beta) - b1*beta
  return(val)
}

lupost_linchpin <- function(m, a1, a0, b0, b1, beta, T)  {
  val <- (m+a1-1)*log(beta) + (beta-1)*sum(log(T)) - b1*beta - (m+a0)*log(b0+sum(T^beta))
  return(val)
}
```

```
MHWG <- function(T, a0, a1, b0, b1, init, n, h)  {
  m = length(T)
  acceptance_prob <- 0
  output <- matrix(0, nrow = n, ncol = 2)
  output[1,] = init
  output[1,1] = rgamma(1, shape = m+a0, rate = b0 + sum(T^output[1,2]))

  for(j in 2:n) {
    output[j,1] = rgamma(1, shape = m+a0, rate = b0 + sum(T^output[j-1,2]))

    proposed <- rgamma(1, output[j-1,2]^2/h, output[j-1,2]/h)
    if(proposed<0)  {
      proposed = output[j-1,2]
    }
    alpha <- exp( lupost_MHWG(m, a1, b1, proposed, T, output[j,1]) + ((proposed^2)/h-1)*log(proposed) - (proposed/h)*proposed -
                    lupost_MHWG(m, a1, b1, output[j-1,2], T, output[j,1]) - ((output[j-1,2]^2)/h-1)*log(output[j-1,2]) + (output[j-1,2]/h)*output[j-1,2])
    if(runif(1) < alpha)  {
      output[j,2] = proposed
      acceptance_prob = acceptance_prob+1
    } else  {
      output[j,2] = output[j-1,2]
    }

  }
  print(acceptance_prob/n)
  return(output)
}
```

```r
linchpin <- function(T, a0, a1, b0, b1, init, n, h) {
  m = length(T)
  acceptance_prob <- 0
  output <- matrix(0, nrow = n, ncol = 2)
  output[1,] = init
  for(j in 2:n) {
    proposed <- rgamma(1, output[j-1,2]^2/h, output[j-1,2]/h)
    if(proposed<0)  {
      proposed = output[j-1,2]
    }
    alpha <- exp( lupost_linchpin(m, a1, a0, b0, b1, proposed, T) + ((proposed^2)/h-1)*log(proposed) - (proposed/h)*proposed -
                  lupost_linchpin(m, a1, a0, b0, b1, output[j-1,2], T) - ((output[j-1,2]^2)/h-1)*log(output[j-1,2]) + (output[j-1,2]/h)*output[j-1,2])
    if(runif(1) < alpha)  {
      output[j,2] = proposed
      acceptance_prob = acceptance_prob+1
    } else  {
      output[j,2] = output[j-1,2]
    }

    output[j,1] = rgamma(1, shape = m+a0, rate = b0 + sum(T^output[j,2]))
  }
  print(acceptance_prob/n)
  return(output)
}
```

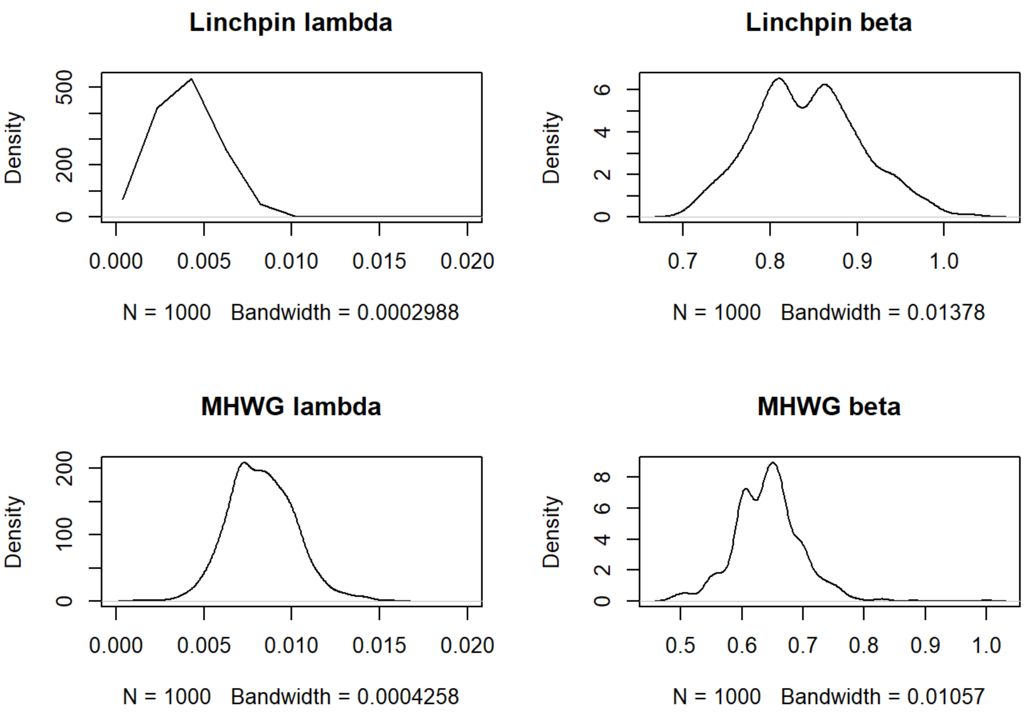
```r
T <- c(387,182,244,600,627, 332,418,300,798,584,660,39,274,174,50,
       34,1895,158,974,345,1755,1752,473,81,954,1407,230,464,380,131,1205)
a0= 2.5
b0= 2350
a1= 1
b1= 1
init <- c(1,1)
n=1e3
h = 0.02
linchpin_output <- linchpin(T, a0, a1, b0, b1, init, n, h)
```

```
## [1] 0.434
```

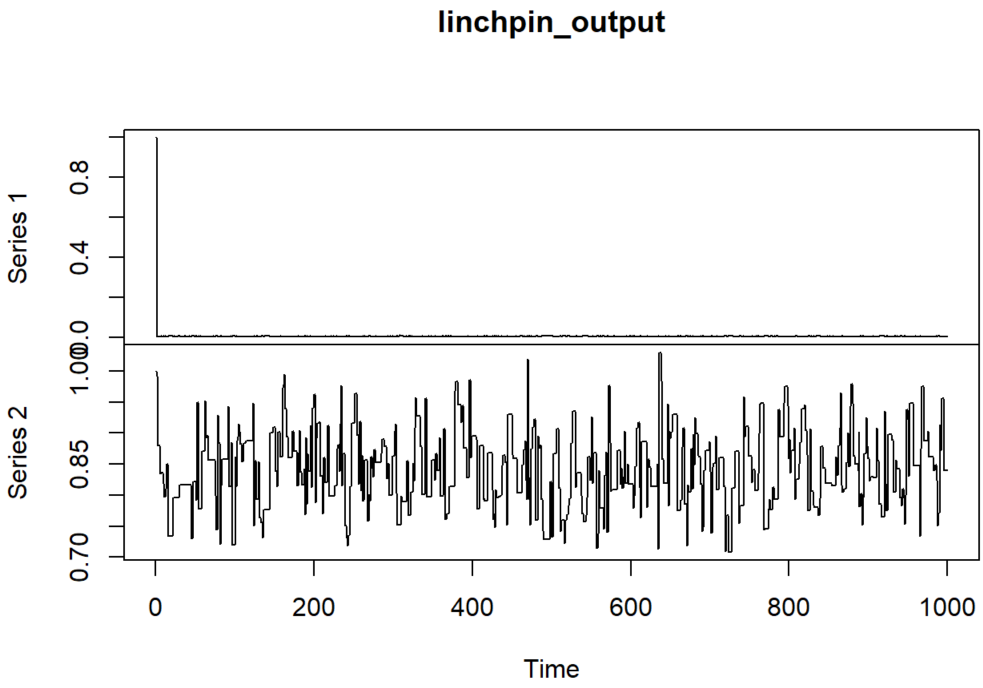
```r
h = 0.009
MHWG_output <- MHWG(T, a0, a1, b0, b1, init, n, h)
```

```
## [1] 0.443
```

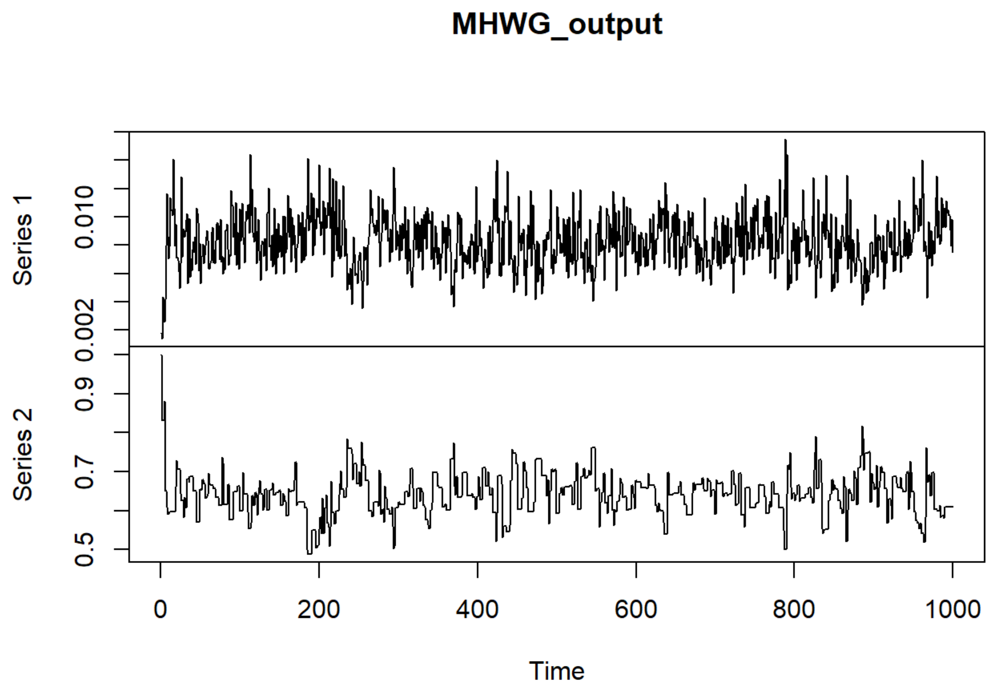
```r
par(mfrow = c(2,2))
plot(density(linchpin_output[,1]), xlim=c(0,0.02), main = "Linchpin lambda")
plot(density(linchpin_output[,2]),  main = "Linchpin beta")
plot(density(MHWG_output[,1]), xlim=c(0,0.02), main = "MHWG lambda")
plot(density(MHWG_output[,2]), main = "MHWG beta")
```



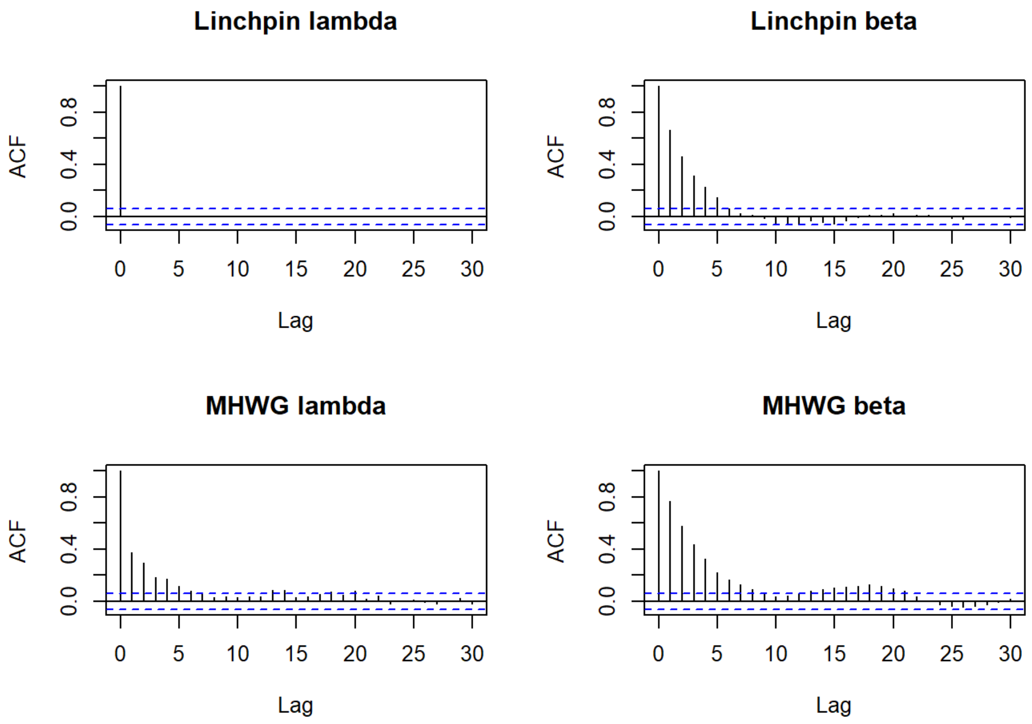
```
par(mfrow = c(2,1))
plot.ts(linchpin_output)
```



```
plot.ts(MHWG_output)
```



```
par(mfrow = c(2,2))
acf(linchpin_output[,1], main = "Linchpin lambda")
acf(linchpin_output[,2], main = "Linchpin beta")
acf(MHWG_output[,1], main = "MHWG lambda")
acf(MHWG_output[,2], main = "MHWG beta")
```



```
#Q5
set.seed(10)

lupost <-function(m, X, y, beta)  {
  val <- 0
  for(i in 1:m) {
    val <- val + (X[i,]%*%beta)*y[i] - log(1 + exp(X[i,]%*%beta) )
  }
  val <- val - (t(beta)%*%(beta))/200
  return(val)
}

proposal <- function(x, h, p)  {
  return(rmvnorm(1, mean = x, sigma = h*diag(1,p,p)))
}
```

```
MH <- function(X, y, h, n, init)  {
  acceptance_prob <- 0
  p <- dim(X)[2]
  m <- dim(X)[1]
  output <- matrix(0, nrow = n, ncol = p)
  output[1,] = init
  for(j in 2:n) {
    proposed <- proposal(output[j-1,], h, p)
    alpha <- exp(lupost(m, X, y, t(proposed)) - lupost(m, X, y, output[j-1,]))
    if(runif(1) < alpha)  {
      output[j,] = proposed
      acceptance_prob = acceptance_prob + 1
    } else  {
      output[j,] = output[j-1,]
    }
  }
  print(acceptance_prob/n)
  return(output)
}
```

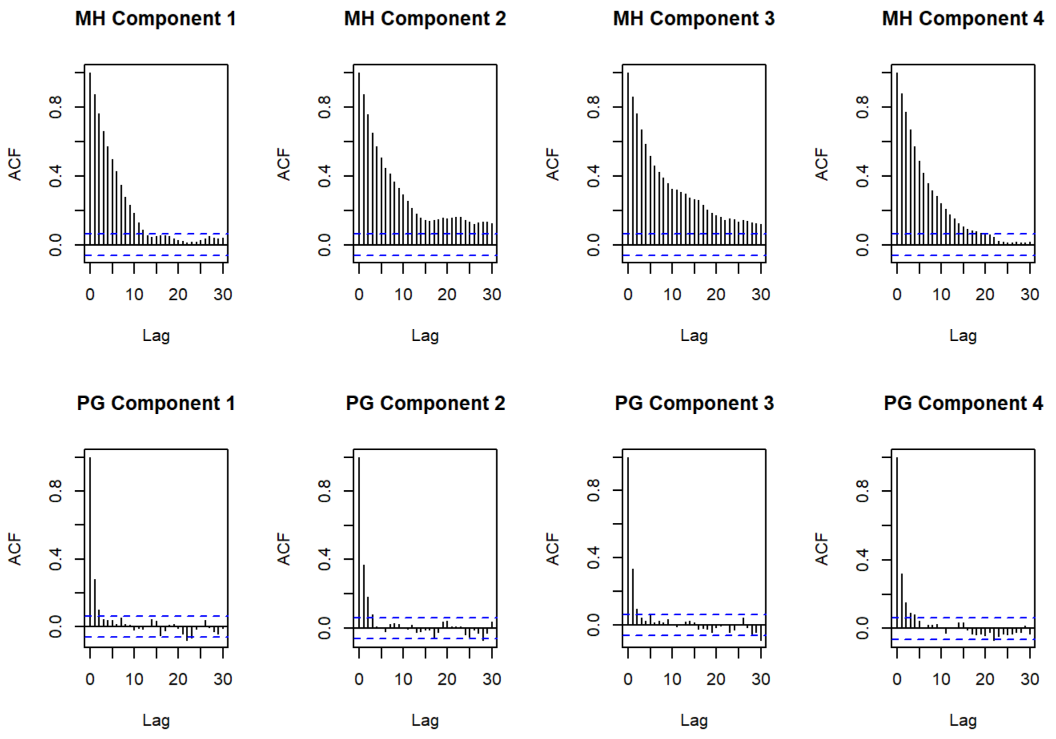
```
polya_gamma <- function(X, y, init, n){
  m <- dim(X)[1]
  p <- dim(X)[2]
  output <- matrix(0, nrow = n, ncol = p)
  output[1,] = init
  K <- y - 1/2
  W <- diag(0, m, m)
  for(j in 2:n) {
    for(i in 1:m) {
      W[i,i] = rpg(1,1, (X[i,]%*%output[j-1,]))
    }
    Vw <- solve(t(X)%*%W%*%X + 0.01*diag(1,p,p))
    mw <- Vw%*%(t(X)%*%K)
    output[j,] <- rmvnorm(1, mean = mw, sigma = Vw)
  }
  return(output)
}
```

```
data(logit)
X <- as.matrix(logit[,-1])
y <- as.matrix(logit[,1])
p <- dim(X)[2]
n <- 1e3
init <- rep(0, p)
h <- 0.2
MH_output <- MH(X, y, h, n, init)
```

```
## [1] 0.242
```

```
PG_output <- polya_gamma(X, y, init, n)
```

```
par(mfrow = c(2,4))
acf(MH_output[,1],main="MH Component 1")
acf(MH_output[,2],main="MH Component 2")
acf(MH_output[,3],main="MH Component 3")
acf(MH_output[,4],main="MH Component 4")
acf(PG_output[,1],main="PG Component 1")
acf(PG_output[,2],main="PG Component 2")
acf(PG_output[,3],main="PG Component 3")
acf(PG_output[,4],main="PG Component 4")
```


MH Component 1
MH Component 2
MH Component 3
MH Component 4
PG Component 1
PG Component 2
PG Component 3
PG Component 4
ACF
Lag